
MSAT: Multi-stage adaptive threshold for Deep Spiking Neural Networks

David S. Hippocampus*

Department of Computer Science
Cranberry-Lemon University
Pittsburgh, PA 15213
hippo@cs.cranberry-lemon.edu

Abstract

Spiking Neural Networks(SNNs) can do inference with low-power consumption natively because of its spike sparsity. Compared with the other two training method: STDP and BP, Conversion from Artificial Neural Networks(ANNs), is a more easier way to achieve deep SNNs and commonly have the approximate performance compared with ANN. However, Conversion SNNs suffer from a accuracy degradation and more latency at inference time. Lots of studies have tried to make a trade-off between improving accuracy and reduce the latency using varied method including adjust ANN topology when mapping ANN to SNN, using a more efficient firing mechanism et.al Here we analyze conversion loss layer-to-layer and point it out that membrane potential matters in both SNN accuracy and inference latency. subsequently, we give a new perspective that most of current conversion method is optimization membrane potential to achieve higher accuracy and short latency. Meanwhile, Different from current conversion schemes which use the same and invariant threshold with inference time in a layer, we propose a multi-stage adaptive threshold for deep spiking neural Networks. We examine the performance on CIFAR-100 and ImageNet for classification. Furthermore, we show the propose method also behave well in objection detection on VOC and COCO. All above provide support on biological interpretability.

1 Introduction

At present, Artificial Neural Network (ANN) is widely used in speech recognition, image processing and other fields. However, with the complexity of neural networks increasing progressively, running such deep networks often requires large amounts of computational resources, such as memory and power. In addition, current ANN's work mechanism differs from our brain. Actually, Neurons in the brain communicate by transmitting sequences (i.e. spike) generated by action potentials. Spiking Neuron network (SNN) works in a similar way. It also transmits the spike sequence to the downstream neurons. These spikes often carry a high amount of information, and the spike distribution is sparse, so it has the characteristics of low power consumption.

SNNs potentially offer an efficient way of doing inference when it combines with neuron computing hardware, furthermore, SNN inherently shows efficiency on processing temporal and spatial data. Its diverse coding mechanisms, and events-driven characteristics are also promising. However, because the internal state variables of neurons do not satisfy the continuously differentiable requirement, it is difficult to be trained. To solve this problem, some algorithms based on the rules of gradient descent and spike-time dependent plasticity (STDP) were proposed, which had partly solved the problem

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

of training SNNs. Frustratingly, It is still difficult to train deeper SNNs with complex network structures, and results in a remaining of huge gap of performance between SNNs and CNNs in complex recognition or detection tasks.

To narrow the performance gap between SNNs and CNNs, methods of converting CNNs to SNNs had been proposed. In these methods, a CNN is firstly trained using the standard stochastic gradient descent and back propagation algorithm, and then the trained weights are mapped to an SNN with the same structure as the CNN. Inference is performed on the converted SNNs. The main idea is that the firing rates of spiking neurons can approximate the activations of their counterparts (RELU) in ANNs with sufficient time steps. This finding has become the fundamental principle underlying the conversion scheme. Converted SNNs often suffer from a accuracy degradation and more latency at inference time. Lots of studys have tryed to make a trade-off between improving accuracy and reduce the latency using var- ied method including adjust ANN topology when mapping ANN to SNN, using a more efficient fring mechanism et.al. Here we forms conversion loss formula and shows that residual membrane potential in each IF neuron increase the latency which mean firing rates approximate to activation value. We also find that most of current conversion schemes, they use threshold invariant with inference time and are same and in a layer. This mechanism is inconsistent with a phenomenon which has been widely observed in the central nervous system, e.g. visual cortex , auditory midbrain, hippocampus, somatosensory cortex. It has been proposed that threshold variability reflects an adaptation of the spike threshold to the membrane potential. Inspired by this, we propose a multi-stage adaptive threshold for deep spiking neural Networks. For each neuron, its threshold vaires with its own membrane potential. We both do experimental on object recognition and detection in non-trivial datasets to prove proposed method is as well as efficiency with the current mainstream schemes when doing visual tasks.

Our major contribution can be summarized as:

- sufficient experimental on object recognition and detection in non-trivial datasets, shows that our proposed method is both efficiency and biological interpretability
- a formula on layer-by-layer conversion error, a new perspective diving existing method into three part
- a multi-stage adaptive threshold mechanism, which is widely existing in the center nervous system and thus more biological plausible. We use it for deep spiking neural Networks.

2 Preliminaries

Our conversion pipeline exploits the threshold balancing mechanism (Diehl et al., 2015; Sengupta et al., 2018) between ANN and SNN with modified ReLU function on the source ANN to reduce the consequential conversion error.

The main idea in ANN-to-SNN conversion is using mean firing rate $r_i^l(t)$ which indicates firing rate of neuron i in layer l for a total time t to approximate the activation value a_i^l in SNN. Here we give analytical explanation for the approximation.

In ANN, the neuron i activation value(after relu) in layer l a_i^l can be computed as:

$$a_i^l = \max \left(0, \sum_{j=1}^{M^{l-1}} W_{ij}^l a_j^{l-1} + b_i^l \right) \quad (1)$$

here $l \in \{1, \dots, L\}$ indicates layer l in a network with L layers; W_{ij}^l indicates weight connection between neuron i in layer l and neuron j in layer $l - 1$; b_i^l indicates neuron i bias in layer l ; it is worth noting that a_i^l start from $l = 0$ and $a^0 = x$.

Neuron Model postsynaptic membrane potential(PSP) at timestep t , $V_i^l(t)$ is a sum of last timestep $t - 1$ membrane potential and current input electric current. When PSP exceeds a certain voltage threshold, it emits an output spike and reset the membrane potential. One of the most widely adopted model is Integrate-and-Fire (IF) neuron, and membrane potential at the next time step t would then be updated by soft-reset mechanism, which subtract threshold in PSP rather than

reset the membrane potential to V_{reset} . The mathematical form is as follows.

$$V_i^l(t) = V_i^l(t-1) + V_{th}^l \left(\sum_j^{M^{l-1}} W_{ij}^l \Theta_{t,j}^{l-1} + b_i^l \right) - V_{th}^l \Theta_{t,i}^l \quad (2)$$

here $\Theta_{t,i}^l$ is a function indicating the neuron i in layer l occurrence of a spike at time t

$$\Theta_{t,i}^l = \Theta(V_i^l(t-1) + z_i^l(t) - V_{th}^l), \text{ with } \Theta(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \quad (3)$$

Here $z_i^l(t)$ is neuron i inputs in layer l and time t

$$z_i^l(t) = V_{th}^l \left(\sum_j^{M^{l-1}} W_{ij}^l \Theta_{t,j}^{l-1} + b_i^l \right) \quad (4)$$

3 Diving Conversion Error

Error comes from two part: one is converting ANN to SNN directly, result quantization error and clip error; the other is neuron transient dynamics and iiregular elicited spike, Spikes of Inactivated Neurons (SIN) error. show as fig 1

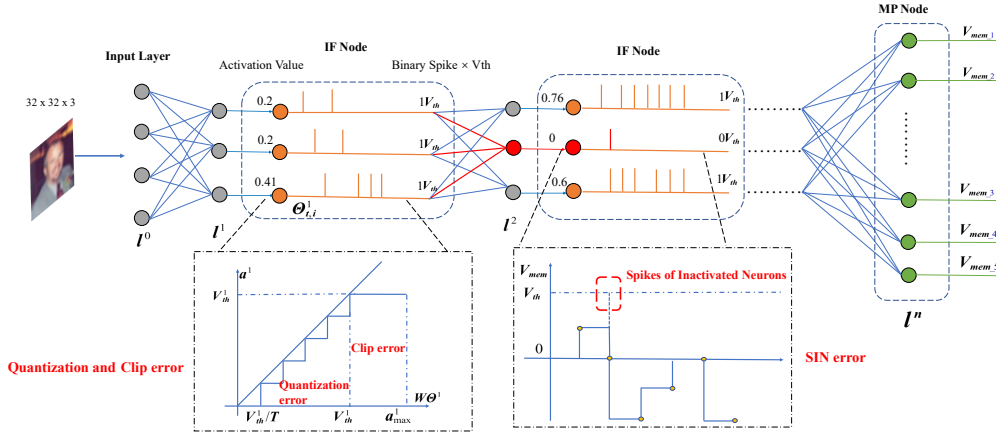


Figure 1: Two part error: one is quantization error and clip error result from converting ANN to SNN directly, the other is spikes of inactivated neurons error whose existence result from iiregular elicited spike. A two layer multilayer perceptron network for demonstration.

3.1 Layer-By-Layer Quantization Error

For equation (2), cumulate the input over the simulation timestep T , we can derive the firing rate relationship layer-to-layer.

$$r_i^l(T) = \sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} + b_i^l - \frac{V_i^l(T)}{TV_{th}^l} \quad (5)$$

Unfolding for each layer, relationship between mean firing rate and activation value can be shown as following:

$$\begin{aligned}
r_i^l(T) &= \sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} + b_i^l - \frac{V_i^l(T)}{TV_{th}^l} \\
&= \sum_j^{M^{l-1}} W_{ij}^l \left(\sum_k^{M^{l-2}} W_{jk}^{l-1} r_k^{l-2} + b_j^{l-1} - \frac{V_j^{l-1}(T)}{TV_{th}^{l-1}} \right) + b_i^l - \frac{V_i^l(T)}{TV_{th}^l} \\
&= \sum_j^{M^{l-1}} W_{ij}^l \left(\sum_k^{M^{l-2}} W_{jk}^{l-1} \left(\underbrace{\sum \dots \sum_m^1 W_{1m}^1 x_m + b_m^1}_{a_m^1} - \frac{V_1^1(T)}{TV_{th}^1} + \dots + b - \frac{V(T)}{TV_{th}} \right) \right. \\
&\quad \left. + b_j^{l-1} - \frac{V_j^{l-1}(T)}{TV_{th}^{l-1}} \right) + b_i^l - \frac{V_i^l(T)}{TV_{th}^l}
\end{aligned} \tag{6}$$

Use ΔV_i^l denotes $\frac{V_i^l(T)}{TV_{th}^l}$

$$r_i^l(T) = a_i^l - \Delta V_i^l - \sum_j^{M^{l-1}} W_{ij}^l \Delta V_j^l - \dots - \sum_j^{M^{l-1}} W_{ij}^l \dots \sum_k^1 W_{1k}^1 \Delta V_k^1 \tag{7}$$

Note that the activation value is strictly fall in $[0, 1]$ by using weight normalization and residual membrane potential $V_i^l(T)$ falls into $[0, V_{th}^l]$ so that it will enable us to estimate the activation function of SNNs ignoring the effect of unevenness error which degenerate to the quantization error as ? mentioned. So the weights are not origin ANN weight and are scaled by V_{th}^{l-1}/V_{th}^l and bias are scaled by V_{th}^l and V_{th} is set to 1. It has the same form with threshold balancing, the different is that threshold balancing use postsynaptic neuron threshold times firing spike to compute mean firing rate. Actually, Two method: weight normalization and threshold balancing are mathematically equivalent. We use threshold balancing in rest of paper for convenience.

Bias is a constant all the time so it doesn't affect the conversion error and we omit it and in threshold balancing, mean firing rate $r_i^l(t)$ is PSP average value, so equation (5) becomes

$$r_i^l(T) = \sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} - \frac{V_i^l(T)}{T} \tag{8}$$

When V_{th}^l is larger than maximum of activation value, $V_i^l(T)$ will be less than V_{th} thus the residual membrane potential cannot be output that's why information transmission suffer a loss. There error is by nature because the discrete of timestep that the mean firing rate is a step function which cannot exactly approximate the source continuous RELU function, which is known as quantization error (flooring error), is can be expressed as

$$r_i^l(T) = \frac{V_{th}^l}{T} \left\lfloor \frac{\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} T}{V_{th}^l} \right\rfloor \tag{9}$$

3.2 Maximum Activation Clip Error

As mentioned in equation (9), if voltage threshold is set less than maximum activation value, then when PSP exceeds voltage threshold, the emitted spike will not transmit efficient information to distinct above PSP. Set voltage threshold to maximum activation value can avoid this but suffer a huge latency. This is a trade-off, and in ?, choose quantile p for different datasets. (Li) propose a

Bayesian Optimization to find this p value. The Quantization error and Clip error and be expressed as

$$r_i^l(T) = \text{clip} \left(\frac{V_{th}^l}{T} \left\lfloor \frac{\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} T}{V_{th}^l} \right\rfloor, 0, V_{th}^l \right) \quad (10)$$

3.3 Spikes of Inactivated Neurons (SIN) Error

Spikes of Inactivated Neurons (SIN) is a group of neuron whose activation value counterparts in ANN is negative. Theoretically they should not fire spike in total timestep to achieve ReLU filtering the negative value, however they fire spike that activation value a_i^l should zero while corresponding mean firing rate r_i^l is larger than zero. We use $\mathcal{R} = \left\{ j \mid \sum_{t=0}^T \Theta_{t,j}^l > 0, a_j^l(t) < 0 \right\}$ to denote SIN, then SIN error can be expressed as

$$r_i^l(T) = \mathbf{0} - \frac{\sum_{j \in \mathcal{R}} \sum_{t=0}^T W_{ij}^l \Theta_{t,j}^l}{T} \quad (11)$$

The neurons with SIN usually fire early and then silence during the conversion process. We statistics SIN proportion in each layer, as shown in fig 2. It shows that SIN is prevalent and have larger proportion in deeper layer. What's more, It is clear that most of SIN appear in early 32 timestep, that's why early timestep accuracy suffer from more degradation.

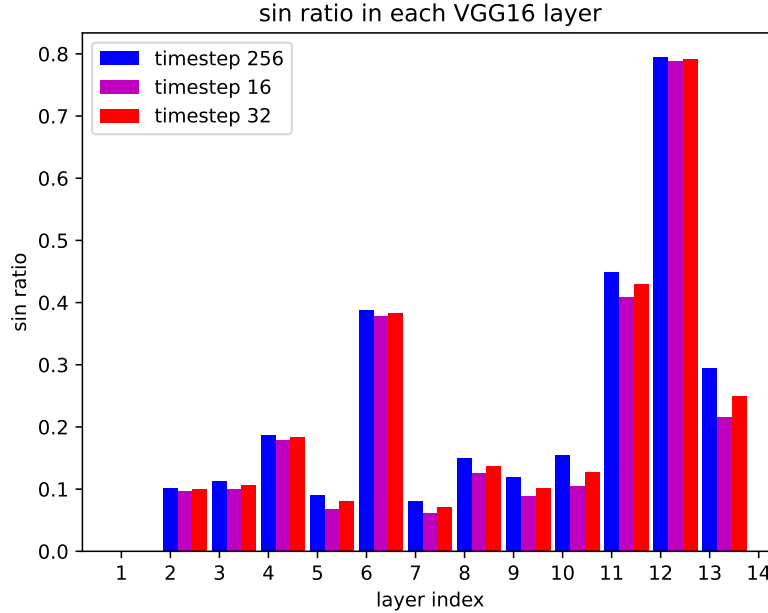


Figure 2: SIN ratio in each VGG16 layer. SIN have larger proportion in deeper layer early, account for early timestep accuracy suffer from more degradation.

4 Method

4.1 adaptive threshold optimization error

Let's take a look at existing methods, no matter weight normalization or threshold balancing, they aim at zipping the gap between ANN and SNN. Though, we should be aware that the real advantage of SNN is its sparse spike which simultaneously low-power and brain-Inspired. Current method,

however, set threshold voltage as the same in the same layer and these threshold will remain unchanged despite inference time increasing. It ignores a fact that neurons in different regions of brain represent distinct dynamics and process information differently from other regions?. The threshold voltage of neurons is also known to have a broad range rather than a single value, known as homeostasis?. Some neuroscience literature also indicates that threshold value is variable in the same neuron and threshold variability is a genuine feature of neurons?????. Thus the voltage threshold should different from neurons and timestep. Here we demonstrate that voltage threshold is a function of timestep t and the transmits the equivalent information with the constant threshold.

omitting the bias, the equation(5) can be rewritten as the following form

$$V_i^l(t) = V_i^l(t-1) + \sum_j^{M^{l-1}} V_{th,j}^l(t) W_{ij}^l \Theta_{t,j}^{l-1} - V_{th,i}^l(t) \Theta_{t,i}^l \quad (12)$$

firing rate during timestep T is computed as

$$r_i^l(T) = \frac{\sum_{t'=1}^T V_{th,i}^l(t') \Theta_{t',i}^l}{T} \quad (13)$$

the firing rate relationship in higher layer, it means equation ?? still satisfy so the conversion error form is the same. But note that the residual neuron membrane potential $V_i^l(T)$ can be appropriately adjusted, so the spike information could be more efficiency and thus shorten the conversion latency.

In threshold adaptation, the optimization target is

$$\min_{V_{th,i}^l(T)} \left(\text{clipfloor} \left(\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1}, T, V_{th,i}^l(T) \right) - \frac{\sum_{j \in \mathcal{R}} \sum_{t=0}^T W_{ij}^l \Theta_{t,j}^l}{T} - \text{ReLU} \left(\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} \right) \right) \quad (14)$$

There is no closed-form solution to above problem, ? use grid search to find final solution, this is still a heuristic method. A trivial solution to this is $V_{th,i}^l = \sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1}$, it means voltage threshold equals to input value for each neuron. With this solution, ANN can be converted to SNN only one timestep, however, this SNN elicit spike every timestep and lose spike sparsity, so it is not reasonable and makes no sense. In next subsection, we will give a more biologically rational method to find a solution qualitatively.

4.2 Multi-stage adaptive threshold

In vivo, the spiking threshold displays large variability. Paper ? show that this phenomenon has been widely observed in the central nervous system, e.g. visual cortex ??, auditory midbrain ?, hippocampus ?, somatosensory cortex ?. However, to our best knowledge, threshold varies in conversion is fewly used in conversion ANN to SNN. Work??? while they only use two-stage or heuristic method and still cannot represent the homeostasis well. The other work? use this biologically inspired dynamic thresholds mechanism to do robot obstacle avoidance and continuous control tasks.

Inspired by this, we propose a adaptive threshold, which is multi-stage and varies with inference time. The method can be briefly summed up as: **varies with firing history and input properties**. Specifically, spike threshold is positively correlated with the average V_i preceding spikes and negatively correlated with the rate of depolarization. It is consistent with neuroscience literature spike-threshold adaptation ?? and ? separately. some other threshold adaptation models such as the threshold increases after each spike and decreases if there is no spike?? is relatively simple and not fitted to this problem so we donot adopted. The relationship between threshold and membrane potential and rate of depolarization is described as

$$V_{th,i}^l(t+1) = \tau_{mp} V_{th_{mp},i}^l(t) + \tau_{rd} V_{th_{rd},i}^l(t+1) \quad (15)$$

Where τ_{mp} and τ_{rd} is the time constant of the dynamic tracking threshold $V_{th_{mp},i}^l(t)$ and dynamic evoked threshold $V_{th_{rd},i}^l(t+1)$ separately.

dynamic tracking threshold(DTT) DTT is a flection of spiking threshold vaires with firing history. It shows that spike threshold depends on preceding membrane potential and tracking the membrane potential at a short timescale due to inactivation of sodium channel??? or the activation of potassium channels??. in ?, the DTT is a similar first-order kinetic equation, we here use steady-state threshold for fitting our SNNs. we use $V_{m,i}^l(t)$ to denote the average membrane potential during timestep t in layer l neuron i , then DTT is following:

$$V_{th_mp,i}^l(t) = \left(\alpha (V_i^l(t) - V_m^l(t)) + V_T^l + k_a \ln \left(1 + e^{\frac{V_i^l(t) - V_m^l(t)}{k_i}} \right) \right) \quad (16)$$

here η, k_i is both time constant, V_T^l is the parameters to optimize. when PSP is less than average membrane potential $V_m^l(t)$, the slope is η on the left side of the knee. The slope on the right side is $\frac{k_a}{k_i} + \alpha$. The curvature C is determined by $\alpha, k_a, k_i, V_T^l, V_i^l(t)$.

the threshold increases with residual membrane potential and thus any voltage fluctuations that are slower than threshold adaptation should not have an impact on output spiking, this is indirectly relieve the spike of inactivated neuron error and clip error.

dynamic evoked threshold(DET) DET is a flection of spiking threshold vaires with input properties. Paper? show spike threshold vaires inversely with preceding rate of depolarization(rd) dV_m/dt by plotting scatter. And dependence of threshold dynamics on rd is due to decrease in the availability of sodium channels. In IF neuron, we use membrane potential before spike variation to express rd thus DET can be expressed as

$$V_{th_rd,i}^l(t+1) = \tau_{rd} e^{-\frac{(V_i^l(t+1) - V_i^l(t))}{C}} \quad (17)$$

Here use bold $V_i^l(t)$ to denote membrane potential before spike and make a distinction with $V_i^l(t)$ which denote residual membrane potential after spike. C is time constant flecing the insensitive to input membrane potential variation. The threshold decreases with input value exponentially.

Interaction of DET and DTT Take together, the link between preceding spike membrane potential and negatively correlated with the rate of depolarization, shows that threshold adaptation neurons selective to fast input variations and remarkably insensitive to slow ones.

In other words, the slow voltage fluctuations are filtered out by threshold adaptation and will not elicit spike. The slow voltage fluctuations may come from unexpected spike of inactivated neuron so it relieve the SIN error partly and reduce the total spike number meanwhile promoting the energy efficiency.

On the other hand, the threshold variation by DTT ensure voltage threshold will not raise big gap with maximum activation value thus narrowed clip error; and the threshold variation by DET makes threshold not increase endlessly and reduce appropriately for fear of large quantization error.

It should be noted that DTT cannot guarantee positive value because the difference between current residual membrane potential and average membrane potential is may negative. Negative threshold is not we want so we let $V_{th,i}^l(t+1)$ pass a sigmoid function and map it to $[0, 1]$. We use this value multiply maximum activation value as truely threshold.

4.3 Spike Confidence

As shown in fig , the early spike elicited is not truely all and some part of them are raised by SIN, though threshold adaptive DTT can partly relieve this, SIN will not be distinguished with other normal neuron untill they keep silence in longer timestep. It hints us we should not totally believe the early spike. As many image cognition do, we also import a confidence to show how confident the elicited spike is that they are outcome of normal neuron so not SIN. So we give different spike confidence to different layer and in early timestep every spike which should elicit will not fire directly but use this spike confidence to determine spike or not. This is somewhat like dropout? to

normalize and relieve overfitting. Formula expression as follows

$$\begin{aligned}
c_j^l &\sim \text{Bernoulli}(p) \\
\tilde{\Theta}_{t,i}^l &= c_j^l * \Theta_{t,i}^l \\
V_i^l(t) &= V_i^l(t-1) + \sum_j^{M^{l-1}} V_{th,j}^l(t) W_{ij}^l \tilde{\Theta}_{t,j}^{l-1} - V_{th,i}^l(t) \tilde{\Theta}_{t,i}^l
\end{aligned} \tag{18}$$

Above equation ensures that all the output neurons are used and adjust the neurons, thresholds to the stimuli for which they become specialized. The pseudocodes for adaptive threshold algorithm are shown in Algorithm 1.

Algorithm 1 Conversion from ANN to SNN: Multi-stage adaptive threshold

Require: Pretrained ANN, training set, SNN’s inference timestep T, spike confidence list C

Ensure: The converted SNN firing rate approximate ANN activation value with shorter latency

```

1: for s = 1 to # of samples do
2:    $a_l \leftarrow$  layer-wise activation value
3:   for l = 1 to L do
4:      $V_{th}^l \leftarrow \max[V_{th}^l, \max(a_l)]$ 
5:      $SNN.layer[l].V_{th} \leftarrow V_{th}^l$ 
6:   end for
7: end for
8: for t = 1 to timestep T do
9:   for l = 1 to L do
10:    spike confidence  $c^l \leftarrow \text{Bernoulli}(C[l])$ 
11:    for j = 1 to neuron number of layer l do
12:      compute DTT and DET as formula (17)(19) to get adaptation value
13:       $SNN.layer[l].V_{th}[j] \leftarrow \text{sigmoid}(DTT + DET) * V_{th}^l$ 
14:      if fire spike and t < early timestep then
15:        spike =  $c^l$  * spike
16:      end if
17:    end for
18:  end for
19: end for

```

5 Related Work

others

6 Experiment

Extensive experiments

Table x: detection mAP on VOC and COCO for our converted SNNs, and compared to other conversion methods and ANN.

7 Discussion

Indeed a trivial solution to the fitting problem is the threshold model defined by and th 0 ms: the spike threshold always equals the membrane potential, in particular at the upstroke of spikes. To avoid these problems, we instead used the threshold model to predict the occurrence of spikes and their precise timing based only on Vm. The trivial solution mentioned above is a poor predictor of spikes since it would predict too many spikes

Table 1: Experimental results on CIFAR100

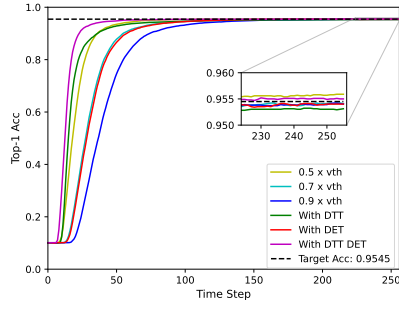
Method	Use DT	ANN	SNN Best	T=32	T=64	T=128	T=256
VGG16, CIFAR100							
p-Norm ?	×	78.49	58.44	44.88	51.89	56.02	58.44
Channel-Norm?	×	78.49	74.74	54.03	67.34	72.50	74.73
Spike-Norm?	×	71.22	70.77	-	-	-	-
TSC?	×	71.22	70.97	-	-	69.86	70.65
RMP-SNN?	×	71.22	70.93	-	-	63.76	68.34
Opt.?	×	77.89	77.71	7.64	21.84	55.04	73.54
Calibration?	×	77.89	77.87	73.55	76.64	77.40	77.68
Burst ?	×	78.49	78.71	74.98	78.26	78.66	78.65
This Work	✓	78.49	78.64	63.26	75.46	78.13	78.58
ResNet20, CIFAR100							
p-Norm ?	×	80.69	67.35	38.13	58.09	64.96	67.33
Channel-Norm?	×	80.69	71.26	52.59	66.05	70.08	71.26
Spike-Norm?	×	68.72	64.09	-	-	-	-
TSC?	×	68.72	68.18	-	-	58.42	65.27
RMP-SNN?	×	68.72	67.82	27.64	46.91	57.69	64.06
Opt.?	×	77.16	77.22	51.27	70.12	75.81	77.22
Calibration?	×	77.16	77.73	76.32	77.29	77.73	77.63
Burst ?	×	80.69	80.72	76.39	79.83	80.52	80.57
This Work	✓	80.69	80.58	71.98	78.58	80.25	80.57

Table 2: Summary of given hyperparameters on different network

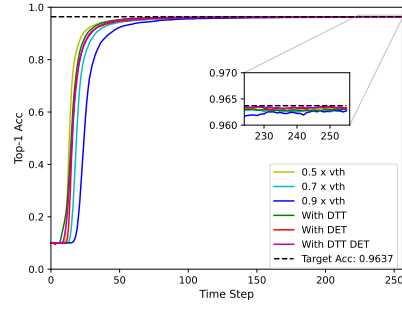
Symbol	VGG16	ResNet20
α	0.03	0.3
k_a	1	1
k_i	1.0	1.0
C	5.0	5.0
τ_{mp}	1	0.5
τ_{rd}	1	0.5

A Appendix

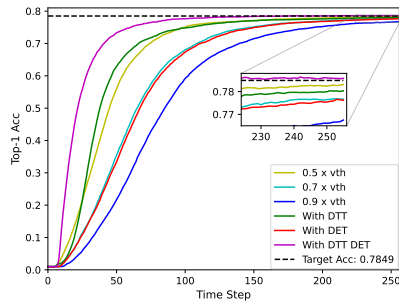
Optionally include extra information (complete proofs, additional experiments and plots) in the appendix. This section will often be part of the supplemental material.



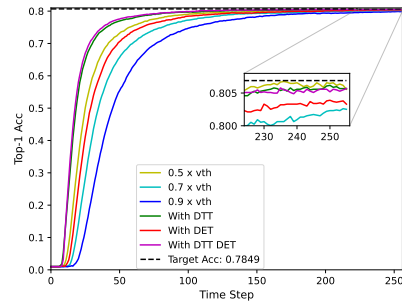
(a) pic1.



(b) pic2.



(c) pic3.



(d) pic4.

Figure 3: pics

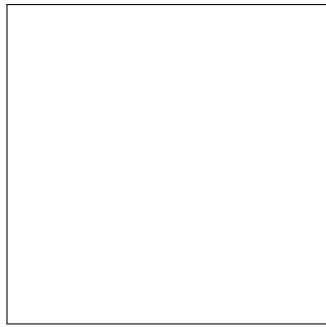


Figure 4: Spike count(efficiency) fig.

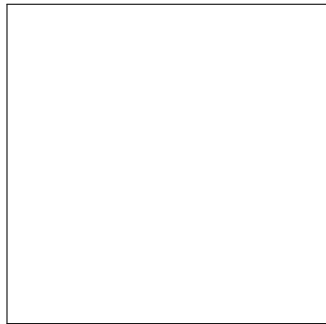


Figure 5: Vth vaires result vs inherent inference timestep.