ONE-HOT MULTI-LEVEL LIF SPIKING NEURAL NETWORKS FOR ENHANCED ACCURACY-LATENCY TRADEOFF

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs) hold significant promise as energy-efficient alternatives to conventional artificial neural networks (ANNs). However, SNNs require computations across multiple timesteps, resulting in increased latency, heightened energy consumption, and additional memory access overhead. Techniques to reduce SNN latency down to a unit timestep have emerged to realize true superior energy efficiency over ANNs. Nonetheless, this latency reduction often comes at the expense of noticeable accuracy degradation. Therefore, achieving an optimal balance in the tradeoff between accuracy and energy consumption by adjusting the latency of multiple timesteps remains a significant challenge. In this paper, we introduce a new dimension to the accuracy-energy tradeoff space using a novel *one-hot multi-level leaky integrate-and-fire* (M-LIF) neuron model. The proposed M-LIF model represents the inputs and outputs of hidden layers as a set of one-hot binary-weighted spike lanes to find better tradeoff points while still being able to model conventional SNNs. For image classification on static datasets, we demonstrate M-LIF SNNs outperform iso-architecture conventional LIF SNNs in terms of accuracy (2% higher than VGG16 SNN on ImageNet) while still being energy-efficient ($20\times$ lower energy than VGG16 ANN on ImageNet). For dynamic vision datasets, we demonstrate the ability of M-LIF SNNs to reduce latency by $3\times$ compared to conventional LIF SNNs while limiting accuracy degradation ($< 1\%$).

1 INTRODUCTION

Neural networks have become a fundamental technique for solving many important problems such as image classification, object detection, and face recognition (Krizhevsky et al., 2012; Redmon et al., 2016). As neural network accuracy improves, models become increasingly complex, making their energy-efficient deployment on the edge a significant challenge. In order to reduce the computational complexity of these tasks, spiking neural networks (SNNs) (Maass, 1997) were proposed as an alternative to traditional artificial neural networks (ANNs) (Krizhevsky et al., 2012; Simonyan & Zisserman, 2014). SNNs infer inputs across multiple timesteps while ANNs perform a one-shot inference, essentially over a single timestep. Neurons in SNNs differ from those in ANNs as they operate on sparse binary spike trains as opposed to non-binary 'analog' activations, resulting in the substitution of multiplications with energy-efficient additions (Han et al., 2016).

To model spikes over time, SNNs employ various techniques, most notably the leaky integrate-and-fire (LIF) neuron model (Hunsberger & Eliasmith, 2015; Burkitt, 2006). Each neuron is characterized by two parameters: firing threshold and membrane leakage. During a timestep, a neuron either remains silent or produces a spike if the membrane potential exceeds its firing threshold. The membrane potential can shrink over time depending on the membrane leakage and is reset if a spike is produced. Using such models, many training methods have emerged and can be categorized into two main approaches: ANN-SNN conversion and direct training. ANN-SNN conversion methods (Rueckauer et al., 2017; Diehl et al., 2015; Bu et al., 2023) convert the weights of a pre-trained ANN to an iso-architecture SNN. However, these methods can require a large number of timesteps (on the order of 1000 in some cases (Sengupta et al., 2019)) to achieve comparable or better accuracy than ANNs. Note that multi-timestep inference results in both a higher number of operations and more memory storage/accesses which can dominate the compute cost (Horowitz, 2014). Therefore, the

need for multi-timestep processing from ANN-SNN conversion has been the primary factor making widespread deployment of SNNs impractical for energy-constrained edge scenarios.

Direct training using surrogate gradient-based optimization and back-propagation through time (BPTT) (Rathi & Roy, 2023; Deng et al., 2022; Yao et al., 2023) has enabled training SNNs with significantly fewer timesteps, occasionally reducing them to just a single timestep (Chowdhury et al., 2022). However, these SNNs still lag ANNs in terms of accuracy. For image classification on static datasets (Rawat & Wang, 2017; Krizhevsky et al., 2012), SNNs cannot bridge this accuracy gap with ANNs without increasing the number of timesteps, which in turn reduces energy efficiency. Moreover, single-timestep SNNs require iterative temporal pruning (Chowdhury et al., 2022) to converge, rendering training more time-consuming. For image classification, multi-timestep SNNs traditionally outperform ANNs on dynamic vision sensor (Leñero-Bardallo et al., 2011) data but suffer from a sharp accuracy degradation (Li et al., 2024) with a reduced number of timesteps.

As conventional LIF models employ binary-valued neuron spike outputs, traditional SNNs are restricted to solely scaling the temporal dimension T (i.e., the number of timesteps) to achieve different accuracy-energy efficiency tradeoffs. To address this limitation, we extend the range of neuron spike outputs to include more values by introducing a new dimension S to the accuracy-energy tradeoff space using a novel *one-hot multi-level leaky integrate-and-fire* (M-LIF) neuron model as illustrated in Figure 1. The proposed one-hot M-LIF model has the following key properties: 1) it uses the S dimension to represent hidden layer outputs (inputs) as a set of S_o (S_i) binary-weighted spike lanes, and 2) it limits the simultaneous firing behavior of those S_o (S_i) spike lanes to only a single lane per timestep. These two properties of our one-hot M-LIF model enable new accuracy-energy tradeoff points for SNNs while still only requiring additions (without multiplications) like the conventional LIF model. Furthermore, the proposed one-hot M-LIF model can be easily integrated into prior existing training frameworks. For static datasets such as CIFAR and ImageNet, we demonstrate that one-hot M-LIF SNNs outperform conventional LIF SNNs in terms of accuracy while achieving better or comparable energy efficiency on various architectures including the high-performance spike-driven transformer (Yao et al., 2023). For dynamic vision datasets such as DVS-CIFAR10 (Li et al., 2017; Orchard et al., 2015), we show that M-LIF SNNs using multi-level input layer encoding can achieve reduced timesteps (energy consumption) compared to conventional LIF SNNs for comparable or better accuracy. To summarize, the main contributions of this paper are:

- We propose a new direction to balance SNN energy efficiency and accuracy by expanding the neuron inputs and outputs using a novel one-hot multi-level leaky-integrate-and-fire (M-LIF) neuron model.
- We enable a new tradeoff and show that one-hot M-LIF SNNs are more accurate than iso-architecture LIF SNNs while consuming comparable or lower energy with fewer timesteps.
- We demonstrate the benefit of one-hot M-LIF SNNs with dynamic vision sensor-based input compared to conventional SNNs. To the best of our knowledge, this is the first SNN work to achieve a top-1 accuracy of 82.5% on DVS-CIFAR10 using only 3 timesteps.

2 Background & Related Works

2.1 Leaky Integrate-and-Fire Model

A conventional spiking neural network (SNN) layer under the leaky integrate-and-fire (LIF) neuron model is described by

$$\boldsymbol{u}^l[t] = \beta^l \boldsymbol{u}^l[t-1] + \boldsymbol{W}^l \boldsymbol{o}^{l-1}[t] - \theta^l \boldsymbol{o}^l[t-1] \tag{1}$$

$$\boldsymbol{o}^l[t] = \begin{cases} 1, & \text{if } 1 < \frac{\boldsymbol{u}^l[t]}{\theta^l} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\boldsymbol{W}^l$ is the weight matrix connecting layers $l-1$ and l, $\boldsymbol{u}$ is a vector containing the membrane potential of output neurons, $\beta \in [0, 1]$ is the leakage factor, $\boldsymbol{o}$ is an output vector of binary spikes, $\theta > 0$ is the firing threshold, and $t \in \{0, 1, 2, ...\}$ represents the discrete timestep. The first term in Equation 1 corresponds to the membrane leakage allowing the potential to shrink (leak) over time, and the final term accounts for resetting the potential to a specific value when an output binary spike

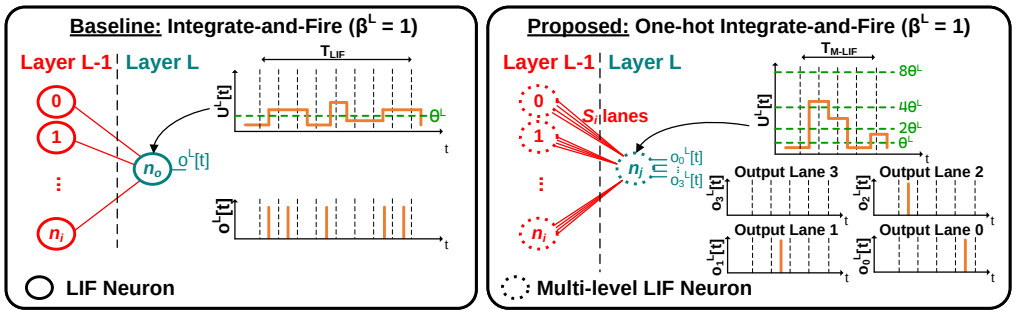

Figure 1: LIF neuron model (left) vs. one-hot M-LIF neuron model with $S_o = 4$ lanes (right).

given by Equation 2 is generated. All neurons in an input/hidden layer typically share the same leakage factor and firing threshold values. As for the final layer, adopting the LIF model without any modifications can significantly impact the accuracy (Deng et al., 2022; Rathi & Roy, 2023). Hence, the final output layer neurons only accumulate incoming inputs without any leakage and do not fire output spikes. Finally, the inference process is repeated for T timesteps from $t = 0$ to $T - 1$, and the output of the last layer is averaged to produce the final result.

2.2 ANN-SNN Conversion

ANN-SNN conversion methods (Sengupta et al., 2019; Rueckauer et al., 2017; Diehl et al., 2015; Bu et al., 2023; Deng & Gu, 2021; Li et al., 2021a) convert the weights of a pre-trained ANN to an iso-architecture SNN. Specifically, these methods convert the output of a rectified linear unit neuron in an ANN into a sequence of binary spikes in the SNN over multiple timesteps. The primary challenge in this technique is determining the firing threshold in such a way that it balances the accuracy-latency tradeoff. In these methods, the firing thresholds are generally determined by profiling the pre-trained ANN and recording a certain percentile of layers' input activation distributions. However, these heuristic techniques can lead to a sub-optimal choice of firing threshold and can also require a large number of timesteps (up to 1000 timesteps) to achieve comparable or better accuracy than ANNs, thus further aggravating the accuracy-latency tradeoff. This multi-timestep processing requirement is a challenge for widespread SNN deployment as it primarily introduces more memory storage and accesses which can be significantly higher than compute cost (Horowitz, 2014).

2.3 Direct Training

An alternate approach to training SNNs is to use gradient-based techniques, such as back-propagation, either from scratch or from a pre-trained iso-architecture ANN (Rathi & Roy, 2023; Deng et al., 2022; Chowdhury et al., 2022; Neftci et al., 2019). These approaches relate the temporal dimension of SNNs to that of recurrent neural networks, and perform back-propagation through time (BPTT) to learn weights across multiple timesteps. The cross-entropy loss L and gradients $\partial L/\partial \boldsymbol{W}^l$ are calculated by

$$L = -\sum_i y_i \log(\Phi(\boldsymbol{o}^L[T-1])_i), \quad \frac{\partial L}{\partial \boldsymbol{W}^l} = \sum_t \frac{\partial L}{\partial \boldsymbol{o}^l[t]} \frac{\partial \boldsymbol{o}^l[t]}{\partial \boldsymbol{u}^l[t]} \frac{\partial \boldsymbol{u}^l[t]}{\partial \boldsymbol{W}^l} \tag{3}$$

where L is the index of the final layer, $\Phi(\cdot)$ denotes the softmax function, and $\boldsymbol{y}$ is the one-hot encoded vector of the true label. The term $\partial \boldsymbol{o}^l[t]/\partial \boldsymbol{u}^l[t]$ in Equation 3 is the discontinuous gradient that is typically replaced by differentiable surrogate gradients. Prior works have explored the use of various surrogate gradient shapes such as triangular (Bellec et al., 2018; Rathi & Roy, 2023), or the derivative of the sigmoid function (Yao et al., 2023) which are given below in Equations 4 and 5, respectively, where γ and α are constants used to scale the shapes of the gradients.

$$\frac{\partial \boldsymbol{o}^l[t]}{\partial \boldsymbol{u}^l[t]} = \text{diag}\left(\frac{\gamma}{\theta^l} \max\{0, 1 - |\frac{\boldsymbol{u}^l[t]}{\theta^l} - 1|\}\right) \tag{4}$$

$$\frac{\partial \boldsymbol{o}^l[t]}{\partial \boldsymbol{u}^l[t]} = \text{diag}\left(\frac{\alpha}{\theta^l}\left(1 - \sigma\left(\alpha(\frac{\boldsymbol{u}^l[t]}{\theta^l} - 1)\right)\right)\sigma\left(\alpha(\frac{\boldsymbol{u}^l[t]}{\theta^l} - 1)\right)\right) \tag{5}$$

Compared to ANN-SNN conversion techniques, direct training approaches typically achieve a better accuracy-latency tradeoff (higher accuracy using fewer timesteps overall) at the cost of more compute- and memory-intensive training (Deng et al., 2020). To achieve competitive accuracy, most employ direct input encoding (Tan et al., 2021) and utilize the first layer as a spike generator by directly feeding pixel values as inputs to the network. Through gradient-based learning, many recent works have sought to optimize different aspects of this training methodology such as loss function definition (Deng et al., 2022), initialization and parameterization (Rathi & Roy, 2023; Chowdhury et al., 2022), and extension to advanced network architectures such as spike-driven transformers (Yao et al., 2023). Notably, Chowdhury et al. (2022) propose temporal pruning to gradually reduce the number of timesteps to successfully train SNNs with as little as a single timestep, despite a noticeable accuracy degradation (up to 4%).

All these works differ from ours as they are restricted to solely scaling the temporal dimension T in order to achieve different accuracy-latency tradeoffs given a fixed neural network architecture. To address this limitation, our approach introduces a new dimension S using our one-hot multi-level LIF (M-LIF) neuron model to improve the accuracy-latency tradeoff space.

2.4 Quantized-Activation ANNs

As the number of timesteps converges to one, conventional SNNs become closely related to binary activated artificial neural networks (BNNs) (Wang et al., 2020; Rastegari et al., 2016) as both use binary activations to perform an inference over a single timestep. Chowdhury et al. (2022) discuss that they are in fact distinct. While SNNs quantize outputs to spikes (*i.e.* $\{0, 1\}$), BNNs quantize outputs to be ± 1. Unlike BNNs which use non-linear activation functions where the firing threshold is zero, the firing threshold is learnable in SNNs. Chowdhury et al. (2022) observe that this allows SNNs to outperform BNNs in terms of accuracy and scale better to larger datasets such as ImageNet. Moreover, LIF enables SNNs to extend the same network for sequential processing unlike BNNs. Similarly, our one-hot M-LIF SNNs become closely related to log-quantized-activation ANNs (LQ-ANNs) (Zhou et al., 2016; Yin et al., 2019) as timesteps converge to one, but remain distinct for analogous reasons. A discussion regarding similarities and differences is provided in Section 3.2.

2.5 Quantized-Activation SNNs

Prior SNN works have explored the interaction between activation bit-width and timesteps (Xiao et al., 2019; Miao et al., 2018; Wang & Zhang, 2023; Feng et al., 2022; Wang et al., 2023; Li & Zeng, 2022). Multi-spike (Xiao et al., 2019; Miao et al., 2018) and multi-threshold (Wang & Zhang, 2023; Feng et al., 2022) neurons increase activation precision via uniform quantization but raise energy costs, disrupting the multiplication-free nature of traditional SNNs. Xiao et al. (2019) propose parallel and cascade multi-threshold (MT) models to enhance activation precision per timestep. The parallel-MT model uses multiple threshold values, summing spikes from all firing lanes at each timestep, while the cascade-MT model processes them sequentially. In contrast, our one-hot M-LIF neuron employs a single threshold and power-of-two multiples to fire one spike lane per timestep, a learned constraint distinct from prior methods. This enables efficient FP32 weight exponent updates via a single INT8 addition as discussed in Section 4, unlike the multiplication demands of uniformly quantized outputs. Miao et al. (2018) utilize non-multiplicative thresholds and focuses on residual networks without results on larger datasets like ImageNet or advanced architectures like spike-driven transformers, while also lacking energy analyses. Burst-spike models (Wang et al., 2023; Li & Zeng, 2022) enhance precision by increasing spike rates per timestep but struggle to scale down to single-timestep processing ($T = 8$ timesteps on ImageNet). Our one-hot M-LIF neuron reduces timestep requirements to $T = 1$, significantly cutting memory access overhead.

3 Proposed Multi-level LIF-based SNNs: M-LIF SNNs

3.1 Multi-level LIF Model

Our goal is to reduce the number of timesteps T during SNN inference while improving accuracy and maintaining the low-spike rates of traditional SNNs. By reducing the timesteps T while maintaining low spike rates, we can consequently decrease the energy overhead associated with multi-timestep

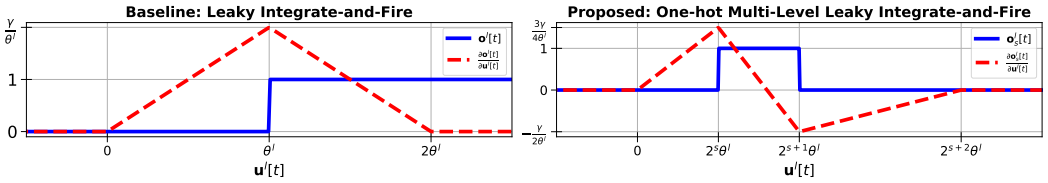

Figure 2: Output activation and surrogate gradient functions for conventional LIF neuron model (left) and one-hot multi-level LIF neuron model for $0 \leq s < S_o - 1$ (right).

processing. To do so, our proposed multi-level leaky integrate-and-fire (M-LIF) neuron model still uses a single membrane potential per output neuron while extending the range of neuron outputs to more than just binary representations. It employs a new dimension S to represent hidden layer outputs (inputs) as a set of S_o (S_i) binary-weighted spike lanes. We denote $\boldsymbol{o}_s^l[t]$ to be the output vector of binary spike lane s in layer l at timestep t. Each spike lane s is weighted by 2^s resulting in a non-binary output range for the neuron output $\mathbf{o}^l[t]$. As a result, Equation 1 needs to be modified to combine weighted spike lanes prior to updating the membrane potential as follows

$$\begin{aligned}\boldsymbol{u}^l[t] &= \beta^l \boldsymbol{u}^l[t-1] + \sum_{s=0}^{S_i-1} 2^s \boldsymbol{W}^l \boldsymbol{o}_s^{l-1}[t] - \sum_{s=0}^{S_o-1} 2^s \theta^l \boldsymbol{o}_s^l[t-1] \\ &= \beta^l \boldsymbol{u}^l[t-1] + \boldsymbol{W}^l \boldsymbol{o}^{l-1}[t] - \theta^l \boldsymbol{o}^l[t-1].\end{aligned} \tag{6}$$

We also define $\omega \leq S_o$ as the maximum number of simultaneously firing output spike lanes in any given timestep. With all spike lanes sharing the same firing threshold and membrane potential, it is non-trivial to devise a firing mechanism with $\leq \omega$ concurrent firing lanes. Section 3.2 discusses the practical one-hot case we propose for SNNs where $\omega = 1$.

3.2 One-hot Multi-level LIF Model

In the one-hot M-LIF model, $\omega = 1$ and *only one of the S_o output spike lanes fires in any given timestep*. The threshold mechanism is given by

$$\boldsymbol{o}_s^l[t] = \begin{cases} 1, & \text{if } \left((2^s < \frac{\boldsymbol{u}^l[t]}{\theta^l} \leq 2^{s+1}) \wedge (0 \leq s < S_o - 1)\right) \vee \left((2^s < \frac{\boldsymbol{u}^l[t]}{\theta^l}) \wedge (s = S_o - 1)\right) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Figure 1 depicts the difference between the conventional LIF model and our proposed one-hot ($\omega = 1$) M-LIF model given $S_o = 4$. Instead of having only one output (input) spiking signal, an M-LIF neuron has multiple ($S_o = 4$) binary-weighted output (input) spike lanes. With the one-hot constraint, only a single spike lane fires at any given timestep. In this example, the membrane potential can increase by one of $S_o = 4$ possible levels in $\{\theta^l, 2\theta^l, 4\theta^l, 8\theta^l\}$ from one timestep to the next, as opposed to the single level θ^l in the conventional LIF model. The output spike lanes are one-hot, meaning that the output range is also subdivided into $S_o = 4$ non-overlapping decision boundaries using the binary weight of each spiking lane as shown in Equation 8. Therefore, for a single activation channel case, we have $\sum_{s=0}^{S_o-1} 2^s \boldsymbol{o}_s^l[t] = \boldsymbol{o}^l[t] \in \{0, 1, 2, 4, \ldots, 2^{S_o-1}\}$.

$$\begin{aligned} \boldsymbol{o}_0^l[t] &= \begin{cases} 1, & \text{if } 1 < \frac{\boldsymbol{u}^l[t]}{\theta^l} \leq 2 \\ 0, & \text{otherwise} \end{cases} \qquad \boldsymbol{o}_1^l[t] = \begin{cases} 1, & \text{if } 2 < \frac{\boldsymbol{u}^l[t]}{\theta^l} \leq 4 \\ 0, & \text{otherwise} \end{cases} \\ \boldsymbol{o}_2^l[t] &= \begin{cases} 1, & \text{if } 4 < \frac{\boldsymbol{u}^l[t]}{\theta^l} \leq 8 \\ 0, & \text{otherwise} \end{cases} \qquad \boldsymbol{o}_3^l[t] = \begin{cases} 1, & \text{if } 8 < \frac{\boldsymbol{u}^l[t]}{\theta^l} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

Note that by setting $S_i = S_o = 1$ (i.e., single lane), Equations (6 - 7) simplify to Equations (1 - 2). Therefore, binary spiking SNNs can be considered as a special case of the proposed M-LIF scheme.

Surrogate Gradient for Back-Propagation Training Given the update to $\mathbf{o}^l[t]$ in Equation 7, the surrogate gradient is extended from the LIF case to the proposed one-hot M-LIF neurons. For

example, Figure 2 illustrates the differences in an updated triangular surrogate gradient for the single activation channel case when $0 \leq s < S_o - 1$. Each spike lane is now a window function of $\mathbf{u}^l[t]$ as opposed to a step function. As a result, the surrogate gradient becomes the difference of two triangular sub-gradients, one for each of the rising and falling window edges. An additional example for the derivative of the sigmoid function is included in the Appendix.

$$\frac{\partial \mathbf{o}^l[t]}{\partial \mathbf{u}^l[t]} = \sum_{s=0}^{S_o-1} 2^s \frac{\partial \mathbf{o}_s^l[t]}{\partial \mathbf{u}^l[t]}$$

$$\frac{\partial \mathbf{o}_s^l[t]}{\partial \mathbf{u}^l[t]} = \begin{cases} \text{diag}\left(\frac{\gamma}{2^s\theta^l} \max\{0, 1 - |\frac{\mathbf{u}^l[t]}{2^s\theta^l} - 1|\}\right), & \text{if } s = S_o - 1 \\ \text{diag}\left(\sum_{a=0}^{1} (-1)^a \frac{\gamma}{2^{s+a}\theta^l} \max\{0, 1 - |\frac{\mathbf{u}^l[t]}{2^{s+a}\theta^l} - 1|\}\right), & \text{if } 0 \leq s < S_o - 1 \end{cases}$$

Discussion As $T \to 1$ (i.e., unit timestep inference), Equations (6 - 7) can be rewritten as

$$\boldsymbol{u}^l = \sum_{s=0}^{S_i-1} 2^s \boldsymbol{W}^l \boldsymbol{o}_s^{l-1} = \boldsymbol{W}^l \boldsymbol{o}^{l-1} \tag{9}$$

where

$$\boldsymbol{o}^l = 2^{\text{clip}\left(\lfloor \log_2\left(\frac{\boldsymbol{u}^l}{\theta^l}\right)\rfloor,\ 0,\ S_o\right)}, \text{ and } \text{clip}(x, v, z) = \begin{cases} -\infty, & \text{if } x \leq v \\ z - 1, & \text{if } x \geq z \\ x, & \text{otherwise} \end{cases}.$$

From this, an observable parallel can be drawn between our one-hot M-LIF SNNs with unit timestep ($T = 1$) and $\log$ quantized-activation ANNs (LQ-ANNs) (Miyashita et al., 2016; Lee et al., 2017). While one-hot M-LIF-based SNNs are trained in a single phase, LQ-ANN training is performed in two phases per epoch. First, using the entire training dataset and full precision inference, a percentile value α of each layer's input activation distribution is recorded. Second, a straight-through estimator is typically applied to approximate the gradient with respect to quantized activations. Using b bits and assuming a ReLU activation function, the neuron output in an LQ-ANN is given by Equation 10. While both one-hot M-LIF SNNs with unit timestep ($T = 1$) and LQ-ANNs share commonalities, they remain slightly distinct. As highlighted by Equations (9 - 10), they namely differ in their choices of firing threshold and their final output value ranges. They also differ in their training methods and abilities to extend to sequential processing (i.e., $T > 1$).

$$\boldsymbol{o}^l = \text{ReLU}(\boldsymbol{W}^l \tilde{\boldsymbol{o}}^{l-1}), \quad \tilde{\boldsymbol{o}}^{l-1} = \begin{cases} \alpha^l 2^{\text{clip}\left(\lfloor \log_2\left(\frac{\boldsymbol{o}^{l-1}}{\alpha^l}\right)\rceil,\ 1 - 2^b,\ 1\right)}, & \text{if } \boldsymbol{o}^{l-1} \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

4 Energy Consumption Estimation

We evaluate the inference energy of our approach based on the approach in Chowdhury et al. (2022); Yao et al. (2023). In conventional SNNs, 32-bit floating-point (FP32) additions replace the FP32 multiply-and-accumulates (MACs) of ANNs except in the first layer which uses direct encoded inputs. For one-hot M-LIF SNNs ($\omega = 1$), inputs (outputs) are restricted to powers of 2, and multiplying by a power of 2 corresponds to adjusting the 8-bit integer (INT8) exponent of the FP32 multiplicand (see Appendix). Therefore, scaling the intermediate FP32 membrane potential by 2^s during integration in Equation 6 corresponds to increasing or decreasing its exponent in the INT8 format. According to Horowitz (2014), an INT8 addition consumes $30\times$ less energy than a FP32 addition, hence the overhead of scaling in M-LIF SNNs is negligible and FP32 additions dominate the energy consumption of one-hot M-LIF SNNs.

It is important to note that due to the one-hot constraint, the overall spiking rate (and consequently, the number of additions) per layer per timestep in one-hot M-LIF SNNs is not necessarily higher than that of conventional SNNs, even though one-hot M-LIF SNNs have multiple spiking lanes per

Table 1: Iso-architecture comparison with SNNs for static image classification on CIFAR and ImageNet. * denotes self-implementation results.

Dataset	Architecture	Method	S	T	Accuracy (%)	δ	Comp Energy (μJ)
CIFAR10	ResNet20	ANN*	-	1	94.31	1.0	9.97E+02
		DIET-SNN (Rathi & Roy, 2023)	1	5	91.78	6.3	1.58E+02
		Temporal Pruning (Chowdhury et al., 2022)	1	1	91.10	16.32	6.11E+01
		1-hot M-LIF (ours)	3	1	**93.19**	**18.10**	**5.55E+01**
	VGG16	ANN*	-	1	94.43	1.0	1.56E+03
		DIET-SNN (Rathi & Roy, 2023)	1	5	92.70	12.4	1.26E+02
		Temporal Pruning (Chowdhury et al., 2022)	1	1	93.05	**33.0**	**4.72E+01**
		BANN (Datta et al., 2024)	1	1	**93.44**	25.08	6.22E+01
		1-hot M-LIF (ours)	3	1	93.34	29.73	5.24E+01
	Transformer-2-512	Spike-Driven Transformer (Yao et al., 2023)*	1	4	95.6	-	4.60E+02
		Spike-Driven Transformer (Yao et al., 2023)*	1	2	94.7	-	2.80E+02
		Spike-Driven Transformer (Yao et al., 2023)*	1	1	94.5	-	**1.92E+02**
		1-hot M-LIF (ours)	3	4	**95.9**	-	1.47E+03
		1-hot M-LIF (ours)	3	2	95.5	-	4.84E+02
		1-hot M-LIF (ours)	3	1	95.4	-	2.59E+02
CIFAR100	ResNet20	ANN*	-	1	67.10	1.0	9.97E+02
		DIET-SNN (Rathi & Roy, 2023)	1	5	**64.07**	6.6	1.51E+02
		Temporal Pruning (Chowdhury et al., 2022)	1	1	63.30	**15.35**	**6.50E+01**
		1-hot M-LIF (ours)	3	1	63.80	14.05	7.10E+01
	VGG16	ANN*	-	1	74.50	1.0	1.56E+03
		DIET-SNN (Rathi & Roy, 2023)	1	5	69.97	12.1	1.29E+02
		Temporal Pruning (Chowdhury et al., 2022)	1	1	70.15	**29.24**	**5.34E+01**
		1-hot M-LIF (ours)	3	1	**72.59**	23.63	6.60E+01
	Transformer-2-512	Spike-Driven Transformer (Yao et al., 2023)*	1	4	78.4	-	5.87E+02
		Spike-Driven Transformer (Yao et al., 2023)*	1	2	76.6	-	**1.90E+02**
		Spike-Driven Transformer (Yao et al., 2023)*	1	1	75.8	-	2.21E+02
		1-hot M-LIF (ours)	3	4	**78.9**	-	1.68E+03
		1-hot M-LIF (ours)	3	2	78.3	-	8.20E+02
		1-hot M-LIF (ours)	3	1	78.2	-	4.78E+02
ImageNet	VGG16	ANN*	-	1	72.56	1.0	7.12E+04
		DIET-SNN (Rathi & Roy, 2023)	1	5	69.00	11.7	6.09E+03
		Temporal Pruning (Chowdhury et al., 2022)	1	1	69.00	**24.61**	**2.89E+03**
		BANN (Datta et al., 2024)	1	1	68.00	20.94	3.40E+03
		1-hot M-LIF (ours)	3	1	**71.05**	20.20	3.37E+03
	Transformer-8-512	Spike-Driven Transformer (Yao et al., 2023)	1	1	71.68	-	**1.13E+03**
		Spike-Driven Transformer (Yao et al., 2023)	1	4	74.57	-	4.50E+03
		1-hot M-LIF (ours)	3	1	**75.33**	-	3.64E+03

neuron. This gives one-hot M-LIF SNNs the opportunity to learn more within a single timestep without increasing the computational complexity and energy compared to conventional SNNs.

It is known that memory access energy can be significantly higher than compute energy (Horowitz, 2014; Han et al., 2015) and the number of memory accesses scales linearly with the number of timesteps in SNNs (Chowdhury et al., 2022). However, estimating memory energy improvements would depend on hardware architecture and system configuration. Therefore, as noted in Chowdhury et al. (2022), we are restricting our attention to the computational energy benefits, δ defined in Equation 11 (Chowdhury et al., 2022), of one-hot M-LIF SNNs and conventional SNNs over ANNs. As a result, we consider δ to be an optimistic energy gain estimate when $T > 1$. Note that when $T = 1$, memory requirements are identical for both SNNs and ANNs. When an iso-architecture ANN does not exist as in the case of spike-driven transformers (Yao et al., 2023) due to unique mechanisms such as spike-driven attention, we compare directly using the computational energy E.

$$\delta = \frac{E_{\text{ANN}}}{E_{\text{SNN}}} = \frac{\sum_{l=1}^{L} \#\text{ANN}_{\text{ops},l} \times 4.6}{\#\text{SNN}_{\text{ops},1} \times 4.6 + \sum_{l=2}^{L} \#\text{SNN}_{\text{ops},l} \times 0.9} \tag{11}$$

In Equation 11, $\#\text{ANN}_{\text{ops},l}$ is the number of operations per layer in an ANN, $\#\text{SNN}_{\text{ops},l} = r_l \times \#\text{ANN}_{\text{ops},l}$ is the number of operations per layer in an SNN, and r_l denotes the spike rate which is the proportion of non-zero input spikes per layer over all spike lanes and timesteps in layer l. Based on Horowitz (2014), we set the relative MAC and addition energy to 4.6pJ and 0.9pJ, respectively.

Table 2: Iso-architecture comparison with LQ-ANNs for static image classification on CIFAR and ImageNet. * denotes self implementation results.

Dataset	Architecture	Method	S	T	b	Accuracy (%)	δ	Comp Energy (μJ)
CIFAR10	ResNet20	LQ-ANN*	-	1	2	93.03	**21.60**	**4.61E+01**
			-	1	3	**93.59**	12.02	8.29E+01
		1-hot M-LIF (ours)	2	1	-	92.61	20.78	4.80E+01
			3	1	-	93.19	18.10	5.51E+01
CIFAR100	VGG16	LQ-ANN*	-	1	2	71.24	25.28	6.17E+01
			-	1	3	**72.87**	15.29	1.02E+02
		1-hot M-LIF (ours)	2	1	-	71.32	**27.30**	**4.88E+01**
			3	1	-	72.59	23.63	5.66E+01
ImageNet	VGG16	LQ-ANN*	-	1	3	62.97	16.95	4.20E+03
			-	1	4	67.85	14.18	5.02E+03
		1-hot M-LIF (ours)	3	1	-	**71.05**	**20.20**	**3.37E+03**

5 Experiments & Results

We validate our one-hot M-LIF model and compare the performance and inference energy of our one-hot M-LIF SNNs with existing SNN works on both static and dynamic image classification tasks. Our proposed neuron model can be integrated into existing SNN training methodologies. We compare against the hybrid training methods (Chowdhury et al., 2022; Rathi & Roy, 2023) for static tasks (Section 5.1) and the temporal efficient training method Deng et al. (2022) for dynamic vision tasks (Section 5.2). We also evaluate the impact of one-hot M-LIF on more complex SNN-based high performance models such as the spike-driven transformer (Yao et al., 2023). As in prior works, we employ direct input encoding for static tasks such that the input layer is fed with full-precision pixels. We also fix all layers to use the same number of (input) output spike lanes, S, as this reduces the number of hyperparameters. The source code is available at: *to be released upon acceptance*.

5.1 Static Image Classification

5.1.1 Implementation Details

We apply hybrid direct training as described in Rathi & Roy (2023); Chowdhury et al. (2022) to evaluate the accuracy of our approach on CIFAR10, CIFAR100, and ImageNet using VGG16 and ResNet20. We train an ANN with batch-norm (Ioffe & Szegedy, 2015) and subsequently fuse the batch-norm parameters with the weights of the corresponding layer. We then copy the weights of the pre-trained ANN to an iso-architecture one-hot M-LIF SNN and use the 90-th percentile of the input activation distribution as each layer's threshold θ^l. The SNN is then trained using BPTT but without temporal pruning. For spike-driven transformer, we evaluate our approach on CIFAR10, CIFAR100, and ImageNet using Transformer-2-512 and Transformer-8-512 by replacing all LIF neurons with one-hot M-LIF neurons while using the same training methodology as in Yao et al. (2023). Note that Transformer-L-D represents a model with L encoder blocks and D channels. These networks are trained from scratch using BPTT without any pre-trained ANN initialization or batch norm fusion. Supplemental network architecture details and hyperparameters are discussed in the Appendix.

5.1.2 Comparison with SNNs

Table 1 compares the accuracy and inference energy of one-hot M-LIF SNNs with iso-architecture conventional SNNs. While our approach offers comparable or slightly lower energy benefits across most benchmarks, it consistently matches or exceeds conventional SNNs in accuracy. The one-hot constraint ensures energy usage comparable to conventional SNNs despite each M-LIF neuron having multiple spiking lanes, discovering new accuracy-energy tradeoff points. Prior work achieved 69% accuracy with a unit timestep on ImageNet using VGG16, while we reached 71.05% with $S = 3$ spike lanes. For spike-driven transformers, M-LIF SNNs boost accuracy by up to 3% on ImageNet compared to LIF counterparts, consuming slightly more energy for a given T but achieving better tradeoffs. This is the case of ($T = 1$, $S = 3$) one-hot M-LIF spike-driven transformer, which achieves comparable or better accuracy to ($T = 4$) LIF on CIFAR100 and ImageNet with $4\times$ less memory access energy (which can dominate overall energy as discussed in Section 4) due to multi-timestep processing.

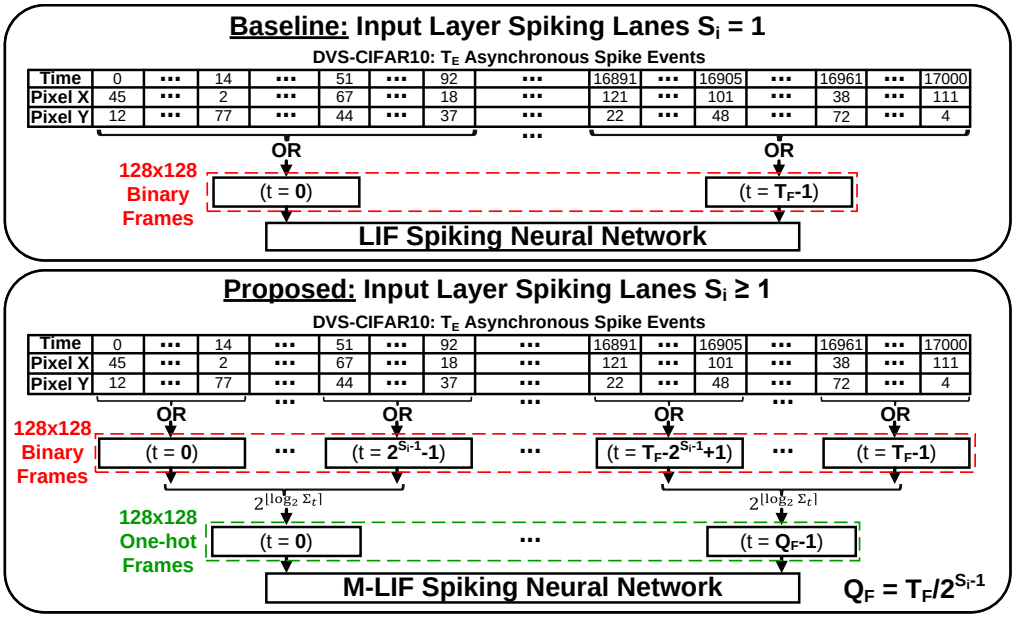

Figure 3: Workflow of multi-level input layer encoding for dynamic vision tasks.

5.1.3 Comparison with LQ-ANNs

As discussed in Section 3.2, LQ-ANNs and unit timestep ($T = 1$) M-LIF SNNs remain distinct while both perform inference using a single timestep. Here, we compare the accuracy and inference energy of b-bit LQ-ANNs and M-LIF-based SNNs using S spike lanes as shown in Table 2. We observe that M-LIF SNNs perform on par or better than LQ-ANNs in terms of accuracy and inference energy. For CIFAR10, we observe similar accuracy and energy benefits to LQ-ANNs. On the other hand, for CIFAR100, we note that one-hot M-LIF SNNs are up to 54% more energy efficient than LQ-ANNs with comparable accuracy. Finally, our approach scales much better on a large challenging dataset such as ImageNet. This gain can be primarily attributed to threshold parameter learning for SNNs as in Rathi & Roy (2023) and the final output value ranges learned during training.

5.2 Dynamic Image Classification

5.2.1 Implementation Details

For the dynamic image classification task where SNN accuracy is generally superior than that of ANNs, we apply temporal efficient training similar to Deng et al. (2022) using our one-hot M-LIF neuron. Here, we train from scratch using BPTT without any pre-trained ANN initialization or batch norm fusion. We perform experiments on DVS-CIFAR10 (Li et al., 2017) (converted from CIFAR10) which is one of the most challenging mainstream dynamic vision datasets. It has 10k images with size 128×128. Following prior works, we reduce the spatial resolution to 48×48, and split the dataset into 9k training and 1k test images (Samadzadeh et al., 2023). We also apply data augmentation techniques such as random horizontal flip and random roll within 5 pixels (Li et al., 2022). For all experiments, we use the VGGSNN architecture (Deng et al., 2022) using 300 epochs, the Adam optimizer with learning rate $\lambda = 0.001$ and a cosine annealing scheduler with 0 decay.

5.2.2 Multi-level Input Layer Encoding

For DVS-CIFAR10, direct input encoding is not applicable as the dataset consists of events recorded using a dynamic vision sensor. The adopted methodology described in Samadzadeh et al. (2023) to prepare the data for SNN training is to split and convert the stream of T_E events into T_F binary frames as depicted in Figure 3 (top). In Deng et al. (2022), a VGGSNN is trained with $T_F = 10$ and a top-1 accuracy of 83.17%. However, M-LIF SNNs are not limited to single spike lanes at the input layer. Therefore, we allow $T_F \neq 10$ and incorporate an additional data preparation step to perform multi-level input layer encoding as depicted in Figure 3 (bottom). After obtaining the

Table 3: Comparison with prior works for dynamic image classification on DVS-CIFAR10. † denotes data augmentation. $*$ denotes self-implementation results.

Method	Architecture	S	T_F	Q_F	Accuracy (%)	Comp Energy (μJ)
STBP-tdBN (Zheng et al., 2021)	ResNet-19	1	10	-	67.8	-
Streaming Rollout (Kugele et al., 2020)	DenseNet	1	10	-	66.8	-
Conv3D (Wu et al., 2021)	LIAF-Net	1	10	-	71.70	-
LIAF (Wu et al., 2021)	LIAF-Net	1	10	-	70.40	-
Dspike (Li et al., 2021b)	ResNet-18	1	10	-	75.4	-
RecDis-SNN (Guo et al., 2022)	ResNet-19	1	10	-	72.4	-
Spike-driven Transformer† (Yao et al., 2023)	Transformer-2-256	1	16	-	80.0	-
PLIF (Fang et al., 2021)	VGGSNN	1	20	-	74.8	-
SEENN-II† (Li et al., 2024)	VGGSNN	1	4.5	-	82.6	-
SEENN-I† (Li et al., 2024)	VGGSNN	1	**2.5**	-	**77.6**	-
TET (Deng et al., 2022)	VGGSNN	1	10	-	77.3	-
TET$*$† (Deng et al., 2022)	VGGSNN	1	**10**	-	**83.1**	3.8E+02
TET$*$† (Deng et al., 2022)	VGGSNN	1	5	-	78.0	1.9E+02
TET$*$† (Deng et al., 2022)	VGGSNN	1	**3**	-	**74.7**	**1.2E+02**
1-hot M-LIF† (ours)	VGGSNN	4	-	**10**	**84.7**	**3.5E+02**
1-hot M-LIF† (ours)	VGGSNN	3	-	10	84.3	3.4E+02
1-hot M-LIF† (ours)	VGGSNN	4	-	5	83.3	1.8E+02
1-hot M-LIF† (ours)	VGGSNN	3	-	5	83.0	1.7E+02
1-hot M-LIF† (ours)	VGGSNN	4	-	**3**	**82.5**	**1.1E+02**
1-hot M-LIF† (ours)	VGGSNN	3	-	3	79.8	9.0E+01

T_F binary frames, we combine every 2^{S_i-1} consecutive frames into a one-hot frame resulting in $Q_F = T_F/2^{S_i-1}$ frames of one-hot S_i spike lanes. This enables M-LIF SNNs to limit accuracy degradation after reducing Q_F below the number of timesteps T_F.

5.2.3 Comparison with SNNs

We compare against existing works on DVS-CIFAR10 in Table 3. The compute energy is calculated using $E_{\text{SNN}} = \sum_{l=1}^{L} \#\text{SNN}_{\text{ops},l} \times 0.9$ pJ, where $\#\text{SNN}_{\text{ops},l}$ is defined in Section 4 and 0.9 pJ is the energy of addition (Horowitz, 2014). We achieve an accuracy of 84.7% using 10 timesteps and 4 spike lanes per neuron. This is also the first SNN work to achieve 82.5% accuracy on DVS-CIFAR10 using 3 timesteps and 4 spike lanes compared to the best prior work (Li et al., 2024) which can only achieve 82.6% using 4.5 timesteps and 77.6% using 2.5 timesteps. These improvements in accuracy stem primarily from introducing the S dimension. By reducing Q_F, not only do we improve the computational energy by $3.45\times$, we also reduce memory access energy which scales linearly with timesteps and can be significantly higher than compute energy (Chowdhury et al., 2022). Table 3 also shows the impact of scaling Q_F and S on accuracy. By increasing S for a fixed Q_F, we are able to recover accuracy degradation unlike prior works which are limited by solely scaling T_F. By increasing S and Q_F, we can scale the accuracy to even higher than conventional SNNs.

6 Conclusion

SNNs hold promise as an energy-efficient alternative to traditional ANNs. However, achieving an optimal balance in the accuracy-energy tradeoff by adjusting latency remains a significant challenge for widespread deployment. To that end, we introduce the dimension of spike lanes to conventional SNNs using a novel M-LIF neuron model without latency and computational complexity overhead. The proposed model represents the inputs and outputs of hidden layers as a set of one-hot binary-weighted spike lanes. Using our one-hot M-LIF neuron model, we are able to find new and better tradeoff points for both static and dynamic vision tasks. In particular, our one-hot M-LIF-based SNNs achieve a top-1 accuracy of 71.05% on ImageNet using VGG16 and enhance the computational efficiency by $20\times$. One-hot M-LIF neurons also improve the accuracy-latency tradeoff for advanced network architectures such as spike-driven transformers ($> 3\%$ higher accuracy with $4\times$ fewer timesteps on ImageNet). For dynamic vision tasks, such as image classification using dynamic vision sensor data, our one-hot M-LIF SNNs retain higher accuracy (82.5%) when scaling down to fewer timesteps (3) on CIFAR10-DVS thus providing better energy efficiency.

7 Reproducibility Statement

The authors make the following efforts for reproducibility: 1) We submit an anonymized repository containing code used to run our experiments in the supplementary material, 2) we provide the detailed settings and hyperparameters in Sections 5 and A.3.

References

Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in neural information processing systems*, 31, 2018.

Tong Bu, Wei Fang, Jianhao Ding, PengLin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ann-snn conversion for high-accuracy and ultra-low-latency spiking neural networks. *arXiv preprint arXiv:2303.04347*, 2023.

Anthony N Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, 95:1–19, 2006.

Sayeed Shafayet Chowdhury, Nitin Rathi, and Kaushik Roy. Towards ultra low latency spiking neural networks for vision and sequential tasks using temporal pruning. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pp. 709–726, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-20082-3. doi: 10.1007/978-3-031-20083-0_42.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 113–123, 2019.

Gourav Datta, Zeyu Liu, and Peter Anthony Beerel. Can we get the best of both binary neural networks and spiking neural networks for efficient computer vision? In *The Twelfth International Conference on Learning Representations*, 2024.

Lei Deng, Yujie Wu, Xing Hu, Ling Liang, Yufei Ding, Guoqi Li, Guangshe Zhao, Peng Li, and Yuan Xie. Rethinking the performance comparison between snns and anns. *Neural networks*, 121:294–307, 2020.

Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. *arXiv preprint arXiv:2103.00476*, 2021.

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

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

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)*, pp. 1–8. ieee, 2015.

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

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. *arXiv preprint arXiv:2210.06386*, 2022.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 326–335, 2022.

Bing Han, Abhronil Sengupta, and Kaushik Roy. On the energy benefits of spiking deep neural networks: A case study. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 971–976, 2016. doi: 10.1109/IJCNN.2016.7727303.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Mark 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)*, pp. 10–14. IEEE, 2014.

Eric Hunsberger and Chris Eliasmith. Spiking deep networks with lif neurons. *arXiv preprint arXiv:1510.08829*, 2015.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

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

Edward H Lee, Daisuke Miyashita, Elaina Chai, Boris Murmann, and S Simon Wong. Lognet: Energy-efficient neural networks using logarithmic computation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5900–5904. IEEE, 2017.

Juan Antonio Leñero-Bardallo, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. A 3.6μs latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE Journal of Solid-State Circuits*, 46(6):1443–1455, 2011.

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

Yang Li and Yi Zeng. Efficient and accurate conversion of spiking neural network with burst spikes. *arXiv preprint arXiv:2204.13271*, 2022.

Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In *International conference on machine learning*, pp. 6316–6325. PMLR, 2021a.

Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in Neural Information Processing Systems*, 34:23426–23439, 2021b.

Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic data augmentation for training spiking neural networks. In *European Conference on Computer Vision*, pp. 631–649. Springer, 2022.

Yuhang Li, Tamar Geller, Youngeun Kim, and Priyadarshini Panda. Seenn: Towards temporal spiking early exit neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

Yu Miao, Huajin Tang, and Gang Pan. A supervised multi-spike learning algorithm for spiking neural networks. In *2018 international joint conference on neural networks (IJCNN)*, pp. 1–7. IEEE, 2018.

Daisuke Miyashita, Edward H Lee, and Boris Murmann. Convolutional neural networks using logarithmic data representation. *arXiv preprint arXiv:1603.01025*, 2016.

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, 2019.

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

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pp. 525–542. Springer, 2016.

Nitin Rathi and Kaushik Roy. Diet-snn: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. In *IEEE Transactions on Neural Networks and Learning Systems*, pp. 3174–3182, 2023. doi: 10.1109/TNNLS.2021.3111897.

Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017. doi: 10.1162/neco_a_00990.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

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.

Ali Samadzadeh, Fatemeh Sadat Tabatabaei Far, Ali Javadi, Ahmad Nickabadi, and Morteza Haghir Chehreghani. Convolutional spiking neural networks for spatio-temporal feature extraction. *Neural Processing Letters*, pp. 1–17, 2023.

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

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

Weihao Tan, Devdhar Patel, and Robert Kozma. Strategy and benchmark for converting deep q-networks to event-driven spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9816–9824, 2021.

Peisong Wang, Xiangyu He, Gang Li, Tianli Zhao, and Jian Cheng. Sparsity-inducing binarized neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 12192–12199, 2020.

Xiaoting Wang and Yanxiang Zhang. Mt-snn: enhance spiking neural network with multiple thresholds. *arXiv preprint arXiv:2303.11127*, 2023.

Ziqing Wang, Yuetong Fang, Jiahang Cao, and Renjing Xu. Bursting spikes: Efficient and high-performance snns for event-based vision. *arXiv preprint arXiv:2311.14265*, 2023.

Zhenzhi Wu, Hehui Zhang, Yihan Lin, Guoqi Li, Meng Wang, and Ye Tang. Liaf-net: Leaky integrate and analog fire network for lightweight and efficient spatiotemporal information processing. *IEEE Transactions on Neural Networks and Learning Systems*, 33:6249–6262, 2021.

Rong Xiao, Qiang Yu, Rui Yan, and Huajin Tang. Fast and accurate classification with a multi-spike learning algorithm for spiking neurons. In *IJCAI*, pp. 1445–1451, 2019.

Man Yao, JiaKui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 64043–64058. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ca0f5358dbadda74b3049711887e9ead-Paper-Conference.pdf.

Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. *arXiv preprint arXiv:1903.05662*, 2019.

Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11062–11070, 2021.

Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

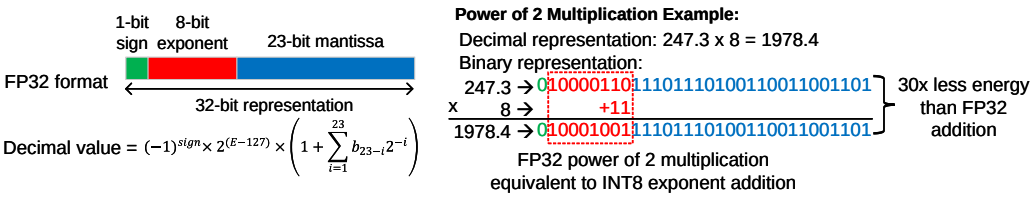

Figure 4: 32-bit floating-point format and power of two multiplication details where b_i is the i-th bit, $sign$ is the most significant bit, and E is the 8-bit exponent.

A Appendix

A.1 Surrogate Gradient

Adapting the derivative of sigmoid function σ' to one-hot M-LIF is performed in a similar fashion to the triangular surrogate gradient described in the main text. The surrogate gradient becomes the difference of σ' sub-gradients, one for each of the rising and falling window edges of the firing function.

$$\frac{\partial \mathbf{o}^l[t]}{\partial \mathbf{u}^l[t]} = \sum_{s=0}^{S_o-1} 2^s \frac{\partial \mathbf{o}_s^l[t]}{\partial \mathbf{u}^l[t]}$$

$$\boldsymbol{x}^l(s) = \sigma\left(\alpha(\frac{\boldsymbol{u}^l[t]}{2^s\theta^l} - 1)\right)$$

$$\frac{\partial \mathbf{o}_s^l[t]}{\partial \mathbf{u}^l[t]} = \begin{cases} \text{diag}\left(\frac{\alpha}{2^s\theta^l}\left(1 - \boldsymbol{x}(s)\right)\boldsymbol{x}(s)\right), & \text{if } s = S_o - 1 \\ \text{diag}\left(\sum_{a=0}^{1}(-1)^a \frac{\alpha}{2^{s+a}\theta^l}\left(1 - \boldsymbol{x}(s+a)\right)\boldsymbol{x}(s+a)\right), & \text{if } 0 \le s < S_o - 1 \end{cases}$$

A.2 FP32 Power of Two Multiplication

As discussed in the main text, both M-LIF SNNs and LIF SNNs perform FP32 additions during integration to calculate the membrane potential at each timestep. For M-LIF SNNs, FP32 weights are scaled by a power of 2 prior to integration. As illustrated in Figure 4 which provides details regarding the FP32 number format, power of two multiplication corresponds to adjusting the 8-bit integer exponent which requires $30\times$ less energy than the FP32 addition (Horowitz, 2014). Hence, the overhead is negligible and M-LIF SNNs are considered to leverage additions instead of multiplications like their LIF counterparts.

A.3 Experimental Details

A.3.1 Static Image Classification

Datasets. We employ the CIFAR datasets (Krizhevsky & Hinton, 2009) which consist of 50k training and 10k test images. CIFAR10 contains 10 classes while CIFAR100 contains 100 classes. Standard data augmentation techniques are applied to training images such as padding by 4 pixels on each side, random horizontal flip and 32×32 crop by randomly sampling the padded image. During test, the original 32×32 images are used. We calculate the channel-wise mean and standard deviation of training images and use those values to normalize both training and test data. Contrary to other works such as Li et al. (2024), we do not apply augmentation techniques on CIFAR for CNN experiments such as Cutout (DeVries & Taylor, 2017) and AutoAugment (Cubuk et al., 2019) in order to faithfully compare with results in Chowdhury et al. (2022). We employ the same augmentation techniques on CIFAR when comparing with results in Yao et al. (2023).

We also employ the ImageNet dataset (Krizhevsky et al., 2012) which contains 1000 classes and consists of more than 1250k training and 50k test images. We employ the same augmentation techniques on ImageNet when comparing with results in Yao et al. (2023); Chowdhury et al. (2022).

Network Architectures. The VGG16 and ResNet20 architectures adopted for static image classification are taken from Chowdhury et al. (2022). Average pooling (2×2) is applied for all cases. The ResNet basic blocks use a 1×1 stride-2 convolutional layer shortcut path where the number of

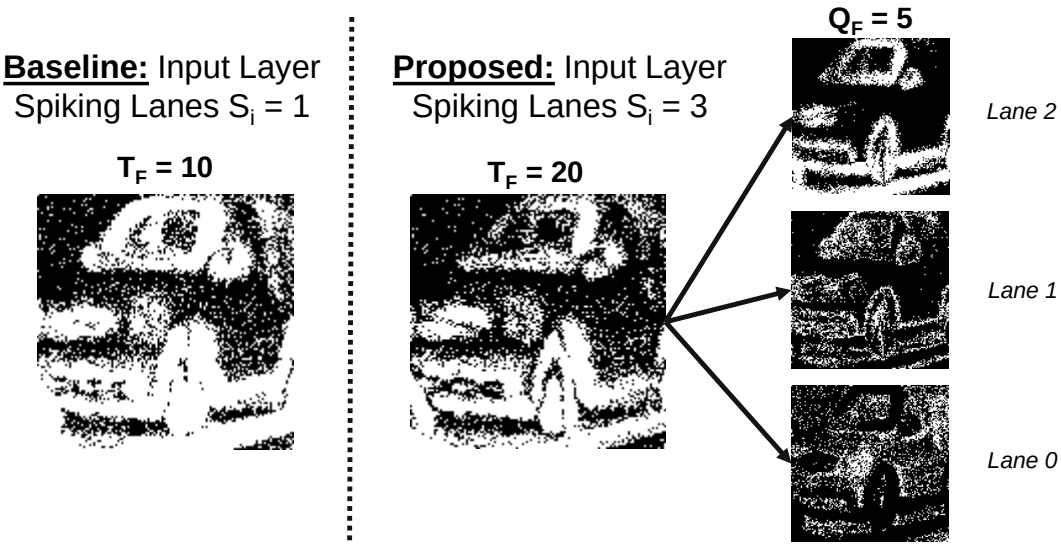

Figure 5: Example image in DVS-CIFAR10 using baseline input layer encoding with $T_F = 10$ binary frames and $S_i = 1$ spike lane (left) and the proposed multi-level input layer encoding with $Q_F = 5$ one-hot frames and $S_i = 3$ spike lanes (right).

input and output channels are different. The architectures of each network are summarized below where BB denotes a basic block (He et al., 2016), D denotes a dropout layer (probability ANN: 0, SNN: 0.2), AP2 denotes a (2×2) average pooling layer, (/2) denotes stride-2, oCk denotes a $(k \times k)$ stride-1 convolutional layer with o output filters, oFC denotes a fully-connected layer with o output filers, and n denotes the number of classes.

VGG16: {64C3, D, 64C3, AP2, 128C3, D, 128C3, AP2, 256C3, D, 256C3, D, 256C3, AP2, 512C3, D, 512C3, D, 512C3, AP2, 512C3, D, 512C3, D, 512C3, D, 4096FC, 4096FC, nFC}

ResNet20: {64C3, D, 64C3, D, 64C3, AP2, 64BB, 64BB, 128BB (/2), 128BB, 256BB (/2), 256BB, 512BB (/2), 512BB, nFC}

The spike-driven transformer architectures adopted for static image classification are taken from Yao et al. (2023). Transformer-L-D networks include spiking patch splitting followed by L spike-driven encoder blocks and a linear classification head. The encoder blocks consist of a spike-driven self attention layer followed by two multi-layer perceptron layers. D refers to the number of channels in the input to the encoder blocks.

Training Hyperparameters. To train ANNs, we use a stochastic gradient descent optimizer with weight decay of 0.0005 and momentum of 0.9. We initialize ANN weights using He initialization (He et al., 2015). We train the ResNet20 and VGG16 ANNs for CIFAR10 and CIFAR100 during 300 epochs with an initial learning rate of 0.01 that is divided by 10 at epochs 120, 180, and 240. We train VGG16 ANN for ImageNet during 90 epochs with an initial learning rate of 0.01 that is divided by 5 at epochs 40, 62, and 81. We also employ batch-norm during ANN training (Ioffe & Szegedy, 2015). We use a pre-trained ANN to initialize the parameters of a corresponding LQ-ANN and perform training using an Adam optimizer with a weight decay of 0.0005 for 50 epochs and an initial learning rate of 0.0001. The learning rate is divided by 5 at epochs 20, 30, and 40.

For iso-architecture M-LIF SNN initialization from a pre-trained ANN, we fuse the batch-norm parameters with the corresponding layer's parameters similar to Chowdhury et al. (2022). For static image classification with VGG16 and ResNet20, we use a triangular surrogate gradient with $\gamma = 0.3$ and learn firing thresholds and membrane leakages during training as per Chowdhury et al. (2022). The learning rate is shared among weights, firing thresholds, and membrane leakages unless otherwise noted. Upon initialization, we train M-LIF SNNs using an Adam optimizer with a weight decay of 0. ResNet20 M-LIF SNNs are trained on CIFAR10 for 150 epochs with an initial learning rate of 0.0001 that is divided by 5 at epochs 90, 120, and 135. ResNet20 M-LIF SNNs are trained on CIFAR100 for 600 epochs with a fixed learning rate of 0.01 for firing thresholds and an initial learning rate of 0.0002 that is divided by 2 at epochs 120, 240, 360, 450, and 540 for the remaining

Table 4: Ablation study varying number of spike lanes with fixed $T = 3$ timesteps for one-hot M-LIF VGGSNN on DVS-CIFAR10.

S	Accuracy (%)
1	74.7
2	78.6
3	79.8
4	82.5
5	82.5

Table 5: Ablation study varying number of spike lanes with fixed $T = 5$ timesteps for one-hot M-LIF VGGSNN on DVS-CIFAR10.

S	Accuracy (%)
1	78.0
2	81.5
3	83.0
4	83.3
5	82.9

parameters. VGG16 M-LIF SNNs are trained on CIFAR10 for 200 epochs. During the first 10 epochs, the learning rate is increased linearly from 0.00001 to 0.0001 and subsequently divided by 5 at epochs 62 and 150. VGG16 M-LIF SNNs are trained on CIFAR100 for 200 epochs with a fixed learning rate of 0.001 for firing thresholds and an initial learning rate of 0.0001 that is divided by 5 at epochs 62 and 150 for the remaining parameters. VGG16 M-LIF SNNs are also trained on ImageNet for 50 epochs. During the first 5 epochs, the learning rate is increased linearly from 0.00001 to 0.0001 and subsequently divided by 5 at epochs 30 and 40.

For spike-driven transformers, we adopt the same hyperparameters as LIF spike-driven transformers in Yao et al. (2023) when training their M-LIF counterparts for all experiments. We use a surrogate gradient based on the derivative of the sigmoid function with $\alpha = 4$. In order to make use of the same hyperparameter configuration, the outputs of M-LIF neurons are scaled to be in the range $[0, 1]$ but continue to be powers of 2 (*i.e.,* $\in 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, ...$). Finally, similar to (Yao et al., 2023), we use a fixed threshold for all M-LIF neurons in spike-driven transformer experiments instead of learning the threshold as in our experiments on other architectures.

A.3.2 Dynamic Image Classification

Dataset. We employ the DVS-CIFAR10 dataset with standard data augmentation techniques as described in Section 4.2 of the main paper. Figure 5 illustrates an example training image of a car in DVS-CIFAR10 using the baseline input layer encoding with $T_F = 10$ binary frames and $S_i = 1$ spike lane (left) and the proposed multi-level input layer encoding with $Q_F = 5$ one-hot frames and $S_i = 3$ spike lanes (right).

Network Architecture. The VGGSNN architecture employed was taken from Deng et al. (2022) and is described as VGGSNN: {64C3, 128C3 (/2), 256C3, 256C3 (/2), 512C3, 512C3 (/2), 512C3, 512C3 (/2), 10FC}.

Training Hyperparameters. In addition to the training settings mentioned in Section 4.2 of the main paper, we use $\gamma = 1.0$ for all SNN surrogate gradient scaling and fix firing thresholds and membrane leakages to 1.0 and 0.5, respectively, for all layers as per Deng et al. (2022).

A.4 Ablation Studies

We conducted an ablation study where we varied the number of spike lanes S while keeping the number of timesteps T fixed at 3 for VGGSNN on DVS-CIFAR10. We provide the results below

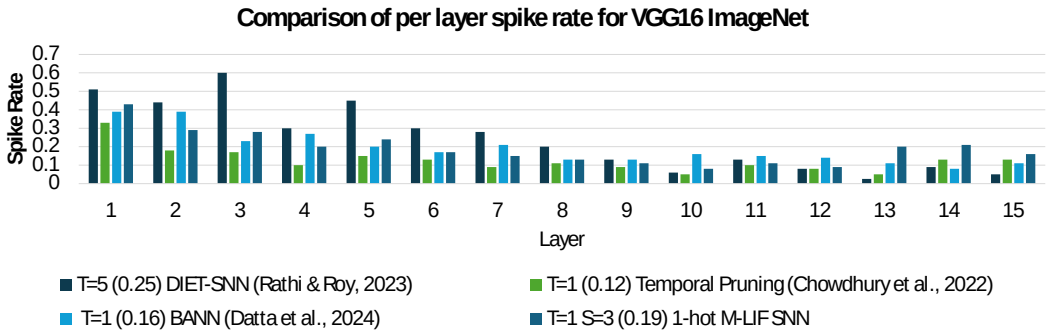

Figure 6: Comparison of per-layer spike rates for VGG16 on ImageNet.

Table 6: Impact of memory access energy for static image classification using VGG16 on ImageNet. $*$ denotes self-implementation results.

Method	S	T	Accuracy (%)	Comp Energy (μJ)	Mem Energy (μJ)	Total Energy (μJ)
ANN*	-	-	72.56	7.12E+04	7.81E+05	8.52E+05
DIET-SNN (Rathi & Roy, 2023)	1	5	69.00	6.09E+03	5.88E+04	6.49E+04
Temporal Pruning (Chowdhury et al., 2022)	1	1	69.00	2.89E+03	1.55E+04	1.84E+04
BANN (Datta et al., 2024)	1	1	68.00	3.40E+03	1.71E+04	2.05E+04
1-hot M-LIF (ours)	3	1	71.05	3.37E+03	2.21E+04	2.58E+04

in Table 4. Our findings indicate that as the number of spike lanes increases from $S = 1$ to $S = 4$, there is a significant improvement in accuracy, after which performance saturates. Additionally, when we increase the number of timesteps to $T = 5$ in Table 5, we see a similar performance saturation around $S = 4$ timesteps. We also note that the accuracy improvement due to spike lanes becomes less pronounced, but the overall accuracy ceiling improves. From these results, we conclude that both timesteps and spike lanes contribute to better model accuracy with the caveat that multi-timestep processing introduces significant energy overhead, particularly with respect to memory access as noted in Section 4.

A.5 Memory Access Energy Impact

Although defining a precise memory architecture requires detailed assumptions regarding weight, input, and partial output reuse, we have included an analysis based on the memory energy model provided in Datta et al. (2024, Appendix A.8.2) for convolutional neural networks. This model allows us to estimate the impact of memory energy by incorporating per-layer spike rates. We focused on VGG16 trained on ImageNet as prior works (Chowdhury et al., 2022; Rathi & Roy, 2023; Datta et al., 2024) provide detailed per-layer spike rates, and we used the same 45nm technology for energy modeling, allowing for consistent energy comparisons. Table 6 illustrates the impact of memory access energy on the overall energy comparison. One-hot M-LIF SNNs consume $33\times$ lower energy than ANNs, while maintaining higher accuracy (up to $> 3\%$) than prior unit-timestep SNN works. We also observe that multi-timestep processing introduces a noticeable energy overhead. In fact, we achieve higher accuracy compared to Rathi & Roy (2023) while consuming $2.5\times$ less total energy, the majority of which stems from the memory energy overhead of multi-timestep ($T = 5$) processing. Figure 6 provides per layer spike rates for VGG16 ImageNet of the models compared in Table 6.