ALL IN ONE TIMESTEP: ENHANCING SPARSITY AND ENERGY EFFICIENCY IN MULTI-LEVEL SPIKING NEURAL NETWORKS

Andrea Castagnetti
Université Côte d'Azur
LEAT
Sophia Antipolis, France
andrea.castagnetti@univ-cotedazur.fr

Alain Pegatoquet
Université Côte d'Azur
LEAT
Sophia Antipolis, France
alain.pegatoquet@univ-cotedazur.fr

Benoît Miramond
Université Côte d'Azur
LEAT
Sophia Antipolis, France
benoit.miramond@univ-cotedazur.fr

ABSTRACT

Spiking Neural Networks (SNNs) are one of the most promising bio-inspired neural networks models and have drawn increasing attention in recent years. The event-driven communication mechanism of SNNs allows for sparse and theoretically low-power operations on dedicated neuromorphic hardware. However, the binary nature of instantaneous spikes also leads to considerable information loss in SNNs, resulting in accuracy degradation. To address this issue, we propose a multi-level spiking neuron model able to provide both low-quantization error and minimal inference latency while approaching the performance of full precision Artificial Neural Networks (ANNs). Experimental results with popular network architectures and datasets, show that multi-level spiking neurons provide better information compression, allowing therefore a reduction in latency without performance loss. When compared to binary SNNs on image classification scenarios, multi-level SNNs indeed allow reducing by 2 to 3 times the energy consumption depending on the number of quantization intervals. On neuromorphic data, our approach allows us to drastically reduce the inference latency to 1 timestep, which corresponds to a compression factor of 10 compared to previously published results. At the architectural level, we propose a new residual architecture that we call Sparse-ResNet. Through a careful analysis of the spikes propagation in residual connections we highlight a spike avalanche effect, that affects most spiking residual architectures. Using our Sparse-ResNet architecture, we can provide state-of-the-art accuracy results in image classification while reducing by more than 20% the network activity compared to the previous spiking ResNets.

Keywords Spiking Neural Networks · multi-level spikes · low-power artificial intelligence

1 Introduction

Spiking Neural Networks (SNNs) are considered as the the third generation of Artificial Neural Networks (ANNs). SNNs focus on the encoding and the processing of the information using binary and asynchronous signals known as spikes. This computing paradigm, inspired by the messaging mechanism used by biological neurons, is thought to be amongst the sources of the energy efficiency of the brain. However, the binary nature of spikes also leads to considerable information loss, i.e. quantization errors, causing performance degradation compared to ANNs using floating-point operations. SNNs quantization error can be reduced by increasing the number of timesteps, therefore the latency over the network. However, with a longer conversion time, more spikes are generated, thus increasing the energy consumption as well. Several techniques have been proposed to minimize both the quantization error and the latency of SNNs. These approaches can be either applied to the ANN-to-SNN conversion or directly during the SNN training using

the surrogate gradient (SG) method. In Li et al. [2022] and Rathi and Roy [2021] the authors adopt an ANN-to-SNN conversion scheme where the firing threshold of the spiking neurons is optimized after conversion to better match the distribution of the membrane potential. In Castagnetti et al. [2023a] a SNN is trained using SG and the Adaptive Integrate-and-Fire (ATIF) neuron. This ATIF neuron has been proposed as an alternative to the original Integrate and Fire (IF) neuron, the firing threshold (V_{th}) being a learnable parameter rather than an empirical hyper-parameter. In Guo et al. [2022a], the authors also use SG to train the SNN, but introduce a distribution loss to shift the membrane potential distribution into the conversion range of the spiking neurons. With these approaches, requiring only few timesteps, it is possible to get SNNs with almost no accuracy loss when compared to the equivalent ANNs. As an example, a SNN trained on 4 timesteps can achieve an accuracy only 1% below the equivalent non-quantized, i.e. Floating Point (FP), ANNs on CIFAR-10 Li et al. [2022]. To further decrease the latency, recent approaches propose to go beyond binary spikes and introduce multi-level spiking neurons. This mechanism expands the output of spiking neurons from a single bit to multiple bits, thus increasing the information that can be transmitted at each timestep. In Xiao et al. [2024] for instance, it has been shown that using a 4-level spiking neuron and one timestep, the same level of accuracy of a well optimized binary SNNs trained on 4 timesteps Li et al. [2022] can be achieved. Most of the previous works only focus on the SNN latency. However, it has been shown Lemaire et al. [2023], Dampfhoffer et al. [2023] that besides the latency, another important parameter that has to be optimized to improve the energy efficiency is the sparsity of the network, in other words the number of spikes, either binary or multi-level, generated during the inference. In this work we propose to enhance the sparsity of SNN from two points of view. First at the neuronal level, where we introduce a multi-level spiking neuron model that can seamlessly deal with both time and the spike value to reduce the quantization error. Then at the architectural level, where a novel spiking ResNet architecture is proposed. Through a careful analysis of the spike propagation in the residual layers of the architecture we highlight an effect that we call *spike avalanche*. Events coming from the direct and residual paths sum up and create peaks of activity that can potentially decrease the sparsity of the SNN. Finally we compare binary and multi-level SNNs from the energy-efficiency point of view. Our analysis based on the metric proposed in Lemaire et al. [2023], is intended to be independent from low-level implementation choices. Moreover, our energy estimation takes into account not only the synaptic operations but also all the memory accesses that take place in an event-driven execution scenario on a neuromorphic hardware accelerator. We therefore provide energy estimation results that are closer to what could be reasonably obtained on a real hardware implementation. This approach is essential to avoid overestimating the energy gains.

The main contributions of our work are listed below:

- We propose a multi-level model of an IF spiking neuron compatible with SNN direct training using SG.
- We train SNNs on different image classification problems and characterize the corresponding spiking activity. We provide state-of-the-art accuracy results on CIFAR-10/100 image datasets, using only 1 timestep while reducing the energy consumption by a factor of 3 compared to an equivalent ANN.
- On neuromorphic data we provide a new state of the art latency/accuracy results for the CIFAR-10-DVS dataset. Here we approach the previously published best accuracies, which are obtained with 10 or more timesteps, while compressing the latency to 1 timestep.
- We introduce Sparse-ResNet, a new spiking residual architecture. Our experimental results show that Sparse-ResNet can achieve accuracy equivalent to the state of the art SEW-ResNet Fang et al. [2021a] while reducing the spiking activity by more than 20% on the CIFAR-10 dataset.

2 Related Works

2.1 SNN training

Training large SNNs models on modern complex datasets has been the subject of an increasing number of studies in the recent years. Two methods are typically used to obtain an efficient SNN: ANN-SNN conversion or direct training with surrogate gradients. The ANN-SNN conversion method is based on the idea that firing rates of spiking neurons should match the activations of analog, i.e. Floating Points, neurons. Earlier works proposed to directly convert from FP ANN models to SNNs. Diehl et al. [2015] uses weight normalization and spiking neuron threshold balancing to minimize the conversion loss. In Rueckauer et al. [2017] and Han et al. [2020] the authors propose to adopt reset-by-subtraction, i.e. soft-reset, spiking neurons to lower the conversion loss. Moreover they show that the SNN accuracy can be improved by using real inputs instead of Poisson spike trains. In Stanojevic et al. [2023] the ANN is converted in an equivalent SNN that uses a Time-To-First-Spike (TTFS) coding method to propagate the information between the spiking neurons. However, directly converting a FP ANN to an SNN typically leads to a very high latency solution. Nevertheless, it has been recently shown in Wang et al. [2024] that it is possible to achieve high accuracy at low latency using a specific neuron initialization, called data-based neuronal initialization (DNI) and by taking into account the asynchronous transmission characteristics of SNNs.

Another line of works has shown that to achieve very low latency and high accuracy at the same time, the quantization process has to be taken into account directly when training the ANN model. In Li et al. [2022] the ANN model is first trained using a Quantization-Aware-Training (QAT) procedure, and then converted into an equivalent SNN. Similarly Ding et al. [2021] replaces the ReLU activation functions with the proposed quantization clip-floor-shift (QCFS) activation. The ANN is trained using stochastic gradient descent and a straight-through estimator Bengio et al. [2013] is used to approximate the derivative of the quantized activation function. In Yang et al. [2025] the conversion method proposed in Ding et al. [2021] is extended to Channel-wise quantization.

Instead of converting from a pre-trained ANN, the direct training method Neftci et al. [2019] proposes to directly train the SNN using standard back-propagation. A surrogate function is used, during gradient back-propagation, to replace the binary non-linearity of spiking neurons. This allows gradient flowing thus making back-propagation possible in the spiking domain. Intuitively, direct training of SNNs with surrrogate gradient share some similarities with the quantized ANN to SNN conversion methods discussed above. The spike information compression is analyzed in Castagnetti et al. [2023a,b], Li et al. [2022], where the quantization scheme is described as a function of the neuron parameters, i.e. V_{th} and the number of timesteps, i.e. T. A spiking neuron can actually be considered as a uniform quantizer that discretizes its input into $T + 1$ quantization intervals. The direct training method leverages this fact and thus optimizes the quantization mapping while seeking to minimize the task loss. Several works improve the accuracy/latency trade-off by using techniques to relieve the gradient exploding/vanishing problem incurred by the back-propagation-through-time (BPTT) algorithm. Guo et al. [2022a] introduces a membrane potential distribution loss to penalize undesired membrane potential shift thus alleviating the gradient vanishing or explosion. Li et al. [2024] proposes Masked Surrogate Gradient (MSG) to keep the balance between the gradient mismatch, caused by a wide surrogate function, and the vanishing gradient risks incurred by a narrow surrogate function.

2.2 Quantization noise in spiking neurons

Several other works try to improve the accuracy/latency trade-off of SNNs by minimizing the quantization noise introduced by the spiking neurons. Guo et al. [2022b] introduces an Information-Maximization loss to minimize the information-loss caused by the quantization from the full-precision tensor, i.e. the input of the spiking neurons, to the spike tensor. Castagnetti et al. [2023a] proposes to minimize the quantization error by learning the firing threshold (V_{th}) of the spiking neuron during training. In Chen et al. [2024] a time-based coding scheme called At-most-two-spikes Exponential Coding (AEC) is introduced. AEC neurons employ quantization-compensating spikes to improve coding accuracy and capacity. Finally, Chen et al. [2024] proposes a neuron model that exploits the option of temporal coding with spike patterns, where the timing of a spike transmits extra information.

2.3 Multi-level spiking neurons

To further decrease the latency, recent approaches propose to go beyond binary spikes and introduce multi-level spiking neurons. This mechanism expands the output of spiking neurons from a single bit to multiple bits, thus increasing the information that can be transmitted at each timestep. In Guo et al. [2023] the authors propose a ternary spiking neuron that transmits information with {-1, 0, 1} spikes. A similar ternary coding is proposed in Sun et al. [2022] but only with positive values. Moreover, in multi-level spiking neurons the spike is extended to a fixed-point unsigned binary number Feng et al. [2022], Xiao et al. [2024]. In Xiao et al. [2024], it has been shown that using a 4-level spiking neuron and one timestep, the same level of accuracy of a well optimized binary SNNs trained on 4 timesteps Li et al. [2022] can be achieved. Wang and Zhang [2024] introduces a spiking neuron model with multiple thresholds, called MT-SNN that can generate multiple spike sequences. MT-SNN can achieve a higher accuracy with fewer timesteps since the precision loss caused by the shorter latency, i.e. T, is restored by the information encoding obtained using multiple thresholds.

2.4 Event-based processing, sparsity and energy consumption estimation

SNNs hold the promise of lower energy consumption in embedded hardware due to their sparse event-driven spike-based computations compared to traditional ANNs. In SNNs the neuron computation consists in accumulating the spikes, weighted by the synaptic connections into the membrane potential and firing an output spikes when the membrane potential exceeds V_{th}. In its simplest form the spiking neuron operations only require accumulate (AC) operations while artificial neurons in ANNs perform multiply-and-accumulate (MAC) operations between input activations and weights. Most research works, considering either binary Wang et al. [2024], Kim et al. [2020], Rathi and Roy [2021] or multi-level SNN Guo et al. [2023], Xiao et al. [2024], Wang and Zhang [2024], posits that the energy efficiency of SNNs is associated with the energy savings obtained by replacing MACs with ACCs for synaptic operations coupled with the inherent sparsity of the SNN models. However, this point of view has been questioned in recent works Lemaire et al. [2023], Dampfhoffer et al. [2023] where it has been shown that the energy cost of synaptic operations is negligible

compared to the one of transferring data, i.e. synaptic weights and neuron potentials, from or to the memory. As an example, Dampfhoffer et al. [2023] estimates that for an SNN with a VGG16 topology the energy consumed by the synaptic operations represents only 1% of the total energy consumption. Moreover, these works point out that the main advantage of SNNs compared to ANNs (on digital hardware) comes primarily from exploiting the sparsity of spikes and not from the replacement of MAC by ACC operations Dampfhoffer et al. [2023]. Based on these results we believe that to improve the SNNs energy/accuracy trade-off, the classical latency/accuracy trade-off has to be revisited. We then propose to analyze the SNNs behavior by focusing on the sparsity as well as its native ability to process data in the time domain. In the following we first analyze how information is coded in binary spiking neuron. Then we introduce a multi-level spiking neuron model compatible with direct training using SG. We characterize the quantization noise of the proposed neuron model. We then move to network level and analyze the propagation of spikes in residual layers. We show that without a careful handling of residual connections in SNNs, the sparsity of residual networks can be compromised, thus increasing the energy consumption.

3 Preliminaries

In this section, we introduce the Integrate-and-Fire (IF) binary spiking model and describe the encoding and decoding process used in SNNs.

3.1 Information coding with spiking neurons

An IF spiking neuron implements the ReLU non-linearity ($max(0, x)$) and discretizes its input signal into spikes. The following equations describe the Integrate-and-Fire (IF) neuron model with soft-reset:

$$H^l(t) = V^l(t-1) + i^l(t) \tag{1}$$

$$i^l(t) = z^{l-1}(t)\, W^l + b^l \tag{2}$$

$$z^l(t) = \Theta(H^l(t) - V_{th}) \tag{3}$$

$$V^l(t) = H^l(t)\,(1 - z^l(t)) + (H^l(t) - V_{th})\, z^l(t) \tag{4}$$

The previous equations describe the behavior of the IF neuron at the output of the layer l of the SNN. The membrane potential after the input integration and after the reset operation are represented by $H^l(t)$ and $V^l(t)$ respectively. The binary spiking output at time t is represented by $z^l(t)$. The input of the spiking neuron, $i^l(t)$, is the sum of the weighted spikes coming from the layer $l-1$ plus a constant bias. As shown in Eq. 1 and 2, the synaptic operations are implemented with ACC in a spiking neuron. Eq. 3 and 4 describe the generation of a spike and the soft-reset operation respectively. The function $\Theta(\cdot)$ represents the Heaviside step function used in the forward pass. The surrogate of $\Theta(\cdot)$, that is $\Theta'(x) = \sigma'(x)$, is used during the backward pass. In this work we use the sigmoid $\sigma(x, \alpha) = \frac{1}{1-e^{-\alpha x}}$ as surrogate function. Where x represents the membrane voltage of the spiking neuron and α is an hyper-parameter that modulates the width of the surrogate derivative.

The integrate-and-fire operations described by the previous equations, are repeated through T timesteps. The output y_s^l is then rate-decoded as follows:

$$y_s^l = \frac{1}{T} \sum_{t=1}^{T} z^l(t) \tag{5}$$

The Eq. 5 represents the firing-rate of the neuron. In this scheme the information is coded by the number of spikes generated by a spiking neuron over a fixed duration of time T. Moreover, since $z^l(t)$ is a binary variable, the decoded output y_s^l is quantized into $T+1$ different values. The number of quantization intervals determines the distortion incurred during the quantization process. For a binary spiking neuron the quantization noise can be reduced by increasing T, i.e. the latency of the SNN. However, augmenting the latency leads to an increase of the SNN energy consumption as more spikes are generated. There is therefore a trade-off between the amount of quantization noise and the energy efficiency of the SNN. Our objective is therefore to reduce the quantization noise without increasing T. The next section proposes a solution for that problem by introducing the multi-level spiking neuron model.

4 Methods

4.1 Multi-level spiking neuron model

The neuron model and the associated decoding scheme are shown in Fig. 1. This model is characterized by two

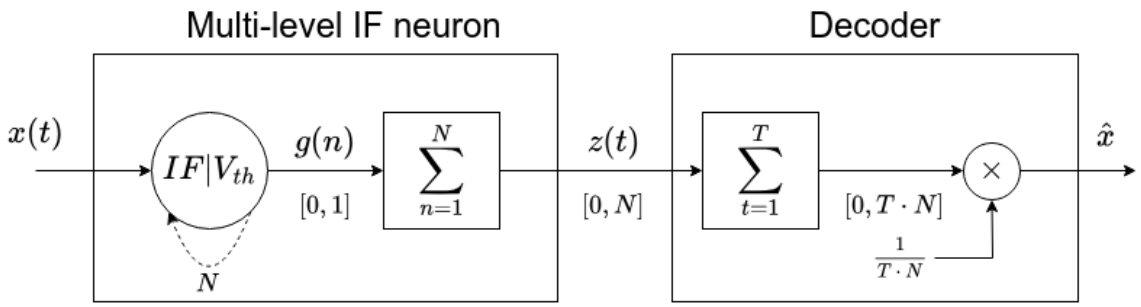

Figure 1: Multi-level IF neuron model and the associated decoding scheme.

parameters: the firing threshold V_{th} and the number of levels of a spike, that we note N. The multi-level spiking neuron is composed by a standard IF neuron, described by the Eq. 1-4 given in Sec. 3. The spike generation process is divided into two phases. At each timestep the IF neuron is iteratively charged, N times, with the current input $x(t)$. Through this operation, the current timestep is indeed subdivided into N intervals, that we call *micro-timesteps*. Then, the internal membrane potential of the IF neuron is discharged through the N micro-timesteps to generate $g(n)$ binary spikes. The value of the spike emitted at the end of the timestep is computed by summing up all the binary spikes $g(n)$ generated through the discharge process:

$$z(t) = \sum_{n=1}^{N} g(n) \tag{6}$$

It is worth noticing that only the valued spike $z(t)$ is communicated to the next layer. The binary spikes $g(n)$ are internal to the spiking neuron and do not contribute to the overall activity of the network.

4.2 Quantization error analysis

The spiking neuron operation described in Sec. 4.1 can be repeated through an arbitrarily number of T timesteps, where the output at each timesteps is a valued spike $z(t) \in [0, N]$. Moreover, the model can be reverted to an IF binary spiking neuron by setting $N = 1$. In that case, the discharge process is executed only once and the output at each timesteps is a binary spike $z(t) \in [0, 1]$. Finally, the output of the spiking neuron can be decoded as follows:

$$\hat{x} = \frac{1}{N\,T} \sum_{t=1}^{T} z(t) \tag{7}$$

Overall, the conversion process leads to a discretization of the input x. When the input of the neuron is a constant signal, that is $x(t) = x$, we can observe that the quantization function of the multi-level spiking neuron, shown in Fig. 2, has exactly $N \times T + 1$ quantization levels. For the binary spiking neuron, that is when $N = 1$, there are instead $T + 1$ uniform quantization intervals Castagnetti et al. [2023a], Li et al. [2022]. A multi-level spiking neuron can therefore communicate at each timestep $\log_2(N)$ bits while the binary spiking neuron is limited to only 1 bit of information. With multi-level spiking neurons it is thus possible to decrease the quantization error without increasing the latency of the SNN, thus achieving a better information compression without hindering, in theory, the computational efficiency of the SNN.

4.3 Spikes propagation in residual connections

Residual learning and shortcuts connections have been evidenced as an important way of deepening ANNs architectures He et al. [2016] and are now widely adopted by almost any state-of-the-art neural network Sandler et al. [2019], Tan and Le [2020], Dosovitskiy et al. [2021]. Particularly, ResNets architectures He et al. [2016] have been converted into SNNs in several previous works Hu et al. [2023a], Fang et al. [2021a], Hu et al. [2023b]. Conversion issues from ANNs have been studied in Hu et al. [2023a], while Fang et al. [2021a] focused on the difficulty of implementing identity mapping with spiking neurons. Finally in Hu et al. [2023b] the authors explored the vanishing/exploding gradient problems that arise when using the direct training method. The main residual architectures for SNNs are shown in Fig. 3. In the original Spiking ResNet proposed in Hu et al. [2023a] the only architectural modification consists in replacing the ReLU non-linearities with spiking neurons. Spiking ResNet converted from ANN achieves state-of-the-art accuracy on nearly all datasets, albeit with high latencies Han et al. [2020]. On the other hand, several challenges have to be addressed to achieve the same accuracy levels with Spiking ResNets trained directly in the spike domain. MS-ResNet

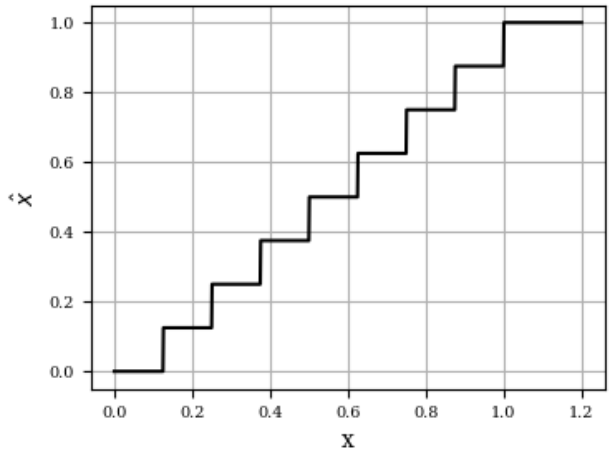

Figure 2: Quantization function of a multi-level IF with soft-reset ($T = 2$, $N = 4$, $V_{th} = 1.0$). We can observe that the output of the neuron has exactly $N \times T + 1$ quantization levels and the output saturates when the input equals V_{th}.

Hu et al. [2023b] addresses this problem by implementing the residual connections between the membrane potentials of the spiking neurons. However, by directly connecting the membrane potential of the spiking neurons, the MS-ResNets cannot be considered a fully event-based network. On the other hand, SEW-ResNets Fang et al. [2021a] proposes to aggregate the spikes at the summation point of the residual connection. Using this topology, SEW-ResNets can indeed achieve comparable results with standard ResNets while maintaining the event-based communication scheme. However, as the spikes are sequentially transferred from one layer to the next, the spikes that come from the residual connection are added to the flow of spikes coming from the direct path. This effect, that we call *spike avalanche* is illustrated in Fig. 4. In this simplified example, we consider that the number of input binary spikes in SEW ResNets, that we call γ_b, is not modified when processed by a convolutional layer. As shown, the γ_b spikes that the first residual block receives as input are propagated through both the direct and the residual path. Therefore, the number of spikes at the output of the first residual block is doubled. In consequence, the following layer receives $2 \times \gamma_b$ spikes at its input. Similarly, the two flows of spikes add up at the output of the second residual block that generates $4 \times \gamma_b$ spikes. The number of spikes increases exponentially as the network becomes deeper, thus creating an *avalanche* of spikes that have to be processed by the final layers of the network. To overcome this problem we propose a modification of the ResNet architecture for SNNs, that we call Sparse-ResNet shown in Fig. 3 (d).

4.4 Sparse-ResNet

Sparse-ResNet makes use of the multi-level spiking neuron both in the direct path like SEW ResNet and after the summation point as Spiking ResNet. The multi-level neuron that is placed after the summation point is called a *barrier neuron*. Its goal is to maintain a low quantization error at the output of the summation point while limiting the number of spikes that are generated. However, as pointed out in Fang et al. [2021a], placing a spiking neuron just after the summation point causes a performance degradation for the directly trained spiking ResNets. The vanishing/exploding gradients, induced by the surrogate function of the spiking neuron, are indeed the main causes of the performance degradation observed in Fang et al. [2021a]. SEW-ResNets solve this problem by removing the spiking neuron after the summation point. Therefore, since $S_l = O_l$ and the g function is the addition, the gradients that are back-propagated both in the direct, A_l and the residual connections, R_l, are equal to the incoming gradient $\frac{\partial L}{\partial O_l}$:

$$\frac{\partial L}{\partial A_l} = \frac{\partial L}{\partial R_l} = \frac{\partial L}{\partial O_l} \tag{8}$$

In the Sparse-ResNet model instead, the incoming gradient first goes through the surrogate gradient function of the spiking neuron before being propagated through the direct and residual connections:

$$\frac{\partial L}{\partial A_l} = \frac{\partial L}{\partial R_l} = \frac{\partial L}{\partial O_l}\frac{\partial O_l}{\partial S_l} = \frac{\partial L}{\partial O_l}\sigma'(S_l - V_{th}) \tag{9}$$

Commonly used surrogate functions Neftci et al. [2019], like the one used in our study and shown in Fig. 5, are highly peaked at V_{th} and drops quickly as the membrane potential of the neuron, V_{mem} moves away from the threshold voltage.

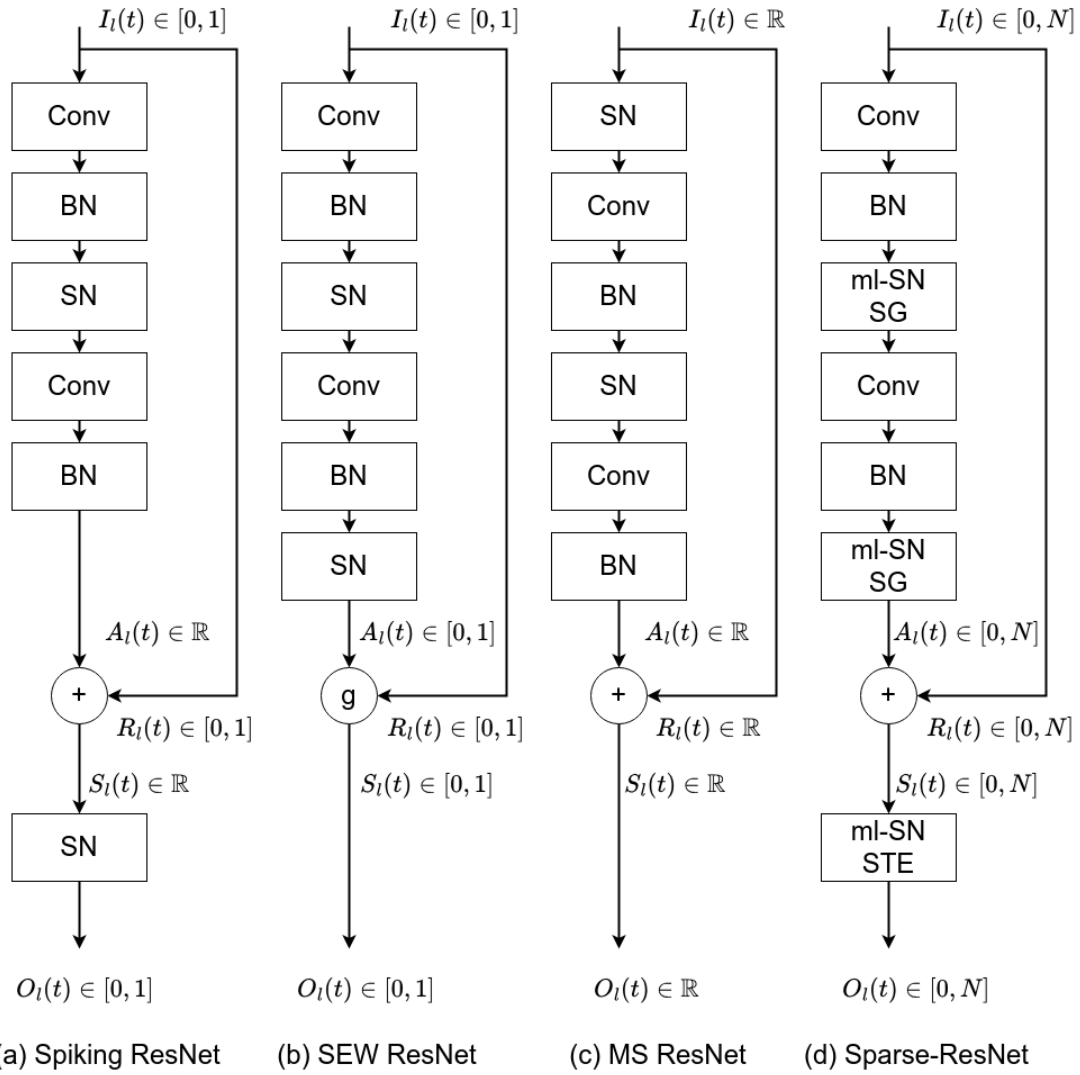

Figure 3: Residual blocks in (a) Spiking ResNets Hu et al. [2023a], (b) SEW-ResNets Fang et al. [2021a], (c) MS-ResNets Hu et al. [2023b] and the proposed Sparse-ResNets. *SN* stands for binary spiking neuron while *ml-SN/STE* and *ml-SN/SG* means a multi-level spiking neuron with a Straight-Through Estimator and a surrogate backward function respectively.

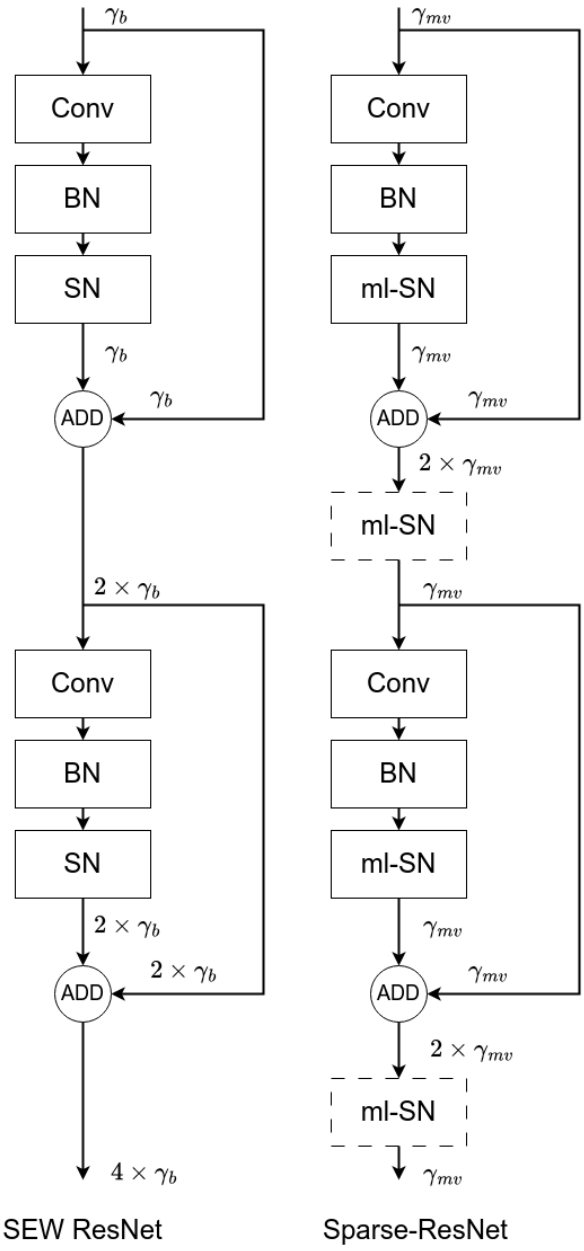

Figure 4: Spike propagation in residual connections. Here γ_b and γ_{mv} represent the amount of binary and multi-level spikes at the input of the first residual connection of the network. Only one Convolutional block is represented in the direct path. In the SEW-ResNets Fang et al. [2021a], the flow of spikes is added at the summation points, thus creating an exponential increase of the number of spikes that have to be processed by deeper layers. Using the barrier neuron (dotted in the figure), the Sparse-ResNets can limit the propagation of spikes without hindering the representation capacity of the network.

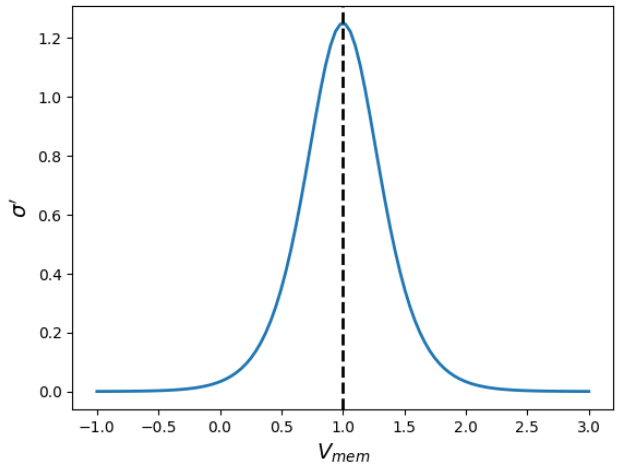

Figure 5: The derivative, σ' , of the sigmoid surrogate used in our study: $\sigma'(x, \alpha) = \alpha\,\sigma(\alpha \cdot x)\,(1 - \sigma(\alpha \cdot x))$. Where $\sigma(x)$ is the sigmoid function and α the scaling factor. Here $\alpha = 5$ and $V_{th} = 1$. The threshold voltage V_{th} is also plotted as a vertical dotted line.

Since the neuron that follows the summation point directly receives the unweighted spikes $S_l(t)$ as input, its membrane potential is either close to zero or higher than V_{th}. As it can be seen from Fig. 5, in both cases the derivative σ' has very low amplitude, thus creating the conditions for the gradient to vanish as already observed in Fang et al. [2021a]. To overcome this vanishing gradient problem we replace the surrogate gradient function of the barrier neurons with a Straight-Through-Estimator (STE) Bengio et al. [2013]. The STE introduces a far larger deviation from the *true gradients* than a surrogate function like the one shown in Fig. 5. But, the rational behind our choice is that we can accept some deviation from the true gradients in exchange for a higher gradient amplitude. Moreover, by using the STE the gradient of the barrier neurons becomes:

$$\frac{\partial O_l}{\partial S_l} = \sigma'(S_l - V_{th}) = 1 \tag{10}$$

By replacing the previous gradient into Eq. 9 we can observe that the gradients that are back-propagated through the direct and the residual connections are now equal for both SEW-ResNets and Sparse-ResNets. We provide experimental results in Sec. 6.3 and Sec. 6.4 to confirm our analysis on the gradients and the sparsity for both Sparse-ResNets and SEW-ResNets.

5 Experimental Results

5.1 Datasets and Models

We train the SNN models using SG training method on two different image classification datasets (CIFAR-10, CIFAR-100) Krizhevsky [2009] as well as a neuromorphic data classification dataset (CIFAR-10-DVS) Li et al. [2017].

The CIFAR-10/100 datasets are made of 32x32 RGB images representing 10 and 100 classes respectively. For CIFAR-10/100 we train the SNNs using stochastic gradient descent (SGD), with a learning rate of $8 \cdot 10^{-2}$. The learning rate is exponentially decayed with a factor of 0.9 each 50 epochs, and data augmentation (random resize and horizontal flip) is used. Each network is trained for 1500 epochs.

The CIFAR-10-DVS neuromorphic dataset is obtained by recording the moving images of the CIFAR-10 dataset on a LCD monitor by a DVS camera. The DVS camera has a spatial resolution of 128×128. The event-to-frame integration method described in Fang et al. [2021b] is used for pre-processing CIFAR-10-DVS. The neuromorphic data are then represented by T frames which contain the summation of the events, in both positive and negative polarities, in each temporal slice. Similar event-to-frame integration methods are widely adopted in SNN research Kaiser et al. [2020], Lee et al. [2016], Xing et al. [2020]. The dataset is composed of 1K images per class for a total of 10K images. In our setting, 80% of the images are used for training and the rest for testing the models. We train Sparse-ResNet18 and VGG16 on the CIFAR-10-DVS using the Adam optimizer and a learning rate of 10^{-3} that is exponentially decayed with a factor of 0.5 each 50 epochs. Finally, the network is trained for 500 epochs.

5.2 Network activity and sparsity

The SNN activity is defined as the total number of spikes generated by the SNN during the inference, that is during T timesteps. Let us call $z_l(t)$, of dimension (T, C, H, W), the output of the l^{th} layer of an SNN with total depth L. Then the total activity of the layer l is computed as follows:

$$\#Spikes_l = \sum^{T}\sum^{C}\sum^{H}\sum^{W} z_l(t)|_{[0,1]} \tag{11}$$

Where $z_l(t)|_{[0,1]}$ represents the value of $z(t)$ restricted to the interval $[0, 1]$. Finally, the total activity of the SNN is computed by summing the activities of each layer as follows:

$$\#Spikes_{SNN} = \sum^{l} \#Spikes_l \tag{12}$$

The activity defined above measures the total number of events, i.e. spikes, that have to be processed and transmitted for each inference. As discussed in the next section, this metric is tightly related to the SNN energy consumption.

5.3 Energy estimation

The SNN energy consumption is estimated using the metric introduced in Lemaire et al. [2023]. The amount of energy consumed by the SNN during the inference is computed by taking into account the network architecture, the number of parameters and activations as well as the sparsity. Based on these information, the energy model outputs the energy consumed by the synaptic operations, the memory accesses and addressing.

Memory accesses are computed based on the assumption that each SNN layer has its own local Static RAM (SRAM) memory. This local memory stores the layer parameters (weights and biases) and also acts as a buffer for the spikes emitted by the spiking neurons.

Synaptic operations are typically considered to be accumulations (ACC) for SNNs. This is indeed the case when $N = 1$ (binary spikes) as the synaptic weight is accumulated at most once each timestep into the membrane potential. In such a case we consider the energy cost of a synaptic operation to be equal to one ACC operation. However when $N > 1$ (multi-level spikes), the synaptic weight could be accumulated up to N times for each received spike. In such a case, we assume the worst case hypothesis described above, and consider that the synaptic energy cost of a multi-level spike corresponds to $N \times$ ACC operations, whatever the exact value carried by each spike. This overestimation is negligible as we discuss in Sec. 6.1 and 6.2.

5.4 Comparison to previous works on image classification

In this section our proposed method is compared with state-of-the-art results on the CIFAR-10/100 image datasets. The experimental results are provided in Tab. 1. To provide a comprehensive comparison we include results from different training methods. In ANN/QANN2SNN methods the resulting SNN is obtained by converting either an FP ANN or a quantized ANN respectively. On the other hand, in SG methods the SNN is directly trained in the spiking domain with surrogate gradients. As it can be observed from Tab. 1, the number of timesteps, i.e. the latency, can be drastically reduced from ANN2SNN to QANN2SNN and SG methods. These results highlight the importance of taking into account the quantization noise during the training process to achieve both high accuracy and low latency. We can also observe that, for all datasets and network architectures, the adoption of multi-level neurons allows further reducing the latency compared to binary SNNs. Our results indeed show that using only 1 timestep we can obtain the best accuracy results compared to both binary and others ternary or multi-level SNNs with equal or similar network architectures. On CIFAR-100 for instance we improve by more than 3% the accuracy compared to IM-Loss Guo et al. [2022b] while reducing the latency by a factor of 5. Similar trends can be observed for the CIFAR-10 dataset where, for example, we improve the accuracy by 0.35% and reduce the latency by 4, compared to RecDis-SNN Guo et al. [2022a] using ResNet18/19 respectively.

Moreover, it can be observed that Sparse-ResNet achieves similar or even slightly better performance results compared to SEW-ResNet, thus validating our approach based on a barrier neuron with STE and described in Sec. 4.4. More results are provided in Sec. 6.4 to show that Sparse-ResNet also improves the sparsity compared to SEW-ResNet. Moreover, in Sec. 6.3 we provide experimental results that validate the effectiveness of the STE in reducing the vanishing gradient problem in spiking residual networks.

5.5 Comparison to previous works on Neuromorphic data classification

The experimental results obtained for the CIFAR-10-DVS dataset are shown in Tab. 2. Here we report the results for a multi-level Sparse-ResNet18 and VGG16 for the $[T = 1, N = 10]$ configuration which provides 10 quantization

Table 1: Comparison with SoTA methods on CIFAR-10/100. Ours results are given for $N = 4$.

Dataset	Method	Type	Neuron	Architecture	Timestep	Accuracy
CIFAR-10	RMP-SNN Han et al. [2020]	ANN2SNN	binary	VGG16	2048	93.63%
	QFFS Li et al. [2022]	QANN2SNN	binary	VGG16	4	92.64%
	QCFS Ding et al. [2021]	QANN2SNN	binary	VGG16	4	93.86%
	DIET-SNN Rathi and Roy [2021]	SG	binary	VGG16	10	93.44%
	IM-Loss Guo et al. [2022b]	SG	binary	VGG16	5	93.85%
	ATIF Castagnetti et al. [2023a]	SG	binary	VGG16	4	93.13%
	Ours	SG	**multi-level**	VGG16	**1**	**94.34**%
	S-ResNet Hu et al. [2023a]	ANN2SNN	binary	ResNet20	350	91.82%
	ACP Li et al. [2021a]	ANN2SNN	binary	ResNet20	4	84.70%
	QFFS Li et al. [2022]	QANN2SNN	binary	ResNet18	4	93.14%
	DIET-SNN Rathi and Roy [2021]	SG	binary	ResNet20	10	92.54%
	IM-Loss Guo et al. [2022b]	SG	binary	ResNet19	4	95.40%
	RecDis-SNN Guo et al. [2022a]	SG	binary	ResNet19	4	95.53%
	Ternary Spike Guo et al. [2023]	SG	ternary	ResNet20	4	94.96%
	MLF Feng et al. [2022]	SG	multi-level	ResNet20	4	94.25%
	MBLIF Xiao et al. [2024]	SG	multi-level	ResNet20	1	94.59%
	Ours	SG	**multi-level**	SEW-ResNet18	**1**	**95.94**%
	Ours	SG	multi-level	Sparse-ResNet18	1	95.69%
CIFAR-100	RMP-SNN Han et al. [2020]	ANN2SNN	binary	VGG16	2048	70.93%
	QCFS Ding et al. [2021]	QANN2SNN	binary	VGG16	4	69.62%
	DIET-SNN Rathi and Roy [2021]	SG	binary	VGG16	5	69.67%
	IM-Loss Guo et al. [2022b]	SG	binary	VGG16	5	70.18%
	Ours	SG	**multi-level**	VGG16	**1**	**73.75**%
	DIET-SNN Rathi and Roy [2021]	SG	binary	ResNet20	5	64.07%
	RecDis-SNN Guo et al. [2022a]	SG	binary	ResNet19	4	74.10%
	Ternary Spike Guo et al. [2023]	SG	ternary	ResNet20	4	74.02%
	MBLIF Xiao et al. [2024]	SG	multi-level	ResNet20	1	75.43%
	Ours	SG	multi-level	SEW-ResNet18	1	75.27%
	Ours	SG	**multi-level**	Sparse-ResNet18	**1**	**75.7**%

Table 2: Comparison with SoTA methods on CIFAR-10-DVS. Ours results are given for $N = 10$.

Dataset	Method	Type	Neuron	Architecture	Timestep	Accuracy
CIFAR-10-DVS	DSR Meng et al. [2022]	SG	binary	VGG11	20	77.27%
	Dspike Li et al. [2021b]	SG	binary	ResNet18	10	75.4%
	MSG Li et al. [2024]	SG	binary	ResNet18	10	79.35%
	SEW-ResNet Fang et al. [2021a]	SG	binary	Wide-7B-Net	16	74.4%
	IM-Loss Guo et al. [2022b]	SG	binary	ResNet19	10	72.6%
	RecDis-SNN Guo et al. [2022a]	SG	binary	ResNet19	10	72.42%
	MLF Feng et al. [2022]	SG	multi-level	ResNet14	10	70.36%
	Ternary Spike Guo et al. [2023]	SG	ternary	ResNet20	10	79.8%
	Ours	SG	multi-level	Sparse-ResNet18	1	79.1%
	Ours	SG	multi-level	VGG16	1	76.97%

intervals. As we can see from Tab. 2 we can reach close to the state of the art accuracy while decreasing the latency to only 1 timestep. Previous state of the art methods, both with binary and multi-level neurons, are significantly outperformed as they require a latency 10 to 20 times higher to get a similar performance level.

More results on sparsity and energy consumption will be provided in Sec. 6.2, to assess wether the reduction in latency also translates in energy efficiency improvement.

6 Ablation studies

6.1 Energy efficiency comparison between multi-level and binary spiking neurons on Image classification

Some experiments were conducted to compare the energy efficiency of binary and multi-level SNNs on the CIFAR-10 dataset. To do so, different configurations of the VGG16 architecture were studied, both in terms of latency T and spikes value N. In the first configuration we fix $N = 1$ and train the VGG16 network with different number of timesteps in the range $T \in [1, 10]$. In the second configuration, we set the number of timestep to $T = 1$ and the value of the spikes in the range $N \in [1, 10]$. This setup allows comparing configurations with equal number of quantization intervals. As an example the configurations $[T = 4, N = 1]$ and $[T = 1, N = 4]$ both provide the same number of quantization intervals. However, in the first case the quantization process takes place through time by processing multiple binary spikes. In the second case the discretization process is performed in one timestep through multi-valued spikes. For each configuration, SNNs are trained using the setup described in Sec. 5.1.

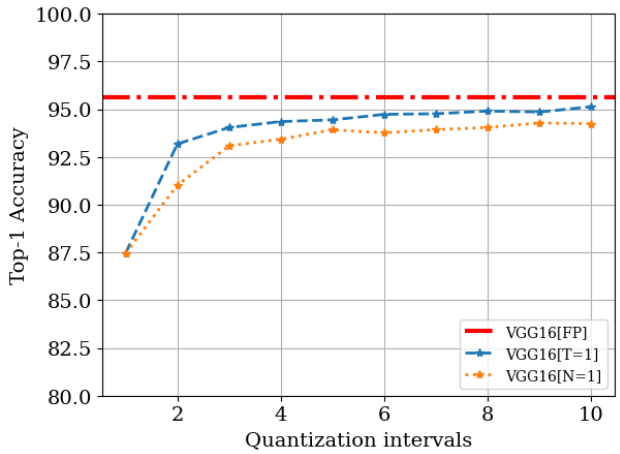

Figure 6: **Top-1 accuracy of VGG16 on the CIFAR-10 dataset**. The horizontal dotted line represents the accuracy of the ANN FP version of the network.

The accuracy of the different SNNs configurations as well as the ANN baseline is shown in Fig. 6. As expected, the accuracy gap between the ANN and the SNNs reduces when increasing the number of quantization intervals. Moreover, as it can be observed, there is a functional equivalence between both SNN configurations. For a given number of quantization intervals, the accuracy converges to very close values. These results thus confirm the analysis of Sec. 4.2, where we have shown that the same quantization function can be obtained with different $[N, T]$ configurations.

The total spiking activity as well as the total energy consumption are shown in Fig. 7 (A) and Fig. 7 (B) respectively. Even though the different binary and multi-level configurations are functionally equivalent, we can observe that they are considerably different both in terms of spiking activity and energy consumption. As an example, the $[T = 4, N = 1]$ configuration generates $130 \cdot 10^3$ binary spikes for each inference, while the same functional configuration with multi-level neurons, $[T = 1, N = 4]$, only produces $57 \cdot 10^3$ valued spikes. That is 43% less spikes that have to be processed for each input image. Moreover, it can be observed that for the binary SNNs the total activity grows almost linearly with the timesteps while there is a (sub)logarithmic increase for the multi-level configurations. The spiking activity directly impacts the total energy consumption shown Fig. 7 (B). It can be observed that even if the binary SNNs are more energy efficient than the ANN at low-timesteps, e.g. the binary SNNs provides 35% energy reduction when $[T = 4, N = 1]$, the gap quickly reduces as T increases. At $T = 8$ for instance, the binary SNNs already consumes 10% more energy than the ANN. On the other hand, the multi-level VGG16 maintains a high level of energy efficiency at each operating point. As an example, when $[T = 1, N = 4]$ there is a 66% energy reduction compared to the ANN. At the $[T = 1, N = 8]$ operating point the energy reduction is still 60% compared to the ANN.

Finally, the complete energy consumption breakdown is shown in Tab. 3. Here we provide the detailed estimation for the ANN and the configurations with 4 and 8 quantization intervals for both the binary and the multi-level SNNs. As it can be seen, the total energy consumption, especially for the SNNs, is strongly dominated by the cost of memory accesses. Each spike generated by a neuron leads to three memory accesses (two read operations for retrieving the weights and the current value of the membrane potential and a write operation) and a synaptic operation (ACC or $N \times$ ACC for the binary and multi-valued case respectively). The energy cost for one memory access is on average 10 to 100 times higher than a synaptic operation Jouppi et al. [2021]. Indeed, we can observe differences of two to

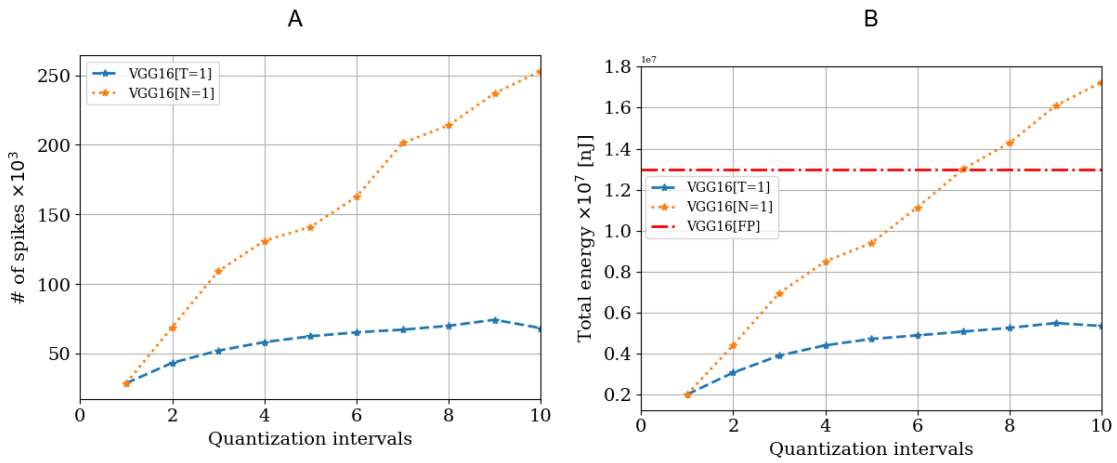

Figure 7: **Spiking activity and total energy consumption of SNNs and ANN for the CIFAR-10 dataset**. We provide (A) the total spiking activity of VGG16 for different configurations of multi-level SNNs $\{T = 1,\ N \in [1, 10]\}$ and binary SNNs $\{T \in [1, 10],\ N = 1\}$. (B) The total energy consumption of the different SNN configurations. The horizontal dotted line represents the total energy consumption of the ANN.

Table 3: Energy consumption breakdown for binary and multi-level VGG16 SNNs and ANN on CIFAR-10. All energy values are expressed in $[nJ]$. E_N corresponds to the total energy consumption of the multi-level SNN, while E_{binary} refers to the binary SNN. The symbol $[\downarrow]$ means lower is better.

	ANN	**SNN**			
		binary $[T = 4, N = 1]$	multi-level $[T = 1, N = 4]$	binary $[T = 8, N = 1]$	multi-level $[T = 1, N = 8]$
Memory					
Potentials	-	$6.0 \cdot 10^6$	$3.1 \cdot 10^6$	$10.0 \cdot 10^6$	$3.6 \cdot 10^6$
Weights	$5.58 \cdot 10^6$	$2.4 \cdot 10^6$	$1.3 \cdot 10^6$	$4.1 \cdot 10^6$	$1.54 \cdot 10^6$
Bias	$6.9 \cdot 10^2$	$27.8 \cdot 10^2$	$6.9 \cdot 10^2$	$55.7 \cdot 10^2$	$6.9 \cdot 10^2$
In/Out	$6.1 \cdot 10^6$	$3.5 \cdot 10^3$	$1.54 \cdot 10^3$	$5.7 \cdot 10^3$	$1.8 \cdot 10^3$
Total	$11.6 \cdot 10^6$	$8.4 \cdot 10^6$	$4.38 \cdot 10^6$	$14.2 \cdot 10^6$	$5.19 \cdot 10^6$
Synaptic Operations	$1.29 \cdot 10^6$	$13.3 \cdot 10^3$	$27.9 \cdot 10^3$	$22.4 \cdot 10^3$	$65.4 \cdot 10^3$
Addressing	$4.9 \cdot 10^2$	$9.0 \cdot 10^2$	$5.5 \cdot 10^2$	$14.8 \cdot 10^2$	$8.7 \cdot 10^2$
Total	$\mathbf{12.9 \cdot 10^6}$	$\mathbf{8.5 \cdot 10^6}$	$\mathbf{4.41 \cdot 10^6}$	$\mathbf{14.28 \cdot 10^6}$	$\mathbf{5.26 \cdot 10^6}$
$\mathbf{E_N/E_{binary}}$ $[\downarrow]$		0.51		0.37	
$\mathbf{E_{SNN}/E_{ANN}}$ $[\downarrow]$		0.65	0.34	1.10	0.4

three orders of magnitude between the energy consumed by the memory accesses and the synaptic operations. We can also observe that, as expected, the energy consumed by the synaptic operations is greater for the multi-valued spike. However, as the amount of generated spikes is lower than binary spikes, this increase is largely counterbalanced by the decrease of energy required for accessing the memory. These results clearly confirm that the total amount of spikes, shown in Fig. 7 (1) is the most important metric to consider for reducing the energy consumption of SNNs. Neither the average activity per timestep nor the latency alone are sufficient to properly evaluate the overall energy consumption. In that sense our results show that, using multi-valued spikes, the energy efficiency of SNNs can be greatly improved by 2 to 3 times depending on the number of quantization intervals.

6.2 Energy efficiency comparison between multi-level and binary spiking neurons on Neuromorphic data classification

In this section we study two configurations of the VGG16 architecture, with multi-level and binary neurons, on the CIFAR-10-DVS neuromorphic classification dataset. Specifically we compare the multi-level VGG16 network given in Tab. 2 with a binary VGG16 trained on $T = 10$ timesteps. Both configurations thus provide 10 quantization

Table 4: Energy consumption breakdown for binary and multi-level VGG16 SNNs and ANN on CIFAR-10-DVS. All energy values are expressed in $[nJ]$. E_N corresponds to the total energy consumption of the multi-level SNN, while E_{binary} refers to the binary SNN. The symbol $[\downarrow]$ means lower is better.

	ANN	**SNN**	
		binary $[T = 10, N = 1]$	multi-level $[T = 1, N = 10]$
Memory			
Potentials	-	$233.3 \cdot 10^6$	$90.6 \cdot 10^6$
Weights	$52 \cdot 10^6$	$32.2 \cdot 10^6$	$12.8 \cdot 10^6$
Bias	$6.9 \cdot 10^2$	$69.7 \cdot 10^2$	$6.9 \cdot 10^2$
In/Out	$186.3 \cdot 10^6$	$40.5 \cdot 10^3$	$15.8 \cdot 10^3$
Total	$238.3 \cdot 10^6$	$265.5 \cdot 10^6$	$103.4 \cdot 10^6$
Synaptic Operations	$12.1 \cdot 10^6$	$170.8 \cdot 10^3$	$664.6 \cdot 10^3$
Addressing	$6.2 \cdot 10^3$	$10.3 \cdot 10^3$	$7.7 \cdot 10^3$
Total	$\mathbf{250.4 \cdot 10^6}$	$\mathbf{265.7 \cdot 10^6}$	$\mathbf{104.1 \cdot 10^6}$
$\mathbf{E_N/E_{binary}}$ $[\downarrow]$		0.39	
$\mathbf{E_{SNN}/E_{ANN}}$ $[\downarrow]$		1.06	0.41

intervals as well as similar accuracy levels: 76.97% when $\{T = 1,\ N = 10\}$ and 75.5% for the binary configuration $\{T = 10,\ N = 1\}$. The energy breakdown for both SNNs as well as the ANN is given in Tab. 4.

First it can be observed that the total energy consumption for both the SNNs and the ANN is almost twenty time higher than the CIFAR-10 results shown in Tab. 3. The higher resolution of CIFAR-10-DVS (128×128) compared to CIFAR-10 (32×32) leads to an increase of the number of memory accesses and operations to process the input frames as well as the feature maps of hidden layers. However, results shown in Tab. 4 confirm the energy efficiency improvements of multi-level SNNs, even on neuromorphic data. As it can be observed, the multi-level VGG16 reduces by 60% the energy consumption compared to the ANN. The energy gains come from the reduction of energy related to memory transfers. The multi-level SNN consumes in total $103.4 \cdot 10^6$ nJ for the memory accesses, while $238.3 \cdot 10^6$ nJ are required for the ANN. In the multi-level SNN, most of the energy is consumed for accessing the neuron potentials ($90.6 \cdot 10^6$ nJ) while only $12.8 \cdot 10^6$ nJ are related to the memory accesses associated with the synaptic weights. On the other hand, the ANN consumes four times more energy for accessing the synaptic weights ($52 \cdot 10^6$ nJ) and twice more for accessing the feature maps, i.e. In/Out $186.3 \cdot 10^6$ nJ. Finally, there is a difference of almost three order of magnitude in the energy consumption of the synaptic operations between the ANN and the SNNs. However, for the ANN the energy related to the synaptic operations accounts for less than 5% of the total energy budget. Our results are close to those already observed in Dampfhoffer et al. [2023], that is the energy cost of synaptic operations is very small compared to the one of transferring data from the memory. Therefore, the energy gains obtained by reducing the cost of the synaptic operations are marginal when considering the total energy consumption. Finally, the binary SNN does not exhibit energy gains compared to the ANN when using latencies that provide state of the art accuracy results on the CIFAR-10-DVS dataset, i.e. 10 or more timesteps. When $T = 10$ the SNN fires $1.5 \cdot 10^6$ spikes while only $591 \cdot 10^3$ spikes are emitted when $N = 1$.

As shown in Tab. 4, it represents a 60% reduction in the total spiking activity which translates in an equivalent energy gain between multi-level and binary SNNs. These results again confirm that, even for neuromorphic datasets, the energy consumption can be lowered by reducing the total amount of spikes generated by the SNN.

6.3 Gradient propagation in spiking ResNet architectures

In this section we study the gradient propagation in the spiking residual architectures SEW-ResNet and the Sparse-ResNet that has been introduced in Sec. 4.4. To assess the improvement provided by the STE we also consider a variant of Sparse-ResNet where all the spiking neurons, including the barrier neurons, use the same surrogate derivative functions shown in Fig. 5. For each block of the three architectures, SEW-REsNet18, Sparse-ResNet18 with STE and Sparse-ResNet18 without STE we measure the gradient norm on the direct path, $\frac{\partial L}{\partial A_l}$, the residual path $\frac{\partial L}{\partial R_l}$ and after the summation point $\frac{\partial L}{\partial O_l}$. The gradient is averaged over 10^4 minibatches of the CIFAR-10 dataset. We also repeat the simulation by training the networks 10 times starting from different initial conditions. The gradient mean and standard deviation shown in Fig. 8 are provided for each residual block of the network . The experimental results

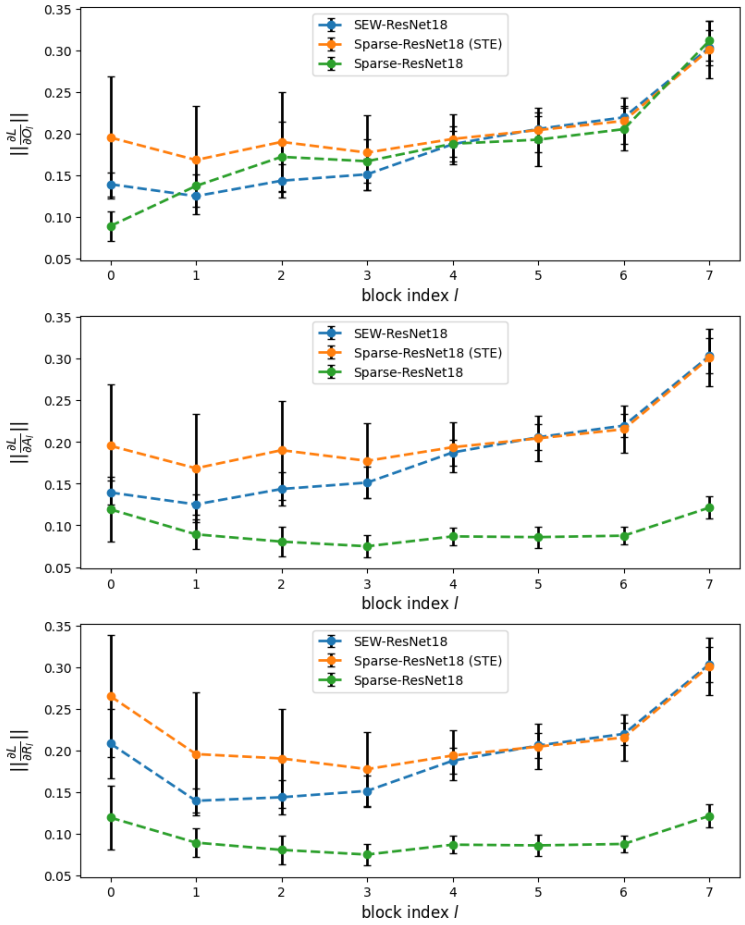

Figure 8: **Using STE helps overcome the vanishing gradient**. The gradients of the residual blocks in SEW-ResNet18, Sparse-ResNet18 with STE and Sparse-ResNet18 without STE on the CIFAR-10 dataset. The block 0 is the first block after the input.

confirm the analysis given in Sec. 4.4. As it can be observed, the gradients norm on the direct and residual paths of SEW-ResNet and Sparse-ResNet with STE are closer to each other and consistently higher than the gradients of Sparse-ResNet without STE. Indeed, when STE is not used the updates of the gradient on Sparse-ResNet are weaker, thus slowing down the training and decreasing the performance. This behavior can be observed in Fig. 9 where we show the validation loss, computed at the end of each training epoch for the three networks on the CIFAR-10 dataset. The loss of Sparse-ResNet without STE declines more slowly and finally reaches a higher value compared to SEW-ResNet and Sparse-ResNet with STE.

6.4 Spikes propagation in residual layers

In this section we provide some experimental results to assess the effectiveness of the barrier neurons in limiting the spike propagation in the residual layers. To do so, we compare the activity of SEW-ResNet and Sparse-ResNet with STE measured on the CIFAR-10 dataset. Both networks are trained using the $[T = 1, N = 4]$ configuration, that provides almost the same accuracy, as shown in Tab. 1. We measure the number of generated spikes in the direct and the residual paths for both networks. The spikes that are generated at the summation point are computed as the sum of the number of spikes coming from the direct path and the spikes coming from the residual path for SEW-ResNet. For Sparse-ResNet, we measure instead the number of spikes after the barrier neuron. Our measure corresponds, in both cases, to the amount of spikes that are transferred from one layer to the next in the context of an event-based communication/processing scheme. The experimental results are shown in Fig. 10, where we can observe that SEW-ResNet consistently generates more spikes in the first layers of the network compared to Sparse-ResNet. The the number of generated spikes stabilizes from the middle to the last layers, where both networks generate almost the same activity. The spike avalanche effect,

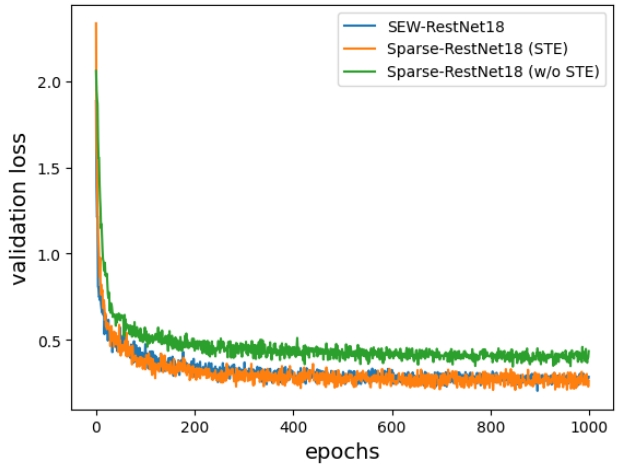

Figure 9: Comparison of the validation loss of SEW-REsNet18, Sparse-ResNet18 with STE and Sparse-ResNet18 without STE on CIFAR-10.

that we qualitatively described in Sec. 4.3, can be clearly observed in Fig. 10. The number of spikes generated by SEW-ResNet at the first (sum_0) and especially at the second (sum_1) summation points are considerably higher compared to Sparse-ResNet.

SEW-ResNet produces 45665 spikes at the summation point sum_0 while only 37150 spikes are fired by Sparse-ResNet. The avalanche effect is even more visible at sum_1 where there are 67848 spikes for SEW-ResNet and only 35819 for Sparse-ResNet. That is 47% less activity in one of the layer that generates most of the spikes in the whole network. These results confirm that the barrier neurons of Sparse-ResNet are able to prevent the spike avalanche effect, thus reducing the network activity as we discuss in more details in the next section.

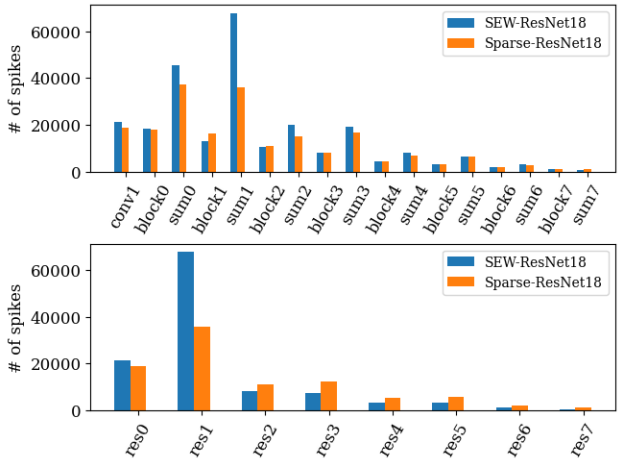

Figure 10: **Layer-wise activity of SEW-ResNet18 and Sparse-ResNet18**. Top: number of spikes generated in the direct path ($block_i$) and after the summation point (sum_i). Bottom: number of spikes generated in the residual path (res_i). The values are averaged over all the images of the CIFAR-10 test set.

6.5 Analysis of sparsity in spiking ResNet architectures

In this final section we provide experimental results on the spiking activity for multi-level SEW-ResNet and Sparse-ResNet. The accuracy as well as the total spiking activity for different configurations on CIFAR-10, i.e. $\{T = 1,\ N \in [1, 8]\}$, are shown in Fig. 11.

As it can be observed, both networks provide very similar accuracy results, with a marginally better accuracy for SEW-ResNet when $N > 2$. However, we can observe that SEW-ResNet generates a much higher level of spiking

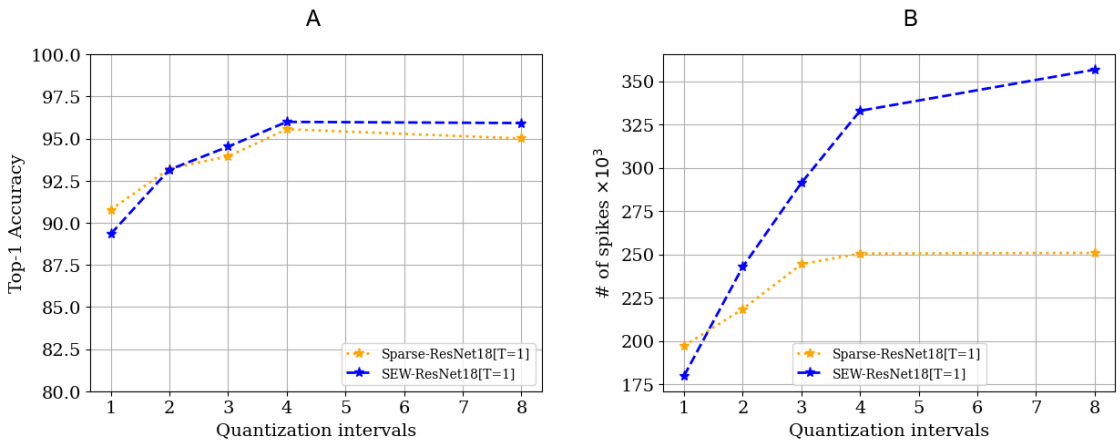

Figure 11: **(A) Top-1 accuracy and (B) spiking activity of Sparse-ResNet18 and SEW-ResNet18 on the CIFAR-10 dataset.** For all the configurations the number of timesteps $T = 1$. The spiking activity is computed by averaging over all the images of the CIFAR-10 test set.

activity for all the configurations except when $N = 1$. As an example, when $N = 2$ SEW-ResNet fires 242662 spikes/image while Sparse-ResNet generates 218343 spikes/image, which corresponds to a 10% reduction in the overall activity while providing very similar accuracy. The spiking activity gap between both network architectures widens as N increases. For the configuration $N = 4$, Sparse-ResNet generates 25% less spikes/image compared to SEW-ResNet with only 0.25% accuracy drop. The spiking activity reduction provided by Sparse-ResNet rises to 30% when $N = 8$ with an accuracy gap less than 1%. As discussed in Sec. 4.3 and shown in Fig. 10, the difference in spiking activity is mainly due to the spike avalanche effect caused by the first two shortcut connections in the ResNet18 network. This effect would be exacerbated in deeper networks, that is when more consecutive layers use skip connections such as in the ResNet34 architecture He et al. [2016], when up to five consecutive residual blocks use shortcut connections. Another example is the MobileNetV2 architectures Sandler et al. [2019], which is composed of 17 consecutive layers of inverted residual blocks. A shortcut connects the input with the output of each block. For the spiking version of that particular architecture, the spike avalanche effect would be significantly higher if the spikes are aggregated at the end of each residual connections as shown in Fig. 4. In that case the use of barrier neurons, as proposed in this paper, could lead to a significant reduction in the total spiking activity and thus an energy efficiency improvement.

7 Conclusion and Future works

In this paper, we proposed to improve the sparsity and energy efficiency of SNNs both at neuronal and at network level. By leveraging multi-level spiking neurons we have shown that the energy efficiency of SNNs cab be improved. By reducing the quantization noise we are able to provide state of the art results using only 1 timestep, therefore at the lowest possible latency. Our approach can be also applied to neuromorphic data, for which state of the art accuracy were obtained while reducing the latency by a factor of 10. At the architectural level, we identified the spike avalanche effect that impacts most of spiking residual architectures. Sparse-ResNet has been proposed as a solution to the avalanche effect. This residual architecture is indeed able to provide state of the art accuracy results on image classification while reducing by more than 20% the network activity. We identified several deep residual architectures, widely used in image processing, that can be impacted by the spike avalanche effect. We believe that it would be interesting to extend our approach to these architectures and evaluate the energy efficiency gains, both with realistic energy models and using real neuromorphic hardware.

References

Chen Li, Lei Ma, and Steve Furber. Quantization Framework for Fast Spiking Neural Networks. *Frontiers in Neuroscience*, 16, 2022. ISSN 1662-453X. URL `https://www.frontiersin.org/articles/10.3389/fnins.2022.918793`.

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*, pages 1–9,

2021. ISSN 2162-2388. doi:10.1109/TNNLS.2021.3111897. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

Andrea Castagnetti, Alain Pegatoquet, and Benoît Miramond. Trainable quantization for Speedy Spiking Neural Networks. *Frontiers in Neuroscience*, 17, 2023a. ISSN 1662-453X. URL https://www.frontiersin.org/articles/10.3389/fnins.2023.1154241.

Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. RecDis-SNN: Rectifying Membrane Potential Distribution for Directly Training Spiking Neural Networks. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 326–335, New Orleans, LA, USA, June 2022a. IEEE. ISBN 978-1-66546-946-3. doi:10.1109/CVPR52688.2022.00042. URL https://ieeexplore.ieee.org/document/9880053/.

Yongjun Xiao, Xianlong Tian, Yongqi Ding, Pei He, Mengmeng Jing, and Lin Zuo. Multi-Bit Mechanism: A Novel Information Transmission Paradigm for Spiking Neural Networks, July 2024. URL http://arxiv.org/abs/2407.05739. arXiv:2407.05739 [cs] version: 1.

Edgar Lemaire, Loïc Cordone, Andrea Castagnetti, Pierre-Emmanuel Novac, Jonathan Courtois, and Benoît Miramond. An Analytical Estimation of Spiking Neural Networks Energy Efficiency. In Mohammad Tanveer, Sonali Agarwal, Seiichi Ozawa, Asif Ekbal, and Adam Jatowt, editors, *Neural Information Processing*, pages 574–587, Cham, 2023. Springer International Publishing. ISBN 978-3-031-30105-6. doi:10.1007/978-3-031-30105-6_48.

Manon Dampfhoffer, Thomas Mesquida, Alexandre Valentian, and Lorena Anghel. Are SNNs Really More Energy-Efficient Than ANNs? an In-Depth Hardware-Aware Study. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(3):731–741, June 2023. ISSN 2471-285X. doi:10.1109/TETCI.2022.3214509. URL https://ieeexplore.ieee.org/document/9927729/?arnumber=9927729. Conference Name: IEEE Transactions on Emerging Topics in Computational Intelligence.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep Residual Learning in Spiking Neural Networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 21056–21069. Curran Associates, Inc., 2021a. URL https://proceedings.neurips.cc/paper/2021/hash/afe434653a898da20044041262b3ac74-Abstract.html.

Peter U. Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2015. doi:10.1109/IJCNN.2015.7280696. ISSN: 2161-4407.

Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification. *Frontiers in Neuroscience*, 11:682, 2017. ISSN 1662-453X. doi:10.3389/fnins.2017.00682. URL https://www.frontiersin.org/article/10.3389/fnins.2017.00682.

Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. RMP-SNN: Residual Membrane Potential Neuron for Enabling Deeper High-Accuracy and Low-Latency Spiking Neural Network. pages 13558–13567, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Han_RMP-SNN_Residual_Membrane_Potential_Neuron_for_Enabling_Deeper_High-Accuracy_and_CVPR_2020_paper.html.

Ana Stanojevic, Stanisław Woźniak, Guillaume Bellec, Giovanni Cherubini, Angeliki Pantazi, and Wulfram Gerstner. An exact mapping from ReLU networks to spiking neural networks. *Neural Networks*, 168:74–88, November 2023. ISSN 0893-6080. doi:10.1016/j.neunet.2023.09.011. URL https://www.sciencedirect.com/science/article/pii/S0893608023005051.

Yuchen Wang, Hanwen Liu, Malu Zhang, Xiaoling Luo, and Hong Qu. A universal ANN-to-SNN framework for achieving high accuracy and low latency deep Spiking Neural Networks. *Neural Networks*, 174:106244, June 2024. ISSN 0893-6080. doi:10.1016/j.neunet.2024.106244. URL https://www.sciencedirect.com/science/article/pii/S0893608024001680.

Jianhao Ding, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Optimal ANN-SNN Conversion for Fast and Accurate Inference in Deep Spiking Neural Networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 2328–2336, Montreal, Canada, August 2021. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-9-6. doi:10.24963/ijcai.2021/321. URL https://www.ijcai.org/proceedings/2021/321.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation, August 2013. URL http://arxiv.org/abs/1308.3432. arXiv:1308.3432 [cs].

Hongchao Yang, Suorong Yang, Lingming Zhang, Hui Dou, Furao Shen, and Jian Zhao. CS-QCFS: Bridging the performance gap in ultra-low latency spiking neural networks. *Neural Networks*, 184:107076, April 2025. ISSN 0893-6080. doi:10.1016/j.neunet.2024.107076. URL https://www.sciencedirect.com/science/article/pii/S0893608024010050.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks. *IEEE Signal Processing Magazine*, 36(6):51–63, November 2019. ISSN 1558-0792. doi:10.1109/MSP.2019.2931595. Conference Name: IEEE Signal Processing Magazine.

Andrea Castagnetti, Alain Pegatoquet, and Benoît Miramond. SPIDEN: deep Spiking Neural Networks for efficient image denoising. *Frontiers in Neuroscience*, 17, 2023b. ISSN 1662-453X. URL https://www.frontiersin.org/articles/10.3389/fnins.2023.1224457.

Yang Li, Feifei Zhao, Dongcheng Zhao, and Yi Zeng. Directly training temporal Spiking Neural Network with sparse surrogate gradient. *Neural Networks*, 179:106499, November 2024. ISSN 08936080. doi:10.1016/j.neunet.2024.106499. URL https://linkinghub.elsevier.com/retrieve/pii/S0893608024004234.

Yufei Guo, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Yinglei Wang, Xuhui Huang, and Zhe Ma. IM-Loss: Information Maximization Loss for Spiking Neural Networks. *Advances in Neural Information Processing Systems*, 35:156–166, December 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/010c5ba0cafc743fece8be02e7adb8dd-Abstract-Conference.html.

Yunhua Chen, Ren Feng, Zhimin Xiong, Jinsheng Xiao, and Jian K. Liu. High-performance deep spiking neural networks via at-most-two-spike exponential coding. *Neural Networks*, 176:106346, August 2024. ISSN 0893-6080. doi:10.1016/j.neunet.2024.106346. URL https://www.sciencedirect.com/science/article/pii/S0893608024002703.

Yufei Guo, Yuanpei Chen, Xiaode Liu, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Ternary Spike: Learning Ternary Spikes for Spiking Neural Networks, December 2023. URL http://arxiv.org/abs/2312.06372. arXiv:2312.06372 [cs].

Congyi Sun, Qinyu Chen, Yuxiang Fu, and Li Li. Deep Spiking Neural Network with Ternary Spikes. In *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 251–254, October 2022. doi:10.1109/BioCAS54905.2022.9948581. URL https://ieeexplore.ieee.org/document/9948581. ISSN: 2163-4025.

Lang Feng, Qianhui Liu, Huajin Tang, De Ma, and Gang 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*, pages 2471–2477, Vienna, Austria, July 2022. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3. doi:10.24963/ijcai.2022/343. URL https://www.ijcai.org/proceedings/2022/343.

Xiaoting Wang and Yanxiang Zhang. MT-SNN: Enhance Spiking Neural Network with Multiple Thresholds, October 2024. URL http://arxiv.org/abs/2303.11127. arXiv:2303.11127 [cs].

Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-YOLO: Spiking Neural Network for Energy-Efficient Object Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11270–11277, April 2020. ISSN 2374-3468, 2159-5399. doi:10.1609/aaai.v34i07.6787. URL https://ojs.aaai.org/index.php/AAAI/article/view/6787.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi:10.1109/CVPR.2016.90. URL http://ieeexplore.ieee.org/document/7780459/.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks, March 2019. URL http://arxiv.org/abs/1801.04381. arXiv:1801.04381 [cs].

Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, September 2020. URL http://arxiv.org/abs/1905.11946. arXiv:1905.11946 [cs, stat].

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021. URL http://arxiv.org/abs/2010.11929. arXiv:2010.11929 [cs].

Yangfan Hu, Huajin Tang, and Gang Pan. Spiking Deep Residual Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):5200–5205, August 2023a. ISSN 2162-2388. doi:10.1109/TNNLS.2021.3119238. URL https://ieeexplore.ieee.org/document/9597475.

Yifan Hu, Lei Deng, Yujie Wu, Man Yao, and Guoqi Li. Advancing Spiking Neural Networks towards Deep Residual Learning, March 2023b. URL http://arxiv.org/abs/2112.08954. arXiv:2112.08954 [cs].

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. CIFAR10-DVS: An Event-Stream Dataset for Object Classification. *Frontiers in Neuroscience*, 11, May 2017. ISSN 1662-453X. doi:10.3389/fnins.2017.00309. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2017.00309/full. Publisher: Frontiers.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothee Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating Learnable Membrane Time Constant to Enhance Learning of Spiking Neural Networks, August 2021b. URL http://arxiv.org/abs/2007.05785. arXiv:2007.05785 [cs].

Jacques Kaiser, Hesham Mostafa, and Emre Neftci. Synaptic Plasticity Dynamics for Deep Continuous Local Learning (DECOLLE). *Frontiers in Neuroscience*, 14, May 2020. ISSN 1662-453X. doi:10.3389/fnins.2020.00424. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2020.00424/full. Publisher: Frontiers.

Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training Deep Spiking Neural Networks Using Backpropagation. *Frontiers in Neuroscience*, 10, November 2016. ISSN 1662-453X. doi:10.3389/fnins.2016.00508. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2016.00508/full. Publisher: Frontiers.

Yannan Xing, Gaetano Di Caterina, and John Soraghan. A New Spiking Convolutional Recurrent Neural Network (SCRNN) With Applications to Event-Based Hand Gesture Recognition. *Frontiers in Neuroscience*, 14, November 2020. ISSN 1662-453X. doi:10.3389/fnins.2020.590164. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2020.590164/full. Publisher: Frontiers.

Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A Free Lunch From ANN: Towards Efficient, Accurate Spiking Neural Networks Calibration. In *Proceedings of the 38th International Conference on Machine Learning*, pages 6316–6325. PMLR, July 2021a. URL https://proceedings.mlr.press/v139/li21d.html. ISSN: 2640-3498.

Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Training High-Performance Low-Latency Spiking Neural Networks by Differentiation on Spike Representation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12434–12443, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-6654-6946-3. doi:10.1109/CVPR52688.2022.01212. URL https://ieeexplore.ieee.org/document/9878999/.

Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable Spike: Rethinking Gradient-Descent for Training Spiking Neural Networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 23426–23439. Curran Associates, Inc., 2021b. URL https://proceedings.neurips.cc/paper/2021/hash/c4ca4238a0b923820dcc509a6f75849b-Abstract.html.

Norman P. Jouppi, Doe Hyun Yoon, Matthew Ashcraft, Mark Gottscho, Thomas B. Jablin, George Kurian, James Laudon, Sheng Li, Peter Ma, Xiaoyu Ma, Thomas Norrie, Nishant Patil, Sushma Prasad, Cliff Young, Zongwei Zhou, and David Patterson. Ten Lessons From Three Generations Shaped Google's TPUv4i : Industrial Product. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pages 1–14, June 2021. doi:10.1109/ISCA52012.2021.00010. ISSN: 2575-713X.