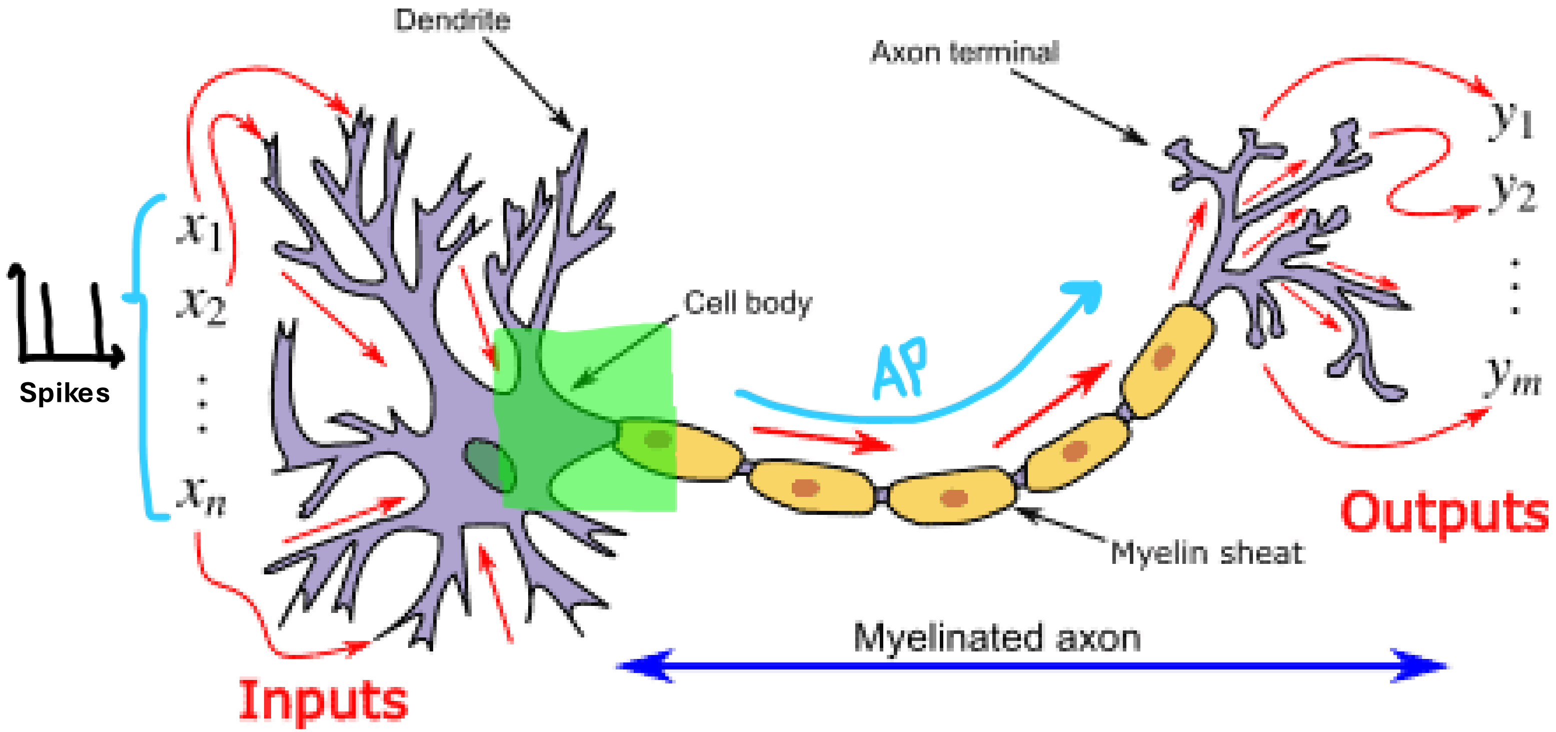
Rethinking the Membrane Dynamics and Optimization of SNNs.

- Modification of Spiking Neural Networks, for their better performance.

- Brief on SNN's
- Observations
- Modifications
- Results

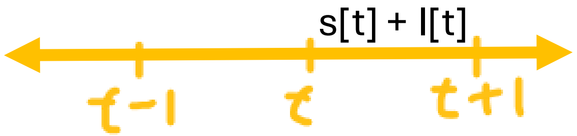
- Abhijeet Vikram
20221011 - IISER Pune

What are SNNs ? - ANN's that closely mimic biological neural networks

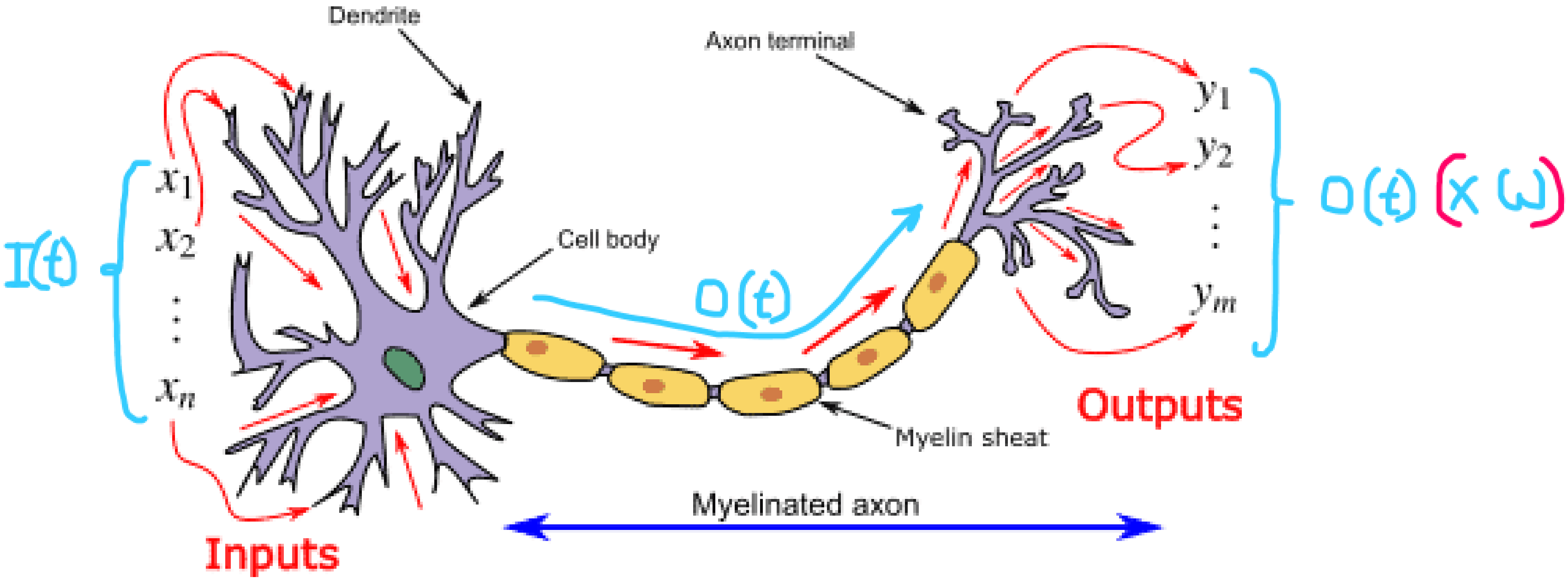


- Some initial membrane potential - IMP (ranges from -85 to –65 mV (~ 70 mV)) [1]
- The input from multiple neurons are accumulated at the axon hillock and
- A binary decision is made (action potential or not)
- If the accumulated input voltage is above a voltage threshold level a AP is generated. (Biological neuron V_{th}- ranges from –52 to –42 mV). [1]

- I [t] - Input potential
- h[t] - instantaneous membrane state
- s[t] - Step membrane state – stored membrane state
- o[t] - Absence [0] or Presence [1] of Action potential



s[0] = IMP

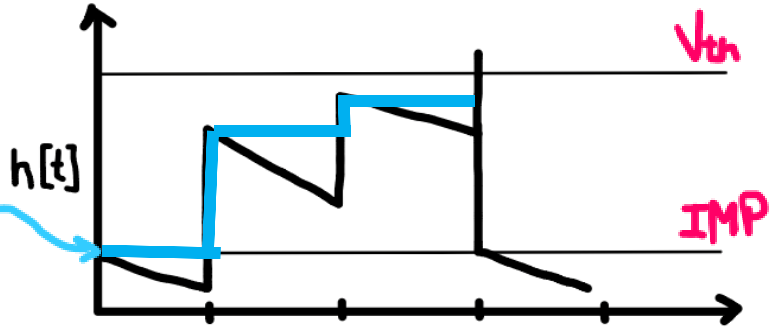


SNN - Leaky Integrate Fire neuron

Intermediate membrane state

Membrane potential decay coefficient

τ = 0 for IF

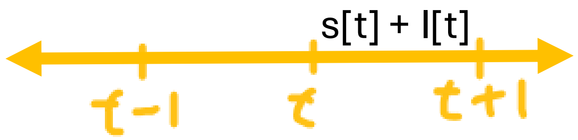


$$h[t] = (1 - \tau)s[t] + I[t], \quad h \in \mathbb{R}^{T \times N}, \quad I \in \mathbb{R}^{T \times N} \tag{1}$$

$$o[t] = h[t] > V_{th}, \quad o \in \{0, 1\}^{T \times N}, \quad V_{th} \in \mathbb{R} \tag{2}$$

$$s[t+1] = h[t] - o[t], \quad s \in \mathbb{R}^{T \times N}, \quad s[0] \in \{0\}^N \tag{3}$$

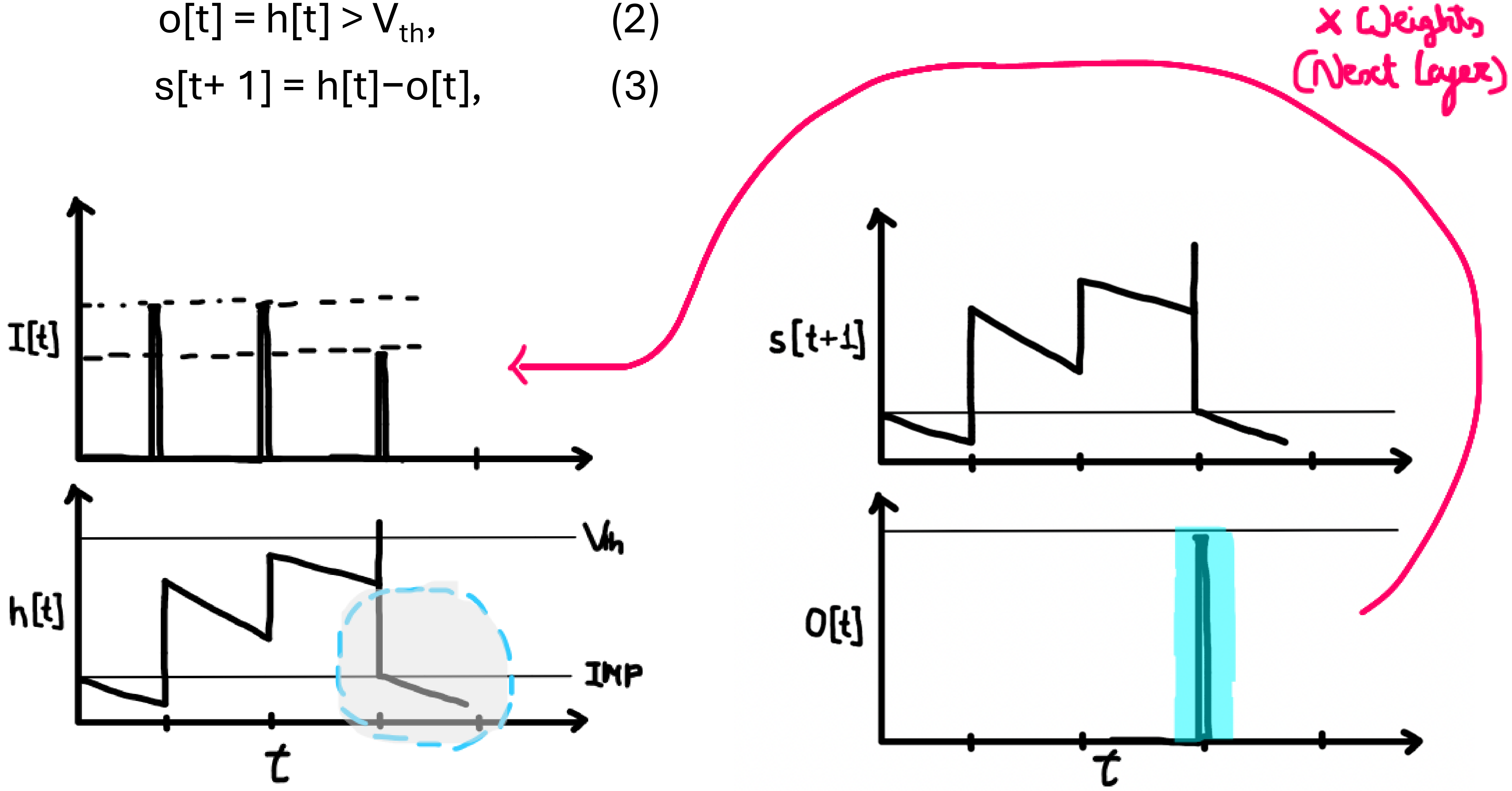
Stored membrane state



$$h[t] = (1-\tau)\, s[t] + I[t], \qquad (1)$$

$$o[t] = h[t] > V_{th}, \qquad (2)$$

$$s[t+1] = h[t] - o[t], \qquad (3)$$



Loss Functions – to Penalize the Errors

p – Predicted probability
q – Observed probability

Cross-Entropy Loss – Used for binary classification (between 0 and 1). [3]

$$L_{CE}(p,q) = -\frac{1}{N}\sum_{C=1}^{M} -p_i.\,log(q_i)$$

1

SDT Loss – Calculates the mean of outputs at each time step and then gets the loss

$$L_{SDT} = L_{CE}\left(\frac{1}{T}\sum_{t=0}^{T} y[t], y_{gt}\right)$$

0

TET Loss – Calculates the Loss at each time step and then averages it - Improves performance on neuromorphic tasks.

$$L_{TET} = \frac{1}{T}\sum_{t=0}^{T} L_{ce}(y[t], y_{gt})$$

Backward Pass

Weights are updated by Surrogate gradient Descent.

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial O[t]} \frac{\partial O[t]}{\partial h[t]} \frac{\partial h[t]}{\partial W}$$

I[t] - equation (1)

Replace it with **Surrogate Gradient**

A smooth, differentiable function that approximates a step:

$$\sigma(h) = \frac{1}{1+e^{-h}}$$

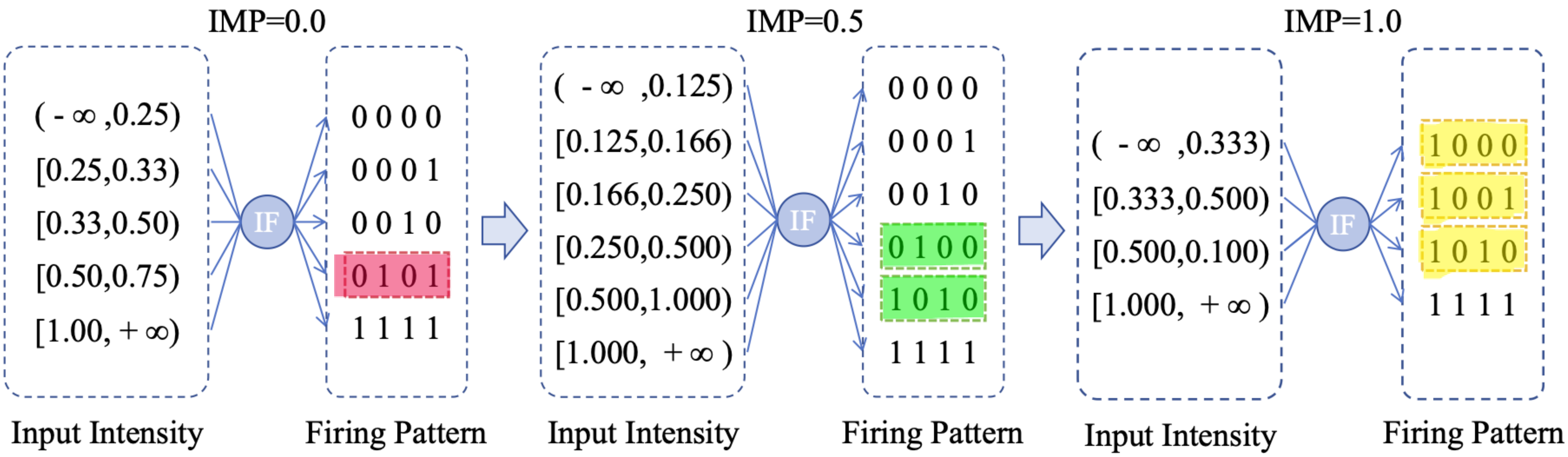
- **Sigmoid -** $\frac{\partial O}{\partial h}$)

$$W \leftarrow W - \eta . \frac{\partial L}{\partial W}$$

- **ReLu -** $\frac{\partial O[t]}{\partial h[t]} \approx \max(0, 1 - |h[t] - V_{TH}|)$

Observations

Observation 1 – New firing patterns can be generated by adjusting IMP with constant intensity input.



Observation 2 – New mapping of firing patterns was observed.

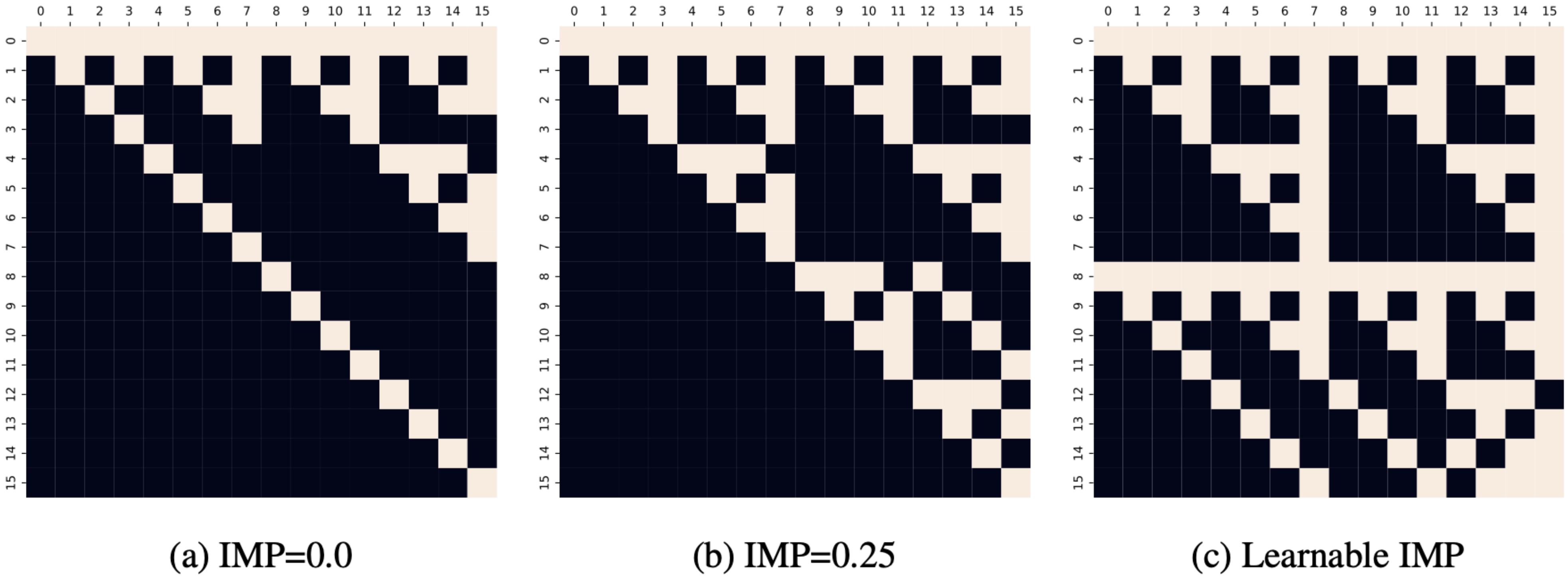


Figure 3: Pattern mapping of IF neuron over 4 time steps. The horizontal and vertical axes in the figure represent all possible spike patterns (16 total) that IF neurons may receive and emit. The white squares indicate that IF neuron can receive the spike pattern from the horizontal axis and emit a spike pattern on the vertical axis, known as pattern mapping.

Observation 3 – In Static tasks, accuracy of SNNs at each time step is sensitive to the current MP.

$$h[t] = (1-\tau)\, s[t] + I[t]$$

$$o[t] = h[t] > V_{th}$$

$$s[t+1] = h[t] - o[t]$$

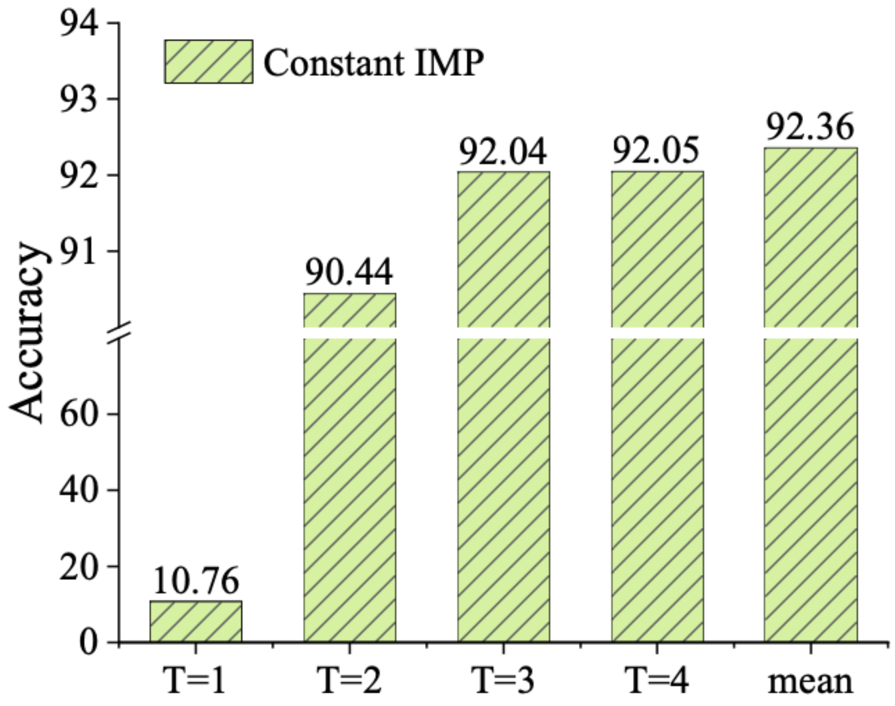


Figure 4: The test accuracy at each time step on the CIFAR10 dataset.

$$s[t+1], y[t] \leftarrow f(\, x[t], s[t], \theta \,),$$

Assuming a constant input intensity x for t = 1,2,…,T, and θ is constant for each time step.

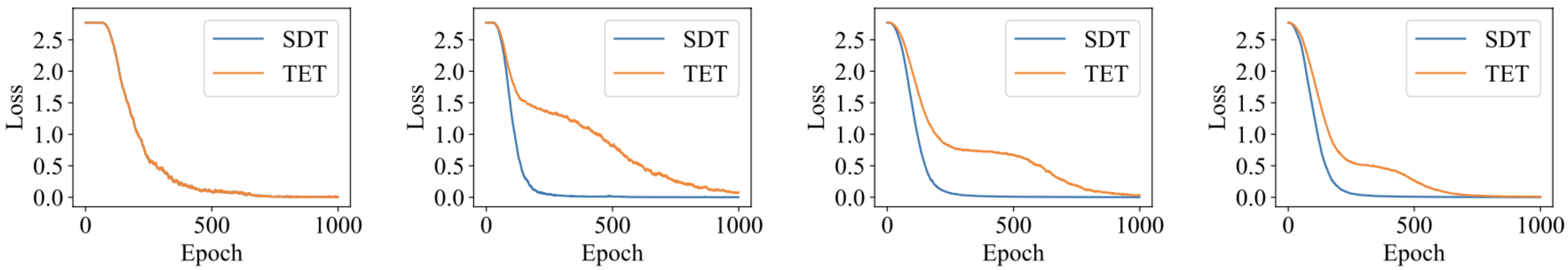
$$y[t] \leftarrow f(\, x, s[t], \theta \,),$$

Output solely depends on the state of the membrane potential.

Observation 4 – TET performs well on neuromorphic tasks but has slow convergence on static tasks.

Table 1: Test accuracy of TET and SDT on the static and neuromorphic datasets.

Loss	**Static Dataset**(SEW-R18)			**Neuromorphic Dataset**(VGG11)		
Function	CIFAR10/100	ImageNet100	ImageNet1k	CIFAR10DVS	DVSG128	NCaltech101
SDT Loss	**94.56/76.58**	**78.42**	**63.21**	84.3	98.26	85.78
TET Loss	94.33/76.40	77.80	62.92	**85.6**	**98.61**	**86.32**



(a) T=1 (b) T=2 (c) T=4 (d) T=6

Figure 5: The convergence speed of TET and SDT on the static data.

Modifications

Modification 1 - Learnable IMP – for IF

For IF Neuron - ($\tau = 0$)

$$h[t] = (1-\tau)\, s[t] + I[t]$$
$$o[t] = h[t] > V_{th}$$
$$s[t+1] = h[t] - o[t]$$

$$h[t] = \hat{s}[t] + x[t]$$

$\hat{s}[0] = 0$ is replaced by

$$\hat{s}[0] = \text{Uniform}(-\lambda, \lambda)$$

- λ is a hyper-parameter

[This finds the best **$\hat{s}$[0]** which has the minimum loss value]

Since the current implementation of SNNs requires memory allocation to store membrane potential states, replacing 0 with a trained IMP during the inference process will not incur additional computational overhead.

Modification 2 - LTS Loss function – for Static Tasks

- TET loss didn't work well on static tasks (Observation 4)

As seen in observation 3 & 4,

$$y[T] = f(s[T], x, \theta)$$

- Uses last time step (T) output to calculate the loss (Observation 3)

$$L_{LTS} = L_{ce}(\, y[T], y_{gt} \,)$$

- Retains the output after LTS (T) as the result of the entire model, ensures most "high quality" membrane potential without interference before the LTS (T).

Modification 3 - Label smooth TET Loss – for Neuromorphic Tasks

- TET Loss is actually used as,

$$L_{Total} = (1-\lambda)L_{TET} + \lambda . L_{Reg}$$

- Where,

$$L_{Reg} = \frac{1}{T}\sum_{t=1}^{T} L_{MSE}(y[t], \phi)$$

- ϕ denotes the target firing level.

1. L_{tet} - Measures how well the model's outputs (spike patterns) match the desired class labels at each time step.
2. L_{reg} - This controls the firing activity of the neurons to ensure the outputs are meaningful (not too high, too low, or constant). [Basically prevents overfitting]

- Flaw,
  - When L_{reg} = 0, the model may output a constant value (as per the highest weights by L_{tet}) at every time steps, rendering the model unable to perform the classification tasks.
  - Will not allow Loss to go to 0 theoretically.

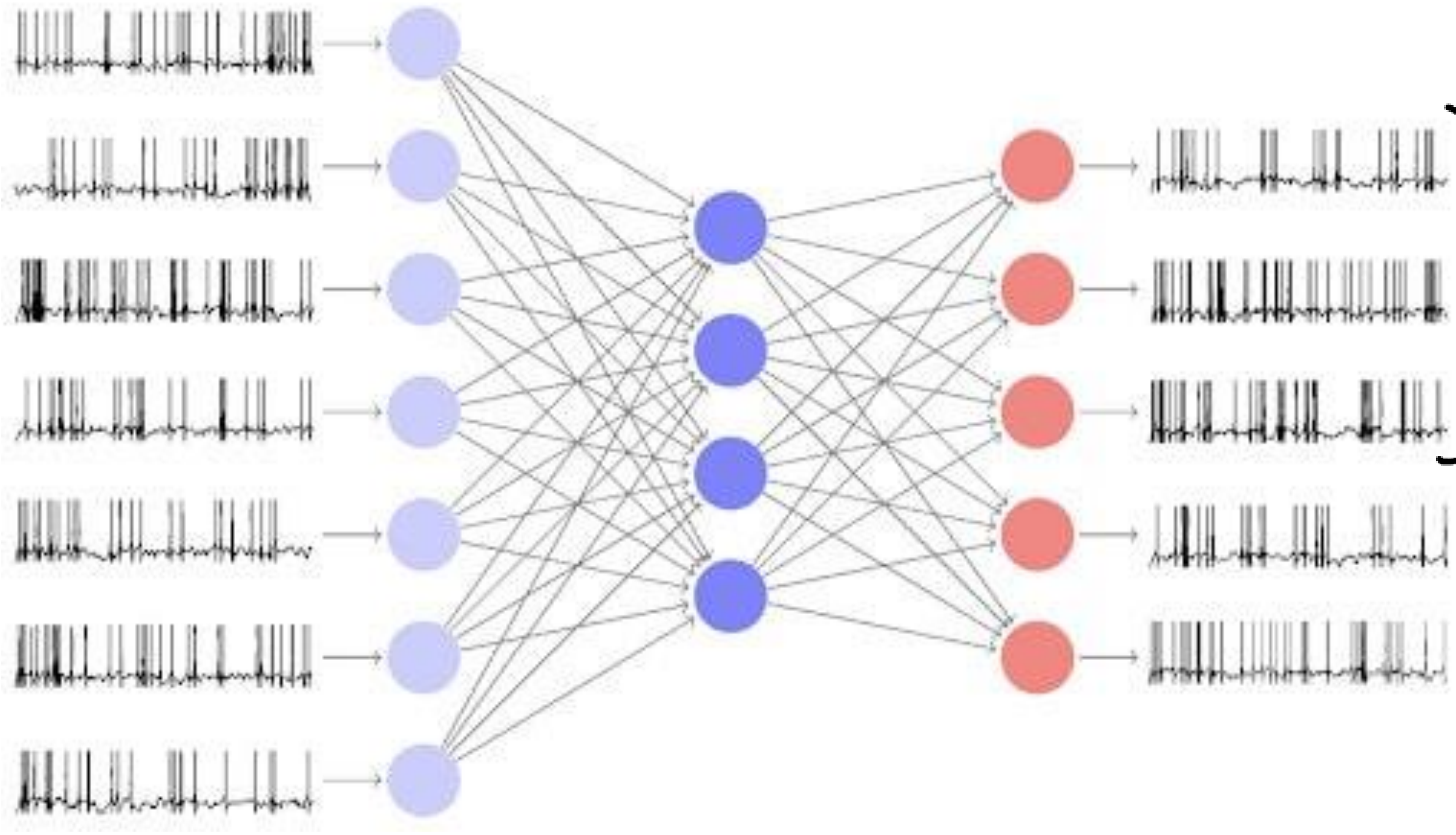
- Proposal -
  - Remove the L reg ,
  - $$L_{TET-S} = \frac{1}{T} \times \sum_{t=1}^{T} L_{ce}(f(s[t], x, \theta), \hat{y}_{gt})$$

Where,

$$\hat{y}_{gt} = (1 - \epsilon) y_{gt} + \frac{\epsilon}{K}$$

- **ε** represents the smoothing factor,
- K represents the number of classes.

  - $\hat{y}_{gt}$ converts hard labels like [1,0,0] to smooth labels [0.9333,0.0333,0.0333], when **ε** = 0.1 in 3 class problem.
  - Now this is compared to the output of the model after time step T and loss is calculated.



After 10 time steps:-

$$\begin{array}{c} 8\ - \\ 1\ - \\ 1\ - \end{array}\begin{bmatrix}0.8\\0.1\\0.1\end{bmatrix}$$

$$y_{gt} = \begin{bmatrix}1\\0\\0\end{bmatrix} \longrightarrow \hat{y}_{gt} = \begin{bmatrix}0.933\\0.033\\0.033\end{bmatrix}$$

$$L = L_{ce}\left(\begin{bmatrix}0.8\\0.1\\0.1\end{bmatrix}, \begin{bmatrix}0.933\\0.033\\0.033\end{bmatrix}\right)$$

Results

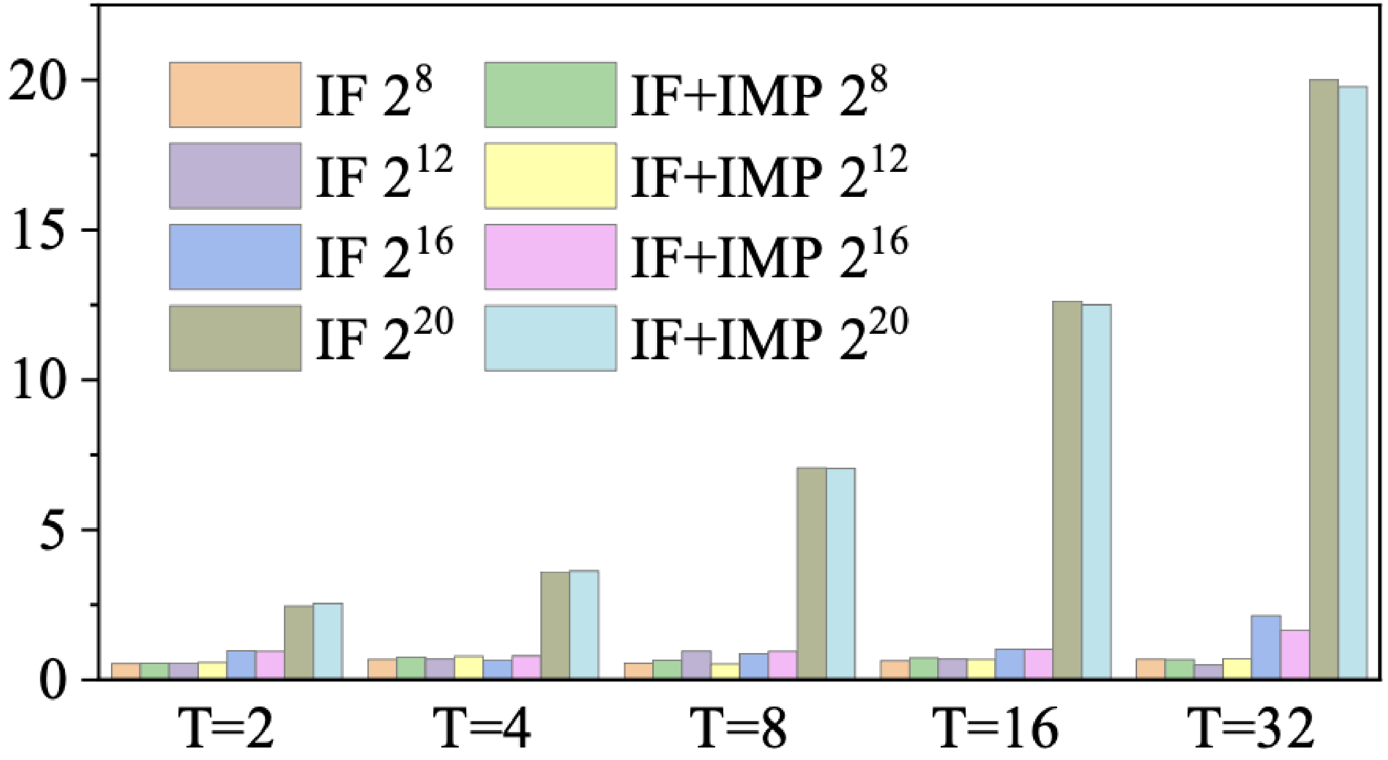
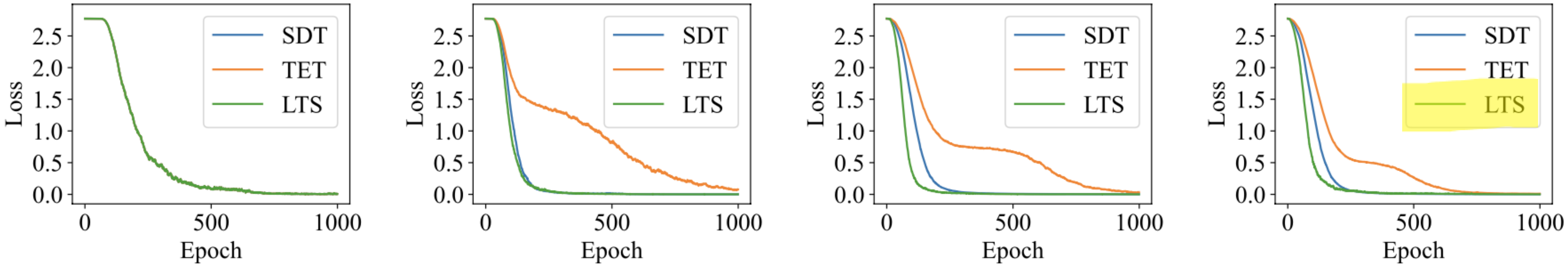


Figure 6: Execution time (ms) for the forward and backward pass of IF neurons, w/wo IMP.

about ±1.03% difference



(a) T=1 (b) T=2 (c) T=4 (d) T=6

Figure 7: Convergence speed of LTS, TET, and SDT on the static data.

Table 3: Comparison of our methods and other methods on the ImageNet1k dataset.

Method	Network Architecture	Reset	Params	Time Steps	Accuracy(%)
PSN[43]	SEW ResNet-18	×	11.69	4	67.63
	SEW ResNet-34	×	21.79	4	70.54
Dspike[50]	ResNet-34	✓	21.79	6	68.19
	VGG-16	✓	138.42	5	71.24
TET[34]	SEW ResNet-34	✓	21.79	4	68.00
TDBN[21]	ResNet-34	✓	21.79	6	67.05
TEBN[45]	SEW ResNet-34	✓	21.79	4	68.28
GLIF[41]	ResNet-34	✓	21.79	4	67.52
Spikformer[23]	Spikformer-6-512	✓	23.37	4	72.64
	Spikformer-8-512	✓	29.68	4	73.38
SEW ResNet[52]	SEW ResNet-18	✓	11.69	4	63.18
	SEW ResNet-34	✓	21.79	4	67.04
	SEW ResNet-50	✓	25.56	4	67.78
	SEW ResNet-101	✓	44.55	4	68.76
	SEW ResNet-152	✓	60.19	4	69.26
LTS	SEW ResNet-18	✓	11.69	4	64.33(+1.15)
	SEW ResNet-34	✓	21.79	4	68.10(+1.06)
	SEW ResNet-50	✓	25.56	4	71.24(+3.46)
IMP+LTS	SEW ResNet-18	✓	14.17	4	**65.38**(+2.20)
	SEW ResNet-34	✓	25.54	4	**68.90**(+1.86)
	SEW ResNet-50	✓	36.67	4	**71.83**(+4.05)

Table 2: Comparison of our methods and other SOTA methods on the neuromorphic datasets. Size refers to the input resolution of SNNs.

Dataset	Method	SNN Architecture	Size	Time Steps	Accuracy(%)
CIFAR10-DVS	GLIF[41]	Wide 7B Net	48	16	78.10
	NDA[49]	VGG	48	10	79.60
	TET[34]	VGG	48	10	83.17
	TEBN[45]	VGG	48	10	84.90
	PSN[43]	VGG	48	10	85.90
	IMP(ours)	VGG	48	10	**85.90**
	IMP+TET-S(ours)	VGG	48	10	**87.10**
	IMP+TET-S(ours)	VGG	48	8	**87.80**
	PLIF[40]	PLIF Net	128	20	74.80
	TDBN[21]	ResNet-19	128	10	67.80
	Dspike[50]	ResNet-18	128	10	75.40
	KLIF[51]	PLIF Net	128	15	70.90
	SEW ResNet[52]	Wide 7B Net	128	16	74.40
	Spikformer[23]	Spikformer	128	10	78.90
	Spikformer[23]	Spikformer	128	16	80.90
	NDA[49]	VGG	128	10	81.70
	IMP(ours)	VGG	128	16	**86.30**
	IMP+TET-S(ours)	VGG	128	16	**87.00**
N-Caltech101	NDA[49]	VGG	48	10	78.20
	EventMix[53]	ResNet18	48	10	79.47
	ESP[54]	SNN7-LIFB	48	10	81.74
	TCJA[55]	TCJA-SNN	48	10	82.50
	TKS[56]	VGG-TKS	48	10	84.10
	IMP(ours)	VGG	48	10	**84.68**
	IMP+TET-S(ours)	VGG	48	10	**85.01**
	EventDrop[57]	VGG	128	10	74.04
	NDA[49]	VGG	128	16	83.70
	EventRPG[58]	VGG	128	10	85.62
	STR[59]	VGG	128	10	85.91
	IMP(ours)	VGG	128	16	**86.12**
	IMP+TET-S(ours)	VGG	128	16	**87.86**

Rethinking the Membrane Dynamics and Optimization Objectives of Spiking Neural Networks

Hangchi Shen[1,2], **Qian Zheng**[3,4], **Huamin Wang**[1,2,*], **Gang Pan**[3,4]
[1]College of Artificial Intelligence, Southwest University
[2]Chongqing Key Laboratory of Brain Inspired Computing and Intelligent Chips
[3]The State Key Lab of Brain-Machine Intelligence, Zhejiang University
[4]College of Computer Science and Technology, Zhejiang University
stephen1998@email.swu.edu.cn, hmwang@swu.edu.cn,
{qianzheng, gpan}@zju.edu.cn

Abstract

Despite spiking neural networks (SNNs) have demonstrated notable energy efficiency across various fields, the limited firing patterns of spiking neurons within fixed time steps restrict the expression of information, which impedes further improvement of SNN performance. In addition, current implementations of SNNs typically consider the firing rate or average membrane potential of the last layer as the output, lacking exploration of other possibilities. In this paper, we identify that the limited spike patterns of spiking neurons stem from the initial membrane poten-

References

- Rethinking the Membrane Dynamics and Optimization Objectives of Spiking Neural Networks - Hangchi Shen, Qian Zheng, Huamin Wang, Gang Pan et al.

[1] **A Threshold Equation for Action Potential Initiation -** Jonathan Platkiewicz, Romain Brette

[2] Convergence of Nanotechnology and Machine Learning: The State of the Art, Challenges, and Perspectives - by **Arnav Tripathy, Akshata Y. Patne**

[3] https://en.wikipedia.org/wiki/Cross-entropy

- https://cnvrg.io/spiking-neural-networks
- https://medium.com/@nghihuynh_37300/understanding-loss-functions-for-classification-81c19ee72c2a