# Towards Neuromorphic Computing on Edge: A Survey on Efficient Techniques for Spiking Neural Networks

***Abstract*— Recent advancements in Spiking Neural Networks (SNNs) for machine learning have established them as an energy-efficient alternative to conventional Artificial Neural Networks (ANNs). However, training and deploying a deep SNN comes with complications related to propagation of various errors including vanishing gradients, ANN-to-SNN activation mismatch, and loss of transmitted information in the spiking patterns. This work surveys techniques to mitigate those errors and to increase the expressive capacity of the networks on the ground of spiking and neuronal dynamics. Latency vs accuracy results for the reviewed methods are reported on standard benchmarks such as ImageNet and CIFAR. Additionally, the temporal metric $T$ and its relationship to model efficiency is discussed.**

## I. Introduction

Efficiency improvement in SNNs owes to the sparsity of firing activity in spiking neurons. However, training SNNs introduces two major challenges. First, the non-differentiability of spike events requires surrogate gradients results in approximation bias during backpropagation due to the discrepancy between the surrogate function and the spike dynamics [1]. Second, extensive computation and latency costs are associated with basic implementations of SNNs due to Backpropagation Through Time (BPTT) over multiple time steps [2]. By default, a large window of simulated time steps is needed to compensate for approximation error and to increase the learning capacity of SNN models. Recent work has shown that strategies for encoding and processing temporal information can substantially reduce the number of required time steps [3–7]. Additionally, by improving the biological plausibility of spiking neuronal models, their learning capacity and performance are improved [8–13]. In this work, recent methods in both spiking and neuronal dynamics are surveyed. The performance is reported against ImageNet, CIFAR-10, and CIFAR-100 datasets given their popularity as image classification benchmarks for SNNs.

## II. Background

SNNs leverage both temporal and spatial dynamics to process information through discrete action potential (spike) events. An SNN's behavior relies on the choice of neuron models, synaptic connectivity patterns, and the interactions between them. These components determine the network expressive capacity a constant temporal window. Neuron models are based on the electric circuit analogy of a biological neuron's membrane potential. A common neuronal model is the Leaky Integrate-and-Fire (LIF) which represents the dynamics of membrane cell gates by a capacitor and resistor [14]. The differential equation representing the membrane potential $u$ is shown in **(1)** where $I_R$ is simply $u(t)/R$. Therefore, the equation can be re-arranged to **(2)** with a time constant $\tau_m = RC$, where $RC$ is the leak resistance and membrane capacitance respectively [14], which represents a critical hyperparameter in the modeling techniques to come in section IV.

$$C\frac{du(t)}{dt} = -I_R + I(t) \tag{1}$$

$$\tau_m\frac{du(t)}{dt} = -u(t) + RI(t) \tag{2}$$

### A. Preliminary

The LIF model have been widely adopted in SNN research due to its simplicity while still capturing the core spiking phenomenon. The discrete version of **(2)**, via forward-Euler approximation, used in digital simulation, can be expressed by **(3)** for $\alpha = e^{-\Delta t/\tau_m} \approx 1 - \Delta t/\tau_m$ as the decay factor of the membrane potential, and $w_k$ is the weights for dendrite $k$ given a spike $S_k[n] \in \{0, 1\}$ [15]. This enables recursive updates of the membrane potential given a current and a previous time step until a spike event is triggered.

$$u[n] = \alpha u[n-1] + \sum_k w_k S_k[n] \tag{3}$$

Since $u(t)$ resets after emitting a spike, the general dynamics representation of a spiking neuron is shown in **(4)** where for dendrite input $X_t$ and previous potential $V_{t-1}$, the pre-spike potential $H_t$ is calculated, using **(3)**, in the case of the LIF model. Then, $S_t$ denotes the output to the heaviside step function $\Theta[n]$ in **(5)**, and the output potential $V_t$ updates the potential in the next time-step in case of no spike, or resets to $V_{reset}$, typically zero, in case of a spike.

$$\begin{aligned} H_t &= f(V_{t-1}, X_t) \\ S_t &= \Theta(H_t - V_{th}) \\ V_t &= H_t(1 - S_t) + V_{reset}S_t \end{aligned} \tag{4}$$

$$\Theta[n] = \begin{cases} 1, n \geq 0 \\ 0, otherwise \end{cases} \tag{5}$$

## III. Spiking Dynamics

ANN-SNN conversion methods are favored for high-performance deep neural networks. The learned weights of common ReLU-activated models can be transferred as-is from an ANN to a rate-encoded SNN. In deep networks, the propagation of approximation errors significantly degrades converted SNN performance [16]. Conversion-Aware Training (CAT) in [3] pre-conditions the network to learn the temporal and fire-rate behaviors of the SNN during ANN training, requiring minimal post-conversion correction. Error-Aware Conversion in [4] avoids re-training by applying post-conversion layer-wise calibration to close the gap between continuous ANN activations and their discrete spiking counterparts. The discrepancy between the ANN activation and SNN patterns also leads to information loss when small $T$ values are used. The work in [5] identifies this error as out-of-bound spikes within a time window $T$ because the spike train tend to be randomly clustered. An example is a spike train of $\{0,0,0,1,1\}$ for rate $2/5$ that loses spikes when $T < 5$, and accumulation of this loss over layers severely degrades accuracy. Averaging Integrate-and-Fire Spike Generation is used to produce an evenly distributed spike train, e.g., converting $\{0,0,0,1,1\}$ to $\{0,1,0,1,0\}$, making it more robust to reduced $T$ values.

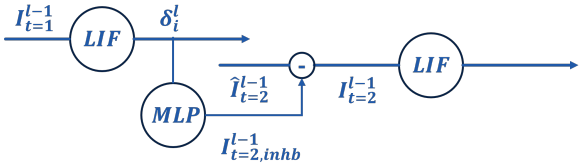


Fig. 1: The CPT topology between two spiking events

In direct training, the capacity of a typical SNN to carry spike signals temporally is limited because a spike depends only on the previous changes in the membrane potential, and not on past spike patterns [7]. To utilize the temporal relationship between spikes, a method called Combine the Previous Timestep (CPT) is used. **Fig. 1** shows how temporal correlation is established using the previous spike as a projected inhibitory current to the next spiking event. The membrane update for the spiking neuron in (**3**) is modified to (**6**) where $I^{(l)}_{ihb}$ is a ReLU-activated MLP projection within the same neuron but at consecutive timesteps.

$$u^{(l)}[n] = (1 - \frac{1}{\tau})u^{(l)}[n-1] + I^{(l)}_{in}[n] - I^{(l)}_{ihb}[n] \quad (6)$$

Another way to reduce the temporal window is to have $T$ itself adaptive per layer. The work in [6] showed that different layers within the network exhibit variable vulnerability to reduction in $T$ depending on their activity. Measurement of layer activity through fired spikes per $T$ indicates that layers with high activity need richer temporal resolution, therefore higher $T$ value. The adaptive temporal window works with both rate and temporal encoding, and for directly trained SNNs, a regularization term (**7**) is used to encourage lowered layer-wise time steps.

$$\lambda_r = \sum_{l=1}^{L} \frac{\sum_{l=1}^{L} \text{Activity}(l) \times \#Param(l)}{\text{Activity}(l)} \quad (7)$$

## IV. Neuronal Dynamics

A basic extension to LIF is the Parametric LIF (PLIF) model. PLIF allows the time constant $\tau$ of the membrane to be learnable. To avoid instabilities during training, $\tau$ is not optimized directly but instead through a function $k(a)$ defined in (**8**). In this case, $a$ is the learnable parameter where $\tau = 1/k(a) \in (1, +\infty)$. The pre-spike potential $H_t$ in (**4**) is defined in (**9**) [15].

$$k(a) = \frac{1}{1 + e^{-a}} \quad (8)$$

$$H_t = u_{t-1} + k(a)(-(u_{t-1} - V_{reset}) + X_t) \quad (9)$$

Other learnable parameters introduced in [9] include the firing threshold, which is made adaptive in both time and space by modifying the spike function $S_t$ in (**4**) into (**10**) for all neurons $I$ in a given layer, and temporal and spatial factors $F_t$ and $F_s$ respectively.

$$S_I[t] = \Theta(u_I[t] - V_{th} \cdot F_t(\phi[t]) \odot F_s(t, I)) \quad (10)$$

Another form of implementing adaptive threshold is presented in [10] that employs two types of temporal varying thresholds inspired by existing properties of the nervous system. First, a Dynamic Tracking Threshold (DTT) that increases as the membrane potential increases, acting as a high-pass filter to prevent small-voltage fluctuations from contributing to the output spike. A Dynamic Evoked Threshold (DET) that depends rather on the input's rate of change, making the neuron more responsive to rapid input updates. Both techniques allow faster information transmission between neurons by eliminating spike redundancy. Additionally, a dynamic firing threshold is presented in [12] for near-lossless ANN-SNN conversion frameworks for very low temporal window $T$. The technique, named Group Integrate-and-Fire (GIF) introduces multiple thresholds, as multipliers of a base threshold $V_{th}$, for a shared membrane potential per neuron. Only one threshold fires at a time at the maximum value below the shared membrane potential. The potential update is modified as in (**11**) where $k$ is the firing threshold index.

$$V_t = V_{t-1} - k \cdot V_{th} \quad (11)$$

In [13], value-dependent threshold is used through a Multi-Level Firing (MLF) technique that targets the problem of vanishing gradients in direct training frameworks. The authors discussed that the gradient

of the spike function $\partial O/\partial H$ for equation (**4**) is near zero except for $H_t$ values near the constant threshold $V_{th}$ when surrogate gradient functions are used. This effect is demonstrated in **Fig. 2** which MLF mitigates by replacing a single threshold per neuron with multiple parallel thresholds and membrane potentials.

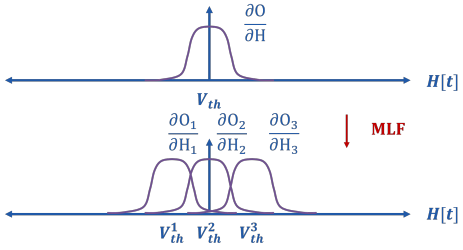


Fig. 2: Surrogate gradient of single vs multiple $V_{th}$

The output at a given time step is the union of spikes at the thresholds of which the independent membrane potentials had reached. This widens the active range of membrane potentials where the gradient value is not saturated, preventing excessive gradient vanishing during backpropagation. Another approach in [11] works on reshaping the distribution of the input to the heaviside step function $S_i$, as in (**12**), to be closer to either zero or the firing threshold.

$$S_t = \Theta(\hat{H}_t - V_{th}) \quad \text{for} \quad \hat{H}_t = \phi(H_t) \tag{12}$$

The process is called Membrane Potential Rectification in which the original membrane potential is not affected, but only fed to a non-linear function $\phi$ before being processed through the step function $\Theta_t$. This approach also uses a soft-reset mechanism where $V_t = H_t - O_t$ instead of resetting to zero after a spike event. The soft reset retains residual potential information from previous spiking activity, enriching the model's expressive capacity, and the rectification process reducing quantization error of the membrane potential by minimizing its distance from $\{0, 1\}$.

## V. Meta Analysis

**Table I** summarizes accuracy vs temporal window $T$ for the methods surveyed, revealing several trends including latency-accuracy trade-offs, superior performance of parameter calibration approaches, and reduced $T$ requirements for adaptive neurons compared to pure spike encoding.

Notably, accuracy gains saturate beyond a specific increase in $T$ depending on the method used, indicating an optimal $T^*$ value after which diminishing performance returns occur for $T > T^*$. This behavior is explained in [17] which shows that quantization error maps to a logarithmic signal-to-error ratio. This relationship is described in (**13**) between the input signal $x$ and its quantized version $\hat{x}$.

$$SNR = 10log(\frac{E[x^2]}{E[(x-\hat{x})^2]}) \approx 20log(T + C) \tag{13}$$

So, each doubling of $T$ yields only a fixed improvement. It is also notable that for the same method and network architecture, the $T^*$ value is lower for easier tasks (CIFAR10) and greater for harder tasks (ImageNet). For example, in the GIF method, $T$ increase becomes insignificant after values 2, 8, and 16 for CIFAR10, CIFAR100, and ImageNet benchmarks respectively.

For converted SNNs, post-conversion calibration through Fine-Tuning (FT) or Two-stage calibration shows 69-70 % ImageNet accuracy at $T = 32$ whereas 512-1024 timesteps needed for CAT-SNN to get similar results without calibration. Neuronal adaptation seems to be superior at much lower latency across all tested benchmarks. For example, the GIF model can achieve the same ImageNet performance at just 2 timesteps.

### A. Energy Efficiency Calculation

*1) Theoretical Analysis:* The effective temporal window needed for low-latency energy-efficient SNN inference depends on an optimum firing threshold, membrane leak, and network weights as optimizable parameters [18]. A common approach to quantify inference energy reduction is to assign energy cost per 32-bit Multiplication-Accumulation (MAC) or Accumulation (AC) operations within the network. Typically, a standard 45nm [6, 10, 18] CMOS reference process is used with 4.6 pJ and 0.9 pJ energy cost for MAC and AC operations respectively [19]. SNN operations are purely addition (AC) except for the first layer where the input is encoded [10]. Energy reduction ratio $E_{ratio}$ of SNN to ANN is obtained through (**14**) for a number of operations $OP^l$ per layer $l$.

$$E_{ratio} = \frac{OP_{SNN}}{OP_{ANN}} = \frac{4.6OP^1_{ANN} + \sum_{l=2}^{L} 0.9OP^l_{SNN}}{\sum_{l=1}^{L} 4.6OP^l_{ANN}} \tag{14}$$

The number of operations $OP^l_{SNN}$ is the sum of spike rate per layer $l$ times $OP^l_{ANN}$. The spike rate is the total number of spike events $\Theta$ in a layer divided by the number of neurons $M^l$ in that layer [10, 18, 20] as defined in (**15**).

$$\begin{aligned} OP_{SNN} &= \text{Spike Rate} \times OP_{ANN} \\ \text{Spike Rate}_l &= \tfrac{1}{M^l} \textstyle\sum_{M^l} \sum_{T} \Theta_{m,t} \end{aligned} \tag{15}$$

*2) Empirical Refinement:* Memory access operations are a crucial factor in energy consumption calculation that is typically discarded in many studies [21]. The energy consumption of a LIF model is not limited to the AC operations but also include reading synaptic weights $E_{R_{weight}}$ and membrane state $E_{R_{state}}$ from memory and updating them ($E_{W_{state}}$). The previous factors have to be calculated for the total number of synapsis $N_{syn}$ given an average spikes per synapse $N_{spikes/syn}$ through the first term in (**16**) [21].

TABLE I: Summary of spiking and neuronal dynamics methods and their performance in terms of accuracy against different temporal windows $T$.

| Group | Method / Detail | T | ImageNet (%) | CIFAR10 (%) | CIFAR100 (%) |
|---|---|---|---|---|---|
| **Spiking Dynamics** | **CAT-SNN: Temporal Encoding** [3] | 20 | – | 92.21 | 69.58 |
| | Conversion, w/out FT | 40 | – | 93.03 | 71.40 |
| | | 80 | – | 93.36 | 72.14 |
| | **CAT-SNN: Rate Encoding** [3] | 512 | 68.98 | 86.40 | 54.62 |
| | [VGG16] | 1024 | 71.68 | 94.50 | 66.06 |
| | | 2048 | 72.43 | 95.08 | 71.09 |
| | | 2500 | 72.47 | 95.13 | 71.75 |
| | **ANN counterpart** | – | 70.69 | 94.81 | 71.37 |
| | **Error-Aware Conversion** [4] | 4 | – | 94.75 | 73.60 |
| | [VGG16] | 8 | – | 95.39 | 76.11 |
| | One-/Two-stage | 32 | 69.04 | 95.53 | 77.83 |
| | | 128 | 74.11 | 95.60 | 77.93 |
| | | 256 | 74.75 | – | – |
| | | 2048 | 75.32 | – | – |
| | **ANN counterpart** | – | 75.36 | 95.60 | 77.93 |
| | **Low-Latency Conversion** [5] | 8 | – | 91.55 | 64.79 |
| | w/ FT [VGG16] | 16 | – | 92.33 | 66.72 |
| | | 32 | 70.43 | – | – |
| | **TEAS** [6] | 2500/45.5 | - | 92.48/91.43 | - |
| | (w/ FT) [VGG16] | 2500/71.3 | - | - | 70.97/70.95 |
| | | 2500/76.9 | 73.46/73.41 | - | - |
| | **ANN counterpart** | – | 73.49 | – | 71.22 |
| | **CPT** [7] Conversion/Direct [VGG16] | 8 | – | 94.57 | 76.35 |
| **Neuronal Dynamics** | **Spatio-Temporal Threshold Adaptation** [9] | 1 | - | 96.18 | 79.23 |
| | [ResNet-19] | 2 | - | 96.66 | 80.35 |
| | | 4 | 63.67 | 96.84 | 81.20 |
| | **MSAT: Adaptive Threshold** [10] | 16 | – | 73.42 | – |
| | | 32 | – | 93.87 | – |
| | | 64 | – | 95.20 | – |
| | | 256 | – | – | 78.50 |
| | | 2045 | 74.93 | – | – |
| | **ANN counterpart** | – | 75.16 | 95.45 | 78.49 |
| | **SRIF** [11] | 2 | - | 94.44 | 75.56 |
| | Membrane Modification [ResNet18/19] | 4 | 64.78 | 96.27 | 78.42 |
| | | 6 | – | 96.49 | 79.51 |
| | **ANN counterpart** | – | – | 96.29 | 78.61 |
| | **GIF: Sub-neurons** [12] | 1 | 64.85 | 94.96 | 72.67 |
| | [VGG16] | 2 | 69.89 | 95.51 | 74.11 |
| | | 4 | 72.96 | 95.58 | 75.14 |
| | | 8 | 74.41 | 95.71 | 75.50 |
| | | 16 | 74.86 | 95.77 | 75.64 |
| | | 32 | 74.99 | 95.79 | 75.76 |
| | | 64 | 74.95 | 95.77 | 75.78 |
| | **ANN counterpart** | – | 74.94 | 95.76 | 75.82 |
| | **MLF: Sub-neurons** [13] | 4 | – | 94.25 | – |

$$
\begin{aligned}
E_{LIF} &= \text{synaptic\_events} + \text{neurons\_updates} \\
&= N_{syn} \times N_{spikes/syn} \\
&\times (E_{R_{weight}} + E_{R_{state}} + E_{W_{state}} + E_{AC}) \\
&+ N_{neur} \times T \times (E_{R_{state}} + E_{W_{state}} + E_{MAC})
\end{aligned} \tag{16}
$$

In the second term, MAC operations are used to account for current integration to the membrane potential per neuron. Another framework in [22] had used synaptic events to define a device-agnostic metric by introducing the Synaptic Activity Ratio (SAR), where SAR $= N_{spikes}/N_{syn}$, and the adjusted ratio is SAR$/E_{ratio}$.

## VI. Limitations and Future Work

Recent efforts to accelerate SNN inference through temporal window reduction have followed notably independent paths. Future work is needed to integrate these orthogonally developed techniques. Moreover, most recent methodologies follow simplified theoretical analysis for energy-gain calculations that is purely based on the number of operations. Existing work that addresses the cost of memory access lacks hardware-agnostic formulation. To address these shortcomings, a unified benchmarking framework is needed including the use of standardized datasets and accounting for hardware variations. Realizing this goal will likely require reproducible replications of the experimental setup across reviewed methodologies.

## VII. Conclusion

In this work, techniques towards efficient implementation of Spiking Neural Networks are explored. Five recent methodologies of each spiking and neuronal dynamics are compared for latency vs accuracy. As an important efficiency metric, the temporal window $T$ reflects a greater capacity of the modeled dynamics to transfer information while omitting redundancies in spiking patterns. The results have shown a great overall reduction in latency compared to basic rate encoding technique while delivering a competitive performance relative to their ANN counterparts.

## References


[1] Z. Wang, R. Jiang, S. Lian, R. Yan, and H. Tang, "Adaptive smoothing gradient learning for spiking neural networks," in *Proceedings of the 40th International Conference on Machine Learning*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., ser. Proceedings of Machine Learning Research, vol. 202, PMLR, Jul. 2023, pp. 35 798–35 816. [Online]. Available: https://proceedings.mlr.press/v202/wang23j.html.

[2] Q. Meng, M. Xiao, S. Yan, Y. Wang, Z. Lin, and Z.-Q. Luo, "Towards memory- and time-efficient backpropagation for training spiking neural networks," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 6143–6153. DOI: 10.1109/ICCV51070.2023.00567.

[3] D. Lew and J. Park, "Cat snn: Conversion aware training for high accuracy and hardware friendly spiking neural networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 13, no. 2, pp. 512–524, 2025. DOI: 10.1109/TETC.2024.3435135.

[4] Y. Li, S. Deng, X. Dong, and S. Gu, "Error-Aware Conversion from ANN to SNN via Post-training Parameter Calibration," *International Journal of Computer Vision*, vol. 132, no. 9, pp. 3586–3609, Sep. 2024, ISSN: 1573-1405. DOI: 10.1007/s11263-024-02046-2. [Online]. Available: https://doi.org/10.1007/s11263-024-02046-2.

[5] Z. Yan, K. Tang, J. Zhou, and W.-F. Wong, "Low latency conversion of artificial neural network models to rate-encoded spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2025. DOI: 10.1109/TNNLS.2025.3526374.

[6] F. Liu, H. Li, N. Yang, Z. Wang, T. Yang, and L. Jiang, "Teas: Exploiting spiking activity for temporal-wise adaptive spiking neural networks," in *Proceedings of the 29th Asia and South Pacific Design Automation Conference*, ser. ASPDAC '24, Incheon, Republic of Korea: IEEE Press, 2024, pp. 842–847, ISBN: 9798350393545. DOI: 10.1109/ASP-DAC58780.2024.10473984.

[7] Q. Xia, Y. Yu, Z. Chang, B. Hui, and H. Luo, "Cpt-snn: A spiking neural network that can combine the previous timestep," *Neurocomputing*, vol. 640, p. 130 253, 2025, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2025.130253. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231225009257.

[8] S. Lian, J. Shen, Q. Liu, Z. Wang, R. Yan, and H. Tang, "Learnable surrogate gradient for direct training spiking neural networks," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, E. Elkind, Ed., Main Track, International Joint Conferences on Artificial Intelligence Organization, Aug. 2023, pp. 3002–3010. DOI: 10.24963/ijcai.2023/335. [Online]. Available: https://doi.org/10.24963/ijcai.2023/335.

[9] J. Fu *et al.*, "Adaptation and learning of spatio-temporal thresholds in spiking neural networks," *Neurocomputing*, vol. 644, p. 130 423, 2025, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2025.130423. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231225010951.

[10] X. He, Y. Li, D. Zhao, Q. Kong, and Y. Zeng, "MSAT: Biologically inspired multistage adaptive threshold for conversion of spiking neural networks," *Neural Computing and Applications*, vol. 36, no. 15, pp. 8531–8547, May 1, 2024, ISSN: 1433-3058. DOI: 10.1007/s00521-024-09529-w. [Online]. Available: https://doi.org/10.1007/s00521-024-09529-w.

[11] Y. Guo *et al.*, "Reducing information loss for spiking neural networks," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham: Springer Nature Switzerland, 2022, pp. 36–52, ISBN: 978-3-031-20083-0.

[12] Z. Ye, W. Zeng, Y. Chen, L. Zhang, J. Xiao, and I. King, "Group if units with membrane potential sharing for high-accuracy low-latency spiking neural networks," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–8. DOI: 10.1109/IJCNN60899.2024.10650174.

[13] L. Feng, Q. Liu, H. Tang, D. Ma, and G. Pan, "Multi-level firing with spiking ds-resnet: Enabling better and deeper directly-trained spiking neural networks," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, L. D. Raedt, Ed., Main Track, International Joint Conferences on Artificial Intelligence Organization, Jul. 2022, pp. 2471–2477. DOI: 10.24963/ijcai.2022/343. [Online]. Available: https://doi.org/10.24963/ijcai.2022/343.

[14] W. Gerstner and W. M. Kistler, *Spiking Neuron Models : Single Neurons, Populations, Plasticity - Record details - EBSCOhost Research Databases*, 2002. (visited on 07/09/2025).

[15] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, "Incorporating Learnable

Membrane Time Constant to Enhance Learning of Spiking Neural Networks," en, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 2641–2651, ISBN: 978-1-6654-2812-5. DOI: 10.1109/ICCV48922.2021.00266. [Online]. Available: https://ieeexplore.ieee.org/document/9711070/ (visited on 07/30/2025).

[16] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in Neuroscience*, vol. Volume 11 - 2017, 2017, ISSN: 1662-453X. DOI: 10.3389/fnins.2017.00682. [Online]. Available: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2017.00682.

[17] A. Castagnetti, A. Pegatoquet, and B. Miramond, "Trainable quantization for speedy spiking neural networks," *Frontiers in Neuroscience*, vol. Volume 17 - 2023, 2023, ISSN: 1662-453X. DOI: 10.3389/fnins.2023.1154241. [Online]. Available: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2023.1154241.

[18] N. Rathi and K. 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*, vol. 34, no. 6, pp. 3174–3182, 2023. DOI: 10.1109/TNNLS.2021.3111897.

[19] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 10–14. DOI: 10.1109/ISSCC.2014.6757323.

[20] S. Hwang and J. Kung, "One-spike snn: Single-spike phase coding with base manipulation for ann-to-snn conversion loss minimization," *IEEE Transactions on Emerging Topics in Computing*, vol. 13, no. 1, pp. 162–172, 2025. DOI: 10.1109/TETC.2024.3386893.

[21] M. Dampfhoffer, T. Mesquida, A. Valentian, and L. Anghel, "Are snns really more energy-efficient than anns? an in-depth hardware-aware study," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 3, pp. 731–741, 2023. DOI: 10.1109/TETCI.2022.3214509.

[22] E. Lemaire, B. Miramond, S. Bilavarn, H. Saoud, and N. Abderrahmane, "Synaptic activity and hardware footprint of spiking neural networks in digital neuromorphic systems," *ACM Trans. Embed. Comput. Syst.*, vol. 21, no. 6, Dec. 2022, ISSN: 1539-9087. DOI: 10.1145/3520133. [Online]. Available: https://doi-org.khalifa.idm.oclc.org/10.1145/3520133.