# TOWARD GENERAL-PURPOSE SPIKING NEURONS FOR VISION AND LANGUAGE MODELING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Widely regarded as the third generation of neural networks, Spiking Neural Networks (SNNs) have attracted considerable attention due to their biological plausibility and energy efficiency. Recent advances in large models call for spiking neurons that combine high performance, adaptability, and training efficiency. In this work, we first present a functional perspective that provides general guidance for designing the next generation of spiking neurons. Guided by these principles, we propose the Adaptive Spiking Neuron (ASN), which incorporates trainable parameters to learn membrane-potential dynamics and enable adaptive firing. ASN adopts an integer-training and spike-inference paradigm that facilitates efficient SNN training. To further improve training stability, we introduce a normalized variant, the Normalized Adaptive Spiking Neuron (NASN). We evaluate the ASN family on 20 datasets spanning five task groups across vision and language, demonstrating its effectiveness and versatility. In particular, NASN exhibits promising scalability in spike-based language modeling, with increasingly pronounced advantages as model and data scales grow. These results highlight the potential of the ASN family to serve as a next-generation general-purpose spiking neuron for large-scale SNNs.

## 1 INTRODUCTION

Spiking Neural Networks (SNNs), widely regarded as the third generation of neural networks (Maass, 1997), have attracted increasing attention due to their high biological plausibility and superior energy efficiency. Unlike conventional Artificial Neural Networks (ANNs), which rely on continuous-valued activations, SNNs communicate through discrete spikes and process information via event-driven dynamics. This spike-based computation paradigm enables neurons to remain inactive in the absence of events, resulting in sparse computation. Accordingly, SNNs replace traditional high-power dense multiply–accumulate (MAC) operations with sparse low-power accumulate (AC) operations, thereby achieving substantial energy savings (Roy et al., 2019). SNNs are increasingly being considered a promising paradigm for energy-efficient artificial intelligence (Fang et al., 2023a).

The pivotal role of spiking neurons within SNNs means that, as artificial intelligence (AI) tasks grow in complexity, the design of these neural units must evolve to meet higher benchmarks of efficiency and functional adaptability. The comparison of the previous foundation spiking neurons in the SNN domain is shown in the Table 1. In addition to training efficiency, architectural compatibility, and spike-driven properties, adaptive firing with respect to the membrane potential constitutes another key property. As an essential characteristic of general-purpose spiking neurons, this adaptability regulates firing dynamics to ensure effective information transmission, preventing both excessive and insufficient neuronal activity. The Leaky Integrate-and-Fire (LIF) neuron is hindered by low training efficiency and a lack of membrane potential adaptivity. The Parametric Leaky Integrate-and-Fire (PLIF) neuron (Fang et al., 2021) introduces learnable membrane time constants to capture dynamic neuronal behavior, enabling adaptive firing. However, PLIF still suffers from low training efficiency, which is further exacerbated by the sigmoid-constrained learnable parameters. Parallel Spiking Neuron (PSN) enhances adaptivity via parameter matrices and improves training efficiency; nevertheless, it retains T-dimensional training, leading to non-negligible computational time and memory overheads. Furthermore, the compatibility of PSN with mainstream spiking backbones, such as Spiking Transformers (Zhou et al., 2023b; 2026b; Yao et al., 2023a), remains an open challenge, likely attributable to optimization difficulties associated with its parameter matrices. On a different trajectory, the Integer LIF (ILIF) pioneers an efficient integer Training and Spike



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Inference paradigm. Building on this, NILIF (Lei et al., 2025) incorporates normalization to enhance neuronal training stability. However, both ILIF and NILIF are constrained by insufficient membrane potential adaptivity, limiting their potential as a universal neuron. Although the recent SpikingBrain (Pan et al., 2025) enhances ILIF by employing an adaptive threshold derived from pre-calculated average membrane potentials, this input-dependent strategy remains incongruent with the inherently spike-driven paradigm essential for efficient neuromorphic computation.

| Spiking Neurons | Efficient Training | Adaptive Firing | Architecture Compatibility | Spike-Driven Inference |
|---|---|---|---|---|
| LIF (Gerstner et al., 2014) | ✗ | ✗ | ✓ | ✓ |
| PLIF (Fang et al., 2021) | ✗ | ✓ | ✓ | ✓ |
| PSN (Fang et al., 2023b) | ❍ | ✓ | ✗ | ✓ |
| ILIF (Luo et al., 2024) | ✓ | ✗ | ✓ | ✓ |
| NILIF (Lei et al., 2025) | ✓ | ✗ | ✓ | ✓ |
| **ASN, NASN** | ✓ | ✓ | ✓ | ✓ |

Table 1: Comparison with representative foundational spiking neurons, including LIF Gerstner et al. (2014), PLIF Fang et al. (2021), PSN Fang et al. (2023b), ILIF Luo et al. (2024), and NILIF (Lei et al., 2025). ❍ indicates that PSN trains substantially faster than LIF and PLIF but remains slower than ILIF and NILIF. The ASN family satisfies all four criteria: efficient training, adaptive firing, architectural compatibility, and spike-driven inference.

Based on the above analysis, we adopt a novel functional perspective to systematically revisit the fundamental design principles of the new generation of spiking neurons in SNNs. We argue that the four basic characteristics of spiking neurons—efficient training, adaptive firing, architecture compatibility, and spike-driven inference—must be considered simultaneously. Guided by these principles, we propose the Adaptive Spiking Neuron (ASN) and its variant, the Normalized Adaptive Spiking Neuron (NASN). Both ASN and NASN are general spike-driven neurons that combine efficient training with adaptive membrane potential dynamics. The ASN family (ASN and NASN) introduces a layer-wise learnable parameter $\alpha$ to dynamically shift the activation window in the ILIF family (ILIF and NILIF), aligning with the evolving distribution of membrane potentials across layers. This adaptive mechanism enhances both representational capacity and distributional flexibility, making ASN and NASN particularly suitable for vision and language modeling. Overall, this work advances the design of spiking neurons in SNNs. Our main contributions are summarized as follows.

- We systematically analyze the limitations of existing foundation spiking neurons and provide a functional view for the new generation of spiking neuron design.
- Guided by these principles, we propose the **A**daptive **S**piking **N**euron (**ASN**) and its normalized variant, the Normalized Adaptive Spiking Neuron (NASN). Both are general spike-driven neurons that enable efficient training and adaptive membrane-potential dynamics.
- We evaluate the ASN family on 20 datasets across vision and language tasks, demonstrating its effectiveness and versatility. Notably, NASN exhibits promising scalability in spike-based language modeling, highlighting the potential of the ASN family as a next-generation general-purpose spiking neuron.

## 2 Related Work

Spiking neural networks are commonly constructed through two approaches: ANN-to-SNN conversion (Liu et al., 2025; Cao et al., 2026; Wang et al., 2026) and direct training (Fang et al., 2022; Zhou et al., 2023b; 2026b; 2024a). This work focuses on direct training. Due to space constraints, a detailed review of SNN models for vision and language tasks is deferred to Appendix A.

**Foundation Spiking Neurons.** The LIF (Gerstner et al., 2014) neuron model is the most commonly used neuron in SNNs due to its simplicity. In recent years, several improved LIF-based neurons have been proposed to better adapt to modern artificial intelligence tasks. The PLIF neuron (Fang et al., 2021) introduces learnable membrane time constants to capture dynamic neuronal behavior. PSN (Fang et al., 2023b) further extends LIF by removing the reset mechanism and incorporating a learnable parameter matrix, generating hidden states that are independent across time steps and enabling parallelizable neuronal dynamics. LIF, PLIF, and PSN represent key approaches within the

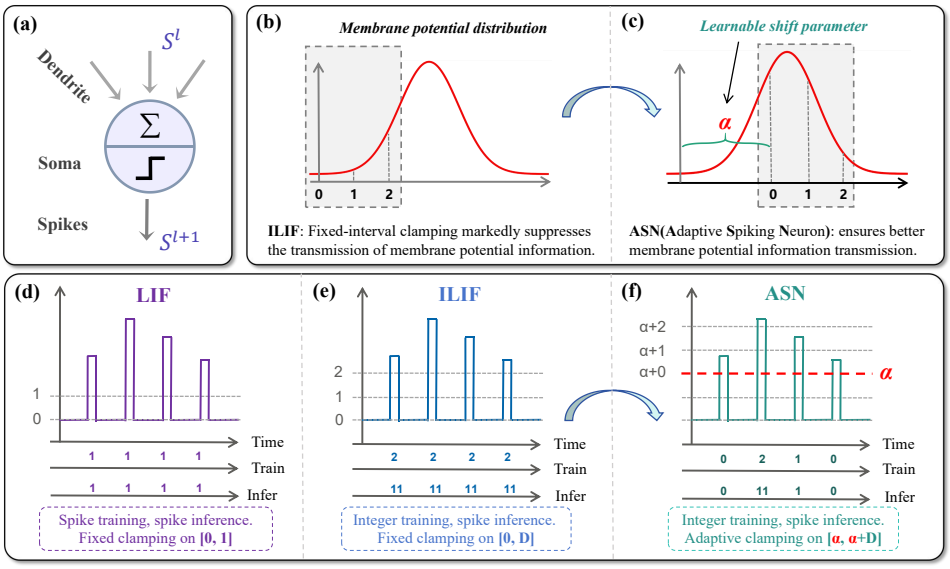


Figure 1: Overview of Adaptive Spiking Neuron(ASN). (a) schematics of a biological spiking neuron. (b) depicts a neuron lacking membrane potential adaptation, thereby suppressing the transmission of membrane potential information. (c) illustrates the proposed neuron, where a learnable shift parameter enables adaptive firing, thereby ensuring better membrane potential information transmission. From the perspective of spike dynamics at the neural level. (d) and (e) show the dynamics of LIF (spike training and spike inference) and ILIF (integer training and spike inference) neurons, respectively. Both employ fixed-interval clamping, resulting in significant membrane potential quantization bias. (f) shows the spike dynamics of our proposed ASN with adaptive firing, achieving more precise spike quantization of membrane potential.

spike training and spike inference paradigm. An alternative training approach is the integer training and spike inference paradigm. ILIF (Luo et al., 2024) implements this paradigm by employing integer-valued activations during training while maintaining spike-driven inference through virtual timesteps. NILIF (Lei et al., 2025) introduces a normalized version of ILIF to improve training stability for SNNs with complex architectures. SpikingBrain (Pan et al., 2025) further adopts an adaptive threshold in ILIF by precomputing the average membrane potential based on the input; however, this input-dependent adaptation is not well-suited to the spike-driven nature of neuromorphic computation. Previous works on neurons primarily validated them in a single modality. In this work, we will simultaneously validate the effectiveness and generalizability of our methods in both visual and language modalities.

# 3 METHOD

## 3.1 A FUNCTIONAL VIEW OF SPIKING NEURONS

We start with a review of the principles of the three most representative neurons: LIF, ILIF, and PSN.

**LIF Model.** The dynamics of LIF can be formulated as follows:

$$U[t] = H[t-1] + X[t], \tag{1}$$

$$S[t] = \Theta\left(U[t] - V_{\text{th}}\right), \tag{2}$$

$$H[t] = \beta(U[t] - S[t]), \tag{3}$$

where $t$ denotes the time step and $X[t]$ is the input current. When the membrane potential $U[t]$ exceeds the firing threshold $V_{\text{th}}$, the neuron emits a spike $S[t]$. $\Theta(v)$ is the Heaviside step function, equal to 1 when $v \geq 0$ and 0 otherwise. After spike generation, the spike value is subtracted from $U[t]$, and the residual membrane potential decays by a factor of $\beta$ to produce $H[t]$. As shown in Figure 1(d), LIF uses spike-based training and inference, with neuronal activity constrained to the

binary range $\{0, 1\}$. Owing to its $T$-dimensional temporal expansion, the LIF neuron Gerstner et al. (2014) incurs substantial training overhead, making it more suitable for small- and medium-scale SNNs. Moreover, LIF lacks adaptive membrane-potential dynamics.

**ILIF Model.** The dynamics of ILIF are formulated as follows:

$$\begin{aligned} U[t] &= H[t-1] + X[t], \\ S[t] &= \mathrm{clip}(\mathrm{round}(U[t]), 0, D), \\ H[t] &= \beta(U[t] - S[t]), \end{aligned} \tag{4}$$

where $\mathrm{clip}(U[t], x_{\min}, x_{\max})$ clips $U[t]$ to $[x_{\min}, x_{\max}]$. $D$ is the maximum quantized integer value; during inference on neuromorphic hardware, each integer activation is unfolded over $D$ binary spike steps. Thus, a configuration with $T$ integer steps produces a spike sequence of length $T \times D$. For example, ILIF ($1 \times 4$) produces four binary spike steps. As shown in Figure 1(e), ILIF and its variant NILIF (Lei et al., 2025) use integer-valued training and spike-based inference Luo et al. (2024), with the membrane potential clamped to $[0, D]$ during training. Although these neurons enable efficient training, their lack of adaptive neuronal dynamics limits their generality.

**PSN Model.** PSN Fang et al. (2023b) removes the reset process in LIF and achieves adaptive dynamics by applying a linear transformation to the $T$-dimensional input membrane potential using a $T \times T$ parameter matrix.

$$\boldsymbol{H} = \boldsymbol{W}\boldsymbol{X}, \qquad \boldsymbol{W} \in \mathbb{R}^{T\times T}, \boldsymbol{X} \in \mathbb{R}^{T\times N}, \tag{5}$$

$$\boldsymbol{S} = \Theta(\boldsymbol{H} - \boldsymbol{B}), \qquad \boldsymbol{B} \in \mathbb{R}^{T}, \boldsymbol{S} \in \{0, 1\}^{T\times N}, \tag{6}$$

where $\boldsymbol{X}$ is the input sequence, $\boldsymbol{W}$ is the learnable weight matrix, $\boldsymbol{H}$ is the hidden-state sequence, $\boldsymbol{B}$ is the learnable threshold, and $\boldsymbol{S}$ is the binary output spike sequence. PSN retains a $T$-dimensional training representation, yielding training efficiency between that of LIF and ILIF. More critically, its parameter matrix can be difficult to optimize, which complicates integration with spiking transformers and limits general applicability.

Based on the above analysis, we present a novel functional perspective on the new generation of spiking neuron design to better accommodate vision and language modeling. We argue that a next-generation foundational spiking neuron should jointly satisfy the following four criteria:

1) **Efficient Training**: The neuron should enable efficient and scalable SNN training, making it suitable for large-scale models and contemporary learning paradigms.

2) **Adaptive Firing**: The neuron should adapt its firing behavior to membrane-potential dynamics, thereby avoiding both excessive firing activity and insufficient spike transmission.

3) **Architectural Compatibility**: The neuron should generalize effectively across domains, such as vision and language, while remaining compatible with advanced architectures, including spiking transformers.

4) **Spike-Driven Inference**: During inference, the neuron should preserve a spike-driven computational paradigm to fully exploit the energy-efficiency advantages of SNNs.

Table 1 compares representative spiking neurons. The ASN family, including ASN and NASN, satisfies all four criteria.

### 3.2 Adaptive Spiking Neuron

We propose the Adaptive Spiking Neuron (ASN), a general spike-driven neuron model that simultaneously achieves efficient training and adaptive membrane potential dynamics. By adopting the integer-training and spike-inference paradigm of ILIF, ASN emits integer values during training and unfolds them into binary spikes during inference, enabling highly efficient training. ASN also introduces a layer-wise trainable parameter $\alpha$ that adaptively shifts the activation window to better align with layer-specific membrane-potential distributions. As shown in Figure 1(c) and (f), this learnable offset enables adaptive firing. ASN is formulated as follows:

$$\begin{aligned} U[t] &= H[t-1] + X[t], \\ S[t] &= \mathrm{clip}(\mathrm{round}(U[t]), \alpha, \alpha + D), \\ H[t] &= \beta(U[t] - S[t]), \end{aligned} \tag{7}$$

| Methods | Architecture | Param (M) | Train Size | Test Size | T | E (mJ) | Acc (%) |
|---|---|---|---|---|---|---|---|
| Spikformer Zhou et al. (2023b) | Spikformer-8-384 | 16.81 | $224^2$ | $224^2$ | 4 | 12.43 | 70.24 |
| SD-Transformer Yao et al. (2023a) | SD-Transformer-8-384 | 16.81 | $224^2$ | $224^2$ | 4 | 3.90 | 72.28 |
| LIF (Spikingformer) Zhou et al. (2026b) | Spikingformer-8-384 | 16.81 | $224^2$ | $224^2$ | 4 | 4.69 | 72.45 |
| LIF (Spikingformer$^\dagger$) Zhou et al. (2026b) | Spikingformer-8-384 | 16.81 | $224^2$ | $224^2$ | 4 | 5.61 | 74.35 |
| NILIF (Spikingformer$^\dagger$) | Spikingformer-8-384 | 16.81 | $224^2$ | $224^2$ | 4 | 5.58 | 75.41 |
| **Ours** | Spikingformer-8-384 | 16.81 | $224^2$ | $224^2$ | 4 | 5.67 | **75.53** |

Table 2: ImageNet-1K classification results. "T" denotes the number of time steps, and "E (mJ)" denotes energy consumption in millijoules, as detailed in Appendix D. Spikingformer$^\dagger$ is the variant with improved downsampling Zhou et al. (2023a).

where $\mathrm{round}(\cdot)$ rounds to the nearest integer, $\mathrm{clip}(X, x_{\min}, x_{\max})$ clips $X$ to $[x_{\min}, x_{\max}]$, and $\alpha$ is the learnable shift parameter. Because the rounding and clipping operations in Eq. 7 are non-differentiable, Section 3.4 describes the surrogate gradients for the input $x$ and parameter $\alpha$.

The essential difference from ILIF is that ASN learns the location of the activation window while retaining its width $D$. Thus, $\alpha$ reallocates the same finite representation range to suit each layer without increasing the number of discrete levels or virtual inference steps. Because $\alpha$ is shared at the layer level, this adaptation introduces negligible parameter overhead and avoids input-dependent threshold computation during inference.

The synaptic computation within a layer is illustrated as

$$\mathbf{X}^{l+1} = \mathbf{W}^l S^l[t] = \mathbf{W}^l \,\mathrm{clip}(\mathrm{round}(\mathbf{U}^l), \alpha, \alpha + D), \tag{8}$$

where $\mathbf{U}^l$ is the membrane potential before quantization at layer $l$, and $\mathbf{W}^l$ denotes the synaptic weights of that layer. Similar to ILIF, ASN converts integer activations into binary spikes during inference. The linear computation can be decomposed into spike-based accumulation and a constant term:

$$\mathbf{X}^{l+1} = \mathbf{W}^l S_1^l[t] + \mathrm{C}\,. \tag{9}$$

We decompose the quantized output as $S^l[t] = S_1^l[t] + \lceil \alpha \rceil$ and unfold $S_1^l[t]$ into the binary spike sequence $\left\{S^l[t,d]\right\}_{d=1}^{D}$, which satisfies

$$\sum_{d=1}^{D} S^l[t,d] = S_1^l[t]. \tag{10}$$

The constant term can be formulated as

$$\mathbf{W}^l \lceil \alpha \rceil = \mathrm{C}, \tag{11}$$

where $\lceil \cdot \rceil$ denotes the ceiling operation. Relative to ILIF inference, ASN requires only the additional layer-wise constant $\mathrm{C} = \mathbf{W}^l \lceil \alpha \rceil$.

### 3.3 Normalized Adaptive Spiking Neuron

Similar to NILIF (Lei et al., 2025), ASN may exhibit gradient instability when trained with integer activations. We therefore introduce a normalization factor $N$ to stabilize activation magnitudes and gradients. The resulting variant, termed the Normalized Adaptive Spiking Neuron (NASN), is defined as follows:

$$\begin{aligned} U[t] &= H[t-1] + X[t], \\ S[t] &= \mathrm{clip}(\mathrm{round}(U[t]), \alpha, \alpha + D)/N, \qquad (12) \\ H[t] &= \beta\,(U[t] - NS[t])\,. \qquad (13) \end{aligned}$$

For simplicity, we usually set $N = D$. The synaptic computation within a layer is illustrated as follows:

$$\mathbf{X}^{l+1} = \mathbf{W}^l S^l[t] = \frac{1}{N}\mathbf{W}^l \,\mathrm{clip}(\mathrm{round}(\mathbf{U}^l), \alpha, \alpha + D). \tag{14}$$

As with ASN, NASN inference decomposes the linear computation into spike-based accumulation and a constant term:

$$\mathbf{X}^{l+1} = \mathbf{W}_1^l S_1^l[t] + \mathrm{C}, \tag{15}$$

| Model | T | MNLI | QQP | QNLI | SST-2 | CoLA | STS-B | MRPC | RTE | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| $\text{BERT}_{\text{base}}$ (Devlin et al., 2019) | – | 83.4 | 71.2 | 90.5 | 93.5 | 52.1 | 85.8 | 88.9 | 66.4 | 79.6 |
| Q2BERT (Zhang et al., 2020) | – | 47.3 | 67.0 | 61.3 | 80.6 | 0.0 | 4.7 | 81.2 | 52.7 | 49.1 |
| SpikeLM (Xing et al., 2024b) | 4 | 77.2 | 83.9 | 85.3 | 87.0 | 38.8 | 84.9 | 85.7 | 69.0 | 76.5 |
| LIF-BERT (Xing et al., 2024b) | 4 | 35.2 | 0 | 50.5 | 50.9 | 0 | 0 | 81.2 | 52.7 | 34.6 |
| PSN-BERT (Xing et al., 2024b) | 4 | 35.2 | 0 | 50.5 | 50.9 | 0 | 6.8 | 81.2 | 52.7 | 34.7 |
| SpikeBERT (Lv et al., 2023) | 4 | 71.0 | 68.2 | 66.4 | 85.4 | 16.9 | 18.7 | 82.0 | 57.5 | 59.7 |
| NILIF (WE-SpikingFormer) | 4 | 70.1 | 85.1 | 77.5 | 89.0 | 27.9 | 42.8 | 81.6 | 55.9 | 66.3 |
| **NASN** (WE-SpikingFormer) | 4 | 75.5 | 86.1 | 82.9 | 87.0 | 33.5 | 45.0 | 79.1 | 50.9 | **67.5** |

Table 3: Results on the General Language Understanding Evaluation (GLUE) benchmark. "Avg." denotes the mean GLUE score, and "T" denotes the number of time steps. The LIF-BERT and PSN-BERT results are reported by Xing et al. (2024b).

where $S_1^l[t]$ follows Eq. 10 and the effective weight is scaled as

$$\mathbf{W}_1^l = \frac{1}{N}\mathbf{W}^l. \tag{16}$$

The constant term can be formulated as follows:

$$\mathrm{C} = \mathbf{W}_1^l\lceil\alpha\rceil. \tag{17}$$

Compared with NILIF inference, NASN requires only an additional constant C at each layer.

Unlike conventional spiking neurons that employ fixed-interval clamping, our ASN family (including ASN and NASN) incorporates a learnable offset parameter $\alpha$, enabling dynamic shifts of the quantization window to better align with the evolving distribution of membrane potentials across layers. This mechanism enhances both the representational capacity and distributional adaptability of SNNs.

### 3.4 Gradient Optimization

**Gradient Estimation via STE.** Since the rounding and clipping operations are non-differentiable (possessing zero derivatives almost everywhere), we employ the Straight-Through Estimator (STE) to facilitate end-to-end backpropagation.

**Gradient of the Input $x$.** The gradient is allowed to flow only within the active quantization range. We denote the lower bound by $d_{\min} = \alpha$ and the upper bound by $d_{\max} = \alpha + D$. The surrogate gradient for $x$ is defined as

$$\frac{\partial y}{\partial x} = \begin{cases} 1, & \text{if } d_{\min} \le x \le d_{\max}, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

**Gradient of the Learnable Parameter $\alpha$.** The optimization of $\alpha$ is primarily driven by samples falling into the truncated region (outside the boundaries). When $x$ resides at or beyond the boundaries, the output $y$ becomes a direct function of the boundary values, yielding the following approximation:

$$\frac{\partial y}{\partial \alpha} = \begin{cases} 1, & \text{if } x < d_{\min} \text{ or } x > d_{\max}, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

Consequently, the total gradient of the objective loss $\mathcal{L}$ with respect to $\alpha$ is accumulated over the mini-batch as:

$$\nabla_\alpha \mathcal{L} = a \sum_i \frac{\partial \mathcal{L}}{\partial y_i}\, \mathbb{I}(x_i \notin [d_{\min}, d_{\max}]), \tag{20}$$

where $\mathbb{I}(\cdot)$ is the indicator function, and $a$ is a hyperparameter for adjusting gradient scaling. This gradient mechanism allows the network to adaptively shift its activation window to minimize task-specific loss, providing a flexible balance between quantization error and dynamic range preservation.

NASN $(1 \times 4)$ is used as the default configuration in our main experiments and corresponds to four inference time steps.

# 4 EXPERIMENTS

To evaluate the effectiveness and generalization of the proposed adaptive neurons, we conduct experiments on 20 datasets covering vision and language modeling. Detailed CIFAR results are provided in Appendix I.

**ImageNet-1k Classification.** We conduct experiments on ImageNet-1K, which contains approximately 1.28 million training images and 50,000 validation images across 1,000 object classes. We benchmark our model against several state-of-the-art SNN foundation architectures. Among these baselines, Spikingformer is a strong backbone that combines spike-driven operations with global modeling. For a controlled comparison, we adopt the Spikingformer-8-384 architecture and replace its standard spiking neurons with NASN. As summarized in Table 2, our model achieves 75.53% top-1 accuracy, exceeding the LIF-based Spikingformer$^{\dagger}$ baseline by 1.18 percentage points and the NILIF-based variant by 0.12 percentage points under the same architecture. These results underscore the efficacy of the learnable offset $\alpha$ in optimizing the firing threshold to better capture the complex data distributions of large-scale visual tasks.

| Model | E (mJ) | Time Step | ARC-e | ARC-c | BoolQ | HeadQA | OBQA | Avg. |
|---|---|---|---|---|---|---|---|---|
| SpikeLLM-7B (Xing et al., 2024a) | - | 4 | 31.3 | 23.6 | 53.8 | - | - | - |
| Qwen-1.5B (Yang et al., 2024) | 3398.3 | - | 26.2 | 26.9 | 60.3 | 25.7 | 27.1 | 33.2 |
| NILIF (WD-SpikingFormer-0.4B) | 238.4 | 4 | 30.0 | 22.3 | 37.8 | 26.0 | 26.1 | 28.4 |
| NILIF (WD-SpikingFormer-1.0B$^*$) | 653.2 | 4 | 41.9 | 20.4 | 42.1 | 26.7 | 28.2 | 31.9 |
| **NASN** (WD-SpikingFormer-0.4B) | 245.1 | 4 | 35.3 | 20.6 | 37.9 | 25.9 | 27.2 | 29.4 |
| **NASN** (WD-SpikingFormer-1.0B$^*$) | 684.5 | 4 | 44.8 | 21.1 | 57.3 | 27.2 | 27.0 | **35.5** |

Table 4: Results on question-answering tasks (QAT). "E (mJ)" denotes energy consumption in millijoules, and "Avg." denotes average accuracy (%). WD-SpikingFormer-1.0B$^*$ denotes the model pretrained on 0.5B tokens.

| Model | E (mJ) | T | HellaSwag | PIQA | WinoGrande | Avg. |
|---|---|---|---|---|---|---|
| SpikeLLM-7B (Xing et al., 2024a) | - | 4 | 33.9 | 53.4 | 51.5 | 46.3 |
| Qwen-1.5B (Yang et al., 2024) | 3398.3 | - | 26.5 | 53.7 | 52.8 | 44.3 |
| NILIF (WD-SpikingFormer-0.4B) (Zhou et al., 2026a) | 238.4 | 4 | 25.9 | 53.4 | 50.2 | 43.2 |
| NILIF (WD-SpikingFormer-1.0B$^*$) (Zhou et al., 2026a) | 653.2 | 4 | 26.4 | 55.7 | 50.3 | 44.1 |
| **NASN** (WD-SpikingFormer-0.4B) | 245.1 | 4 | 26.0 | 54.6 | 49.4 | 43.3 |
| **NASN** (WD-SpikingFormer-1.0B$^*$) | 684.5 | 4 | 26.4 | 58.0 | 51.0 | **45.1** |

Table 5: Results on commonsense-reasoning tasks (CRT). "E (mJ)" denotes energy consumption in millijoules, and "Avg." denotes average accuracy (%). WD-SpikingFormer-1.0B$^*$ denotes the model pretrained on 0.5B tokens.

**Natural Language Understanding.** We evaluate our method on the GLUE benchmark Wang et al. (2018), a widely used suite of natural language understanding tasks designed to assess the general linguistic capabilities of machine learning models across diverse domains. We pretrain the model on the English Wikipedia corpus Devlin et al. (2019) using masked language modeling (MLM). Pretraining uses eight GPUs and follows the architecture and experimental settings of WE-SpikingFormer (Zhou et al., 2026a), which uses NILIF neurons. We then fine-tune the pretrained model on GLUE. For a comprehensive comparison, we include Artificial Neural Network (ANN) baselines, specifically the original BERT Devlin et al. (2019) and the quantized Q2BERT Zhang et al. (2020), the latter of which utilizes 2-bit weights and 8-bit activations.

The results are summarized in Table 3. All evaluated models contain 12 encoder blocks and have a comparable scale of approximately 0.1B parameters. Our method achieves a mean GLUE score of 67.5, outperforming the NILIF version of WE-SpikingFormer by 1.2 points. In our comparative analysis, we note that while SpikeLM achieves higher performance, it functions as a softmax-based spiking transformer and retains non-spiking GeLU Hendrycks & Gimpel (2016) activations within its MLP blocks. In contrast, our method operates as a softmax-free, spike-driven transformer. Within the category of spike-driven architectures, our method significantly outperforms existing methods such as LIF-BERT (34.6%), PSN-BERT, and SpikeBERT (59.7%), achieving a substantial performance gain with an accuracy of 67.5%. These results demonstrate that the learnable offset $\alpha$ effectively bridges the representational gap in complex language understanding tasks.

**Question Answering Tasks.** We evaluate our model on several question-answering tasks (QAT) covering scientific reasoning, Boolean inference, and domain-specific knowledge. Following WD-SpikingFormer (Zhou et al., 2026a), we pretrain on the FineWeb-Edu corpus (Lozhkov et al., 2024) using 0.1B tokens by default. Results are reported in Table 4. Overall, the proposed NASN consistently improves the performance of WD-SpikingFormer across different model scales and pretraining settings. For the 0.4B model, NASN improves the average accuracy from 28.4% to 29.4% compared with NILIF, while introducing only a small energy increase from 238.4 mJ to 245.1 mJ. The improvement becomes larger when the 1.0B-parameter model is pretrained on more tokens. Compared with NILIF on WD-SpikingFormer-1.0B$^*$, NASN improves average accuracy from 31.9% to 35.5%, a gain of 3.6 percentage points. The improvement is especially notable on ARC-e and BoolQ, where NASN achieves 44.8% and 57.3%, respectively, exceeding the corresponding NILIF baseline by 2.9 and 15.2 percentage points. These results suggest that NASN benefits from additional pretraining data and provides stronger representations for question-answering tasks. Compared with the dense Qwen-1.5B model, NASN with WD-SpikingFormer-1.0B$^*$ achieves a higher average accuracy, 35.5% vs. 33.2%, while consuming much less energy, 684.5 mJ vs. 3398.3 mJ. This corresponds to about 80% lower energy consumption. These results demonstrate that the proposed NASN achieves a favorable trade-off between accuracy and energy efficiency, showing its effectiveness for building energy-efficient spiking language models.

| Model | PT data. | $\alpha_{\text{avg}}$ | $\lvert\alpha\rvert_{\text{avg}}$ | $\min \lvert\alpha\rvert$ | $\max \lvert\alpha\rvert$ | CRT Avg. (%) | QAT Avg. (%) |
|---|---|---|---|---|---|---|---|
| NASN (WD-SpikingFormer-0.4B) | 0.1B | -0.05 | 0.06 | 8.46e-13 | 0.27 | 43.3 | 29.4 |
| NASN (WD-SpikingFormer-1.0B) | 0.1B | -0.07 | 0.08 | 5.81e-08 | 0.57 | 43.5 (+0.2) | 29.8 (+0.4) |
| NASN (WD-SpikingFormer-1.0B) | 0.5B | -0.20 | 0.68 | 8.44e-07 | 3.10 | 45.1 (+1.8) | 35.5 (+6.1) |
| NASN (WD-SpikingFormer-1.0B) | 2.5B | -0.81 | 1.77 | 7.25e-06 | 8.99 | 47.8 (+4.5) | 37.2 (+7.8) |

Table 6: Statistics of the learnable shift parameter $\alpha$ during training. "PT data" denotes the number of pretraining tokens.

**Commonsense Reasoning Tasks.** We evaluate our methods on several Commonsense Reasoning Tasks (CRT) that assess a model's ability to perform grounded inference, physical reasoning, and context-aware understanding. Our experimental configuration follows WD-SpikingFormer (Zhou et al., 2026a). Table 5 reports the commonsense-reasoning results. For WD-SpikingFormer-0.4B, NASN improves average accuracy from 43.2% to 43.3%, while estimated energy consumption increases only from 238.4 mJ to 245.1 mJ. The advantage of NASN becomes more evident for larger models pretrained on more tokens. On WD-SpikingFormer-1.0B$^*$, NASN improves average accuracy from 44.1% to 45.1% over NILIF, a gain of 1.0 percentage point. The improvement mainly comes from PIQA and WinoGrande, where NASN achieves 58.0% and 51.0%, exceeding NILIF by 2.3 and 0.7 percentage points, respectively. This result indicates that NASN can better exploit additional pretraining data for commonsense reasoning. Compared with the dense Qwen-1.5B model, NASN with WD-SpikingFormer-1.0B$^*$ achieves higher average accuracy, 45.1% vs. 44.3%, while consuming much less energy, 684.5 mJ vs. 3398.3 mJ. This corresponds to nearly 80% energy reduction. Although SpikeLLM-7B obtains a higher average accuracy of 46.3%, it uses a much larger 7B model scale. Overall, these results demonstrate that NASN provides a favorable trade-off between commonsense reasoning performance and energy efficiency, especially under larger pretraining settings.

| Neuron model | CIFAR-100 (%) |
|---|---|
| LIF (Zhou et al., 2026b) | 80.22 |
| PLIF (Fang et al., 2021) | 80.25 |
| PSN (Fang et al., 2023b) | 77.84 |
| ILIF (Luo et al., 2024) | 80.32 |
| **ASN (Ours)** | 80.36 |
| NILIF (Lei et al., 2025) | 80.98 |
| **NASN (Ours)** | 81.11 |

Table 7: Ablation study of spiking neurons on CIFAR-100.

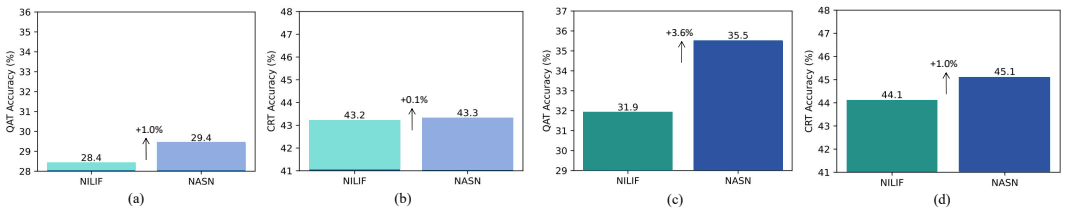


Figure 2: Scaling comparison between NILIF and NASN in WD-SpikingFormer. Panels (a) and (b) report QAT and CRT results for the 0.4B-parameter model pretrained on 0.1B tokens; panels (c) and (d) report the corresponding results for the 1.0B-parameter model pretrained on 0.5B tokens.

## 5 Discussion

**Analysis of the learnable shift parameter $\alpha$.** To better understand the role of the learnable shift parameter $\alpha$, we analyze its behavior under different model sizes and pretraining-data scales. As shown in Table 6, larger changes in the magnitude of $\alpha$ are associated with larger performance gains from NASN. Overall, $\alpha$ shifts in the negative direction, indicating that the fixed $[0, 4]$ activation range used by NILIF is too high for many layers; an adaptive leftward shift helps preserve negative activations. When the model and pretraining corpus are relatively small, the variation in $\alpha$ remains limited. In this case, NASN behaves similarly to NILIF, which helps explain the small performance gap at limited scale. As the model size or pretraining-data scale increases, the variation in $\alpha$ becomes larger. This indicates that the learnable shift increasingly adapts to changes in membrane-potential distributions, thereby improving neuronal representation. The growing advantage of NASN over NILIF suggests that this adaptive mechanism is particularly useful for larger, more extensively pretrained models. These observations support our claim that $\alpha$ is not merely an additional learnable parameter, but plays an increasingly important role in adapting the neuronal dynamics as the model scales up.

**Ablation Study.** We compare the ILIF family (ILIF and NILIF) with the ASN family (ASN and NASN) on CIFAR-100 using Spikingformer$^{\dagger}$ (Zhou et al., 2026b). Table 7 shows that NASN performs best; ASN outperforms ILIF, and NASN outperforms NILIF, supporting adaptive activation-window shifts.

**Neuron performance comparison on scaling.** Based on Tables 4 and 5, Figure 2 visualizes the performance difference between NILIF and NASN as model and data scales increase. The advantage of NASN becomes more pronounced at larger scales, indicating stronger scaling behavior than NILIF in these experiments.

The present study also has several limitations. Energy is estimated from operation counts and standard coefficients; measurements on neuromorphic hardware would better characterize latency, memory traffic, and system-level energy. Moreover, the layer-wise $\alpha$ does not capture channel-wise or token-wise variation, and broader pretraining studies are needed to establish scaling behavior beyond the ranges evaluated here.

## 6 Conclusion

We present a unified perspective for designing next-generation spiking neurons, jointly considering training efficiency, architectural compatibility, spike-driven inference, and adaptive firing. Building on the ILIF family, we introduce learnable membrane-potential dynamics. The resulting Adaptive Spiking Neuron (ASN) enables adaptive firing while preserving efficient integer training and spike-driven inference. Its normalized variant, NASN, improves training stability. Experiments on 20 vision and language datasets demonstrate the effectiveness, versatility, and scalability of the ASN family, particularly in spike-based language modeling. These results establish the ASN family as a practical general-purpose neuron for scalable, energy-efficient SNNs.

## AI USE STATEMENT

AI tools were used solely for language polishing. The methods and experiments were developed and conducted by the authors without AI assistance. The authors take full responsibility for the content of this paper.

## REFERENCES

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Honglin Cao, Shuai Wang, Zijian Zhou, Ammar Belatreche, Wenjie Wei, Yu Liang, Yu Yang, Rui Xi, Malu Zhang, and Haizhou Li. Towards training-free and accurate ann-to-snn conversion via activation-aware redistribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1703–1711, 2026.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

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 (ICCV)*, pp. 2661–2671, 2021.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timoth Masquelier, and Yonghong Tian. Deep Residual Learning in Spiking Neural Networks. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 21056–21069, 2022.

Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, Liwei Huang, Huihui Zhou, Guoqi Li, and Yonghong Tian. Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40):eadi1480, 2023a. doi: 10.1126/sciadv.adi1480. URL https://www.science.org/doi/abs/10.1126/sciadv.adi1480.

Wei Fang, Zhaofei Yu, Zhaokun Zhou, Ding Chen, Yanqi Chen, Zhengyu Ma, Timothée Masquelier, and Yonghong Tian. Parallel spiking neurons with high efficiency and ability to learn long-term dependencies. *Advances in Neural Information Processing Systems*, 36:53674–53687, 2023b.

Yuetong Fang, Deming Zhou, Ziqing Wang, Hongwei Ren, ZeCui Zeng, Lusong Li, Renjing Xu, et al. Spiking neural networks need high-frequency information. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 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.

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

Souvik Kundu, Massoud Pedram, and Peter A Beerel. Hire-snn: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5209–5218, 2021.

Zhenxin Lei, Man Yao, Jiakui Hu, Xinhao Luo, Yanye Lu, Bo Xu, and Guoqi Li. Spike2former: Efficient spiking transformer for high-performance image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1364–1372, 2025.

Chang Liu, Jiangrong Shen, Xuming Ran, Mingkun Xu, Qi Xu, Yi Xu, and Gang Pan. Efficient ann-snn conversion with error compensation learning. *arXiv preprint arXiv:2506.01968*, 2025.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. Fineweb-edu: the finest collection of educational content, 2024. URL `https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu`.

Xinhao Luo, Man Yao, Yuhong Chou, Bo Xu, and Guoqi Li. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In *European Conference on Computer Vision*, pp. 253–272. Springer, 2024.

Changze Lv, Tianlong Li, Jianhan Xu, Chenxi Gu, Zixuan Ling, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. Spikebert: A language spikformer learned from bert with knowledge distillation. *arXiv preprint arXiv:2308.15122*, 2023.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Yuqi Pan, Yupeng Feng, Jinghao Zhuang, Siyu Ding, Han Xu, Zehao Liu, Bohan Sun, Yuhong Chou, Xuerui Qiu, Anlin Deng, et al. Spikingbrain: Spiking brain-inspired large models. *arXiv preprint arXiv:2509.05276*, 2025.

Priyadarshini Panda, Sai Aparna Aketi, and Kaushik Roy. Toward scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization. *Frontiers in Neuroscience*, 14:653, 2020.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

David Vilares and Carlos Gómez-Rodríguez. Head-qa: A healthcare dataset for complex reasoning. *arXiv preprint arXiv:1906.04701*, 2019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Hongzhi Wang, Xiubo Liang, Tao Zhang, Yue Gu, and Weidong Geng. Pssd-transformer: powerful sparse spike-driven transformer for image semantic segmentation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 758–767, 2024.

Jingya Wang, Xin Deng, Wenjie Wei, Dehao Zhang, Shuai Wang, Qian Sun, Jieyuan Zhang, Hanwen Liu, Ning Xie, and Malu Zhang. Training-free ann-to-snn conversion for high-performance spiking transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2128–2136, 2026.

Xingrun Xing, Boyan Gao, Zheng Zhang, David A Clifton, Shitao Xiao, Li Du, Guoqi Li, and Jiajun Zhang. Spikellm: Scaling up spiking neural network to large language models via saliency-based spiking. *arXiv preprint arXiv:2407.04752*, 2024a.

Xingrun Xing, Zheng Zhang, Ziyi Ni, Shitao Xiao, Yiming Ju, Siqi Fan, Yequan Wang, Jiajun Zhang, and Guoqi Li. Spikelm: Towards general spike-driven language modeling via elastic bi-spiking mechanisms. In *International Conference on Machine Learning*, pp. 54698–54714. PMLR, 2024b.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *Advances in neural information processing systems*, 36:64043–64058, 2023a.

Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023b.

Man Yao, JiaKui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. *arXiv preprint arXiv:2404.03663*, 2024.

Man Yao, Xuerui Qiu, Tianxiang Hu, Jiakui Hu, Yuhong Chou, Keyu Tian, Jianxing Liao, Luziwei Leng, Bo Xu, and Guoqi Li. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. Ternarybert: Distillation-aware ultra-low bit bert. *arXiv preprint arXiv:2009.12812*, 2020.

Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Zhengyu Ma, Huihui Zhou, Xiaopeng Fan, and Yonghong Tian. Enhancing the performance of transformer-based spiking neural networks by snn-optimized downsampling with precise gradient backpropagation. *arXiv preprint arXiv:2305.05954*, 2023a.

Chenlin Zhou, Han Zhang, Liutao Yu, Yumin Ye, Zhaokun Zhou, Liwei Huang, Zhengyu Ma, Xiaopeng Fan, Huihui Zhou, and Yonghong Tian. Direct training high-performance deep spiking neural networks: a review of theories and methods. *Frontiers in Neuroscience*, 18:1383844, 2024a.

Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, Li Yuan, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. Qkformer: Hierarchical spiking transformer using qk attention. *Advances in Neural Information Processing Systems*, 37:13074–13098, 2024b.

Chenlin Zhou, Sihang Guo, Jiaqi Wang, Dongyang Ma, Kaiwei Che, Baiyu Chen, Qingyan Meng, Zhengyu Ma, and Yonghong Tian. Winner-take-all spiking transformer for language modeling. *arXiv preprint arXiv:2604.11321*, 2026a.

Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Han Zhang, Jiaqi Wang, Huihui Zhou, Zhengyu Ma, and Yonghong Tian. Spikingformer: A key foundation model for spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2236–2244, 2026b.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng YAN, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=frE4fUwz_h.

Rui-Jie Zhu, Qihang Zhao, Guoqi Li, and Jason K Eshraghian. Spikegpt: Generative pre-trained language model with spiking neural networks. *arXiv preprint arXiv:2302.13939*, 2023.

# APPENDIX

**Limitation.** Our energy consumption analysis is based on theoretical estimates rather than measurements on real neuromorphic hardware. Although our method has been extended to the 1B-parameter scale through direct training and shows increasing advantages over the ILIF family, a more systematic study is still needed to understand its scaling behavior and potential emergent properties at larger scales. We leave these directions for future work.

## A RELATED WORK SUPPLEMENT

**Spiking Neural Networks in Vision Tasks.** Spiking Neural Networks (SNNs) have gained increasing traction in visual tasks (Zhou et al., 2023b; 2026b; Yao et al., 2023a; 2024; Zhou et al., 2024b; Yao et al., 2025; Wang et al., 2024). A notable advancement is Spikformer (Zhou et al., 2023b), which introduces Spiking Self-Attention (SSA) by replacing softmax with sparse spike-form Query, Key, and Value, achieving 74.81% accuracy on ImageNet-1k with only four time steps—highlighting the potential of transformer-based SNNs. Building on this, Spikingformer (Zhou et al., 2026b) incorporates a pre-activation shortcut to eliminate floating-point multiplications and reduce firing rates, boosting accuracy to 77.64%. As a backbone architecture, it effectively unifies spike-driven computation with global modeling capabilities. Spike-Driven Transformer (Yao et al., 2023a) proposed Spike-Driven Self-Attention (SDSA), which relies solely on masking and addition, achieving 77.07% on ImageNet-1k with substantially lower computational cost. More recently, hierarchical visual spiking transformers (Yao et al., 2024; Zhou et al., 2024b; Yao et al., 2025; Fang et al., 2025) have surpassed 80% accuracy on ImageNet while maintaining high energy efficiency.

**Spiking Neural Networks in Language Tasks.** SpikeBERT (Lv et al., 2023) extends Spikformer (Zhou et al., 2023b) to language modeling through a softmax-free spiking transformer and a two-stage knowledge-distillation strategy. It first distills knowledge from BERT on large-scale unlabeled corpora and then performs task-specific distillation using a BERT model trained on the same downstream data. SpikeGPT (Zhu et al., 2023) builds a spiking language model upon the RWKV architecture (Peng et al., 2023). SpikeLM (Xing et al., 2024b) introduces a spike-based formulation with bidirectional ternary firing, yielding competitive performance in language modeling. SpikeLLM (Xing et al., 2024a) incorporates generalized integrate-and-fire (GIF) neurons within an optimal brain spiking framework for transformer architectures. However, both SpikeLM and SpikeLLM retain non-spiking nonlinear components, such as softmax-based attention or non-spiking activations in their MLP layers. Zhou et al. (2026a) incorporates the Winner-Take-All (WTA) biological mechanism into spiking transformers and proposes the WTA-based Encoder-only Spiking Transformer (WE-SpikingFormer) for masked language modeling, as well as the WTA-based Decoder-only Spiking Transformer (WD-SpikingFormer) for causal language modeling, systematically exploring softmax-free, spike-driven transformers trained directly for language tasks.

## B DATASET DETAILS

**ImageNet-1K** is a benchmark subset of the full ImageNet dataset, widely used for image classification tasks. It contains approximately 1.28 million training images, 50,000 validation images, and 100,000 test images, spanning 1,000 object categories ranging from animals and plants to vehicles and everyday objects. Each image is labeled with a single class. First featured in the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) starting in 2010, it became the de facto standard for evaluating deep learning models. Breakthroughs such as AlexNet, VGG, ResNet, and Vision Transformers were all benchmarked on ImageNet-1K, making it foundational to modern computer vision research. The input size of our model on ImageNet is set to the default $224 \times 224$.

**CIFAR-10/100.** CIFAR-10 consists of 60,000 $32 \times 32$ color images across 10 classes, with 6,000 images per class. CIFAR-100 has the same image format but contains 100 classes with 600 images each. Both datasets are widely used to benchmark image classification on small, low-resolution inputs.

**General Language Understanding Evaluation (GLUE).** GLUE benchmark (Wang et al., 2018) is a widely adopted framework for assessing natural language understanding (NLU) models across a variety of tasks, including single-sentence classification, sentence-pair classification, and linguistic

acceptability. In our experiments, we focus on eight representative datasets within GLUE: MNLI (Multi-Genre Natural Language Inference): Predict whether a hypothesis is entailed, contradicted, or neutral with respect to a given premise across multiple text genres. QQP (Quora Question Pairs): Detect whether two questions from Quora convey the same meaning. QNLI (Question Natural Language Inference): Reformulated from QA, determines whether a sentence contains the answer to a question. SST-2 (Stanford Sentiment Treebank): Perform binary sentiment classification on movie reviews. CoLA (Corpus of Linguistic Acceptability): Judge whether a sentence is grammatically acceptable. STS-B (Semantic Textual Similarity Benchmark): Measure sentence-level semantic similarity on a scale from 0 to 5. MRPC (Microsoft Research Paraphrase Corpus): Identify whether two sentences are paraphrases. RTE (Recognizing Textual Entailment): Decide whether a hypothesis can be inferred from a premise, based on multiple entailment datasets.

**Question Answering Tasks (QAT).** ARC-e (Clark et al., 2018) and ARC-c (AI2 Reasoning Challenge, easy and challenge subsets) assess scientific knowledge and reasoning skills, with ARC-e focusing on simpler multiple-choice science questions and ARC-c including more complex ones that require advanced reasoning. BoolQ (Boolean Questions) (Clark et al., 2019) is a yes/no question-answering dataset derived from natural queries, requiring models to determine the truthfulness of a statement given a supporting passage. HeadQA (Head Question Answering) (Vilares & Gómez-Rodríguez, 2019) is a multilingual medical question-answering benchmark composed of exams for healthcare professionals, testing domain-specific knowledge. OBQA (OpenBookQA) (Mihaylov et al., 2018) evaluates a model's ability to answer elementary science questions by combining provided core facts with external common knowledge.

**Commonsense Reasoning Tasks (CRT).** HellaSwag (Zellers et al., 2019) is a large-scale dataset for grounded commonsense inference, where models must select the most plausible continuation of a given context. PIQA (Physical Interaction Question Answering) (Bisk et al., 2020) focuses on physical commonsense reasoning, requiring models to choose the more feasible solution to everyday tasks. WinoGrande (Sakaguchi et al., 2021) is a coreference-resolution benchmark that tests commonsense reasoning through pronoun references that cannot be resolved from syntax alone.

## C Experimental Setting

The experimental settings for these tasks are summarized in Table 8.

| Tasks | 4090 GPUs |
|---|---|
| ImageNet | 8 × |
| CIFAR-10/100 | 1 × |
| GLUE pretraining | 8 × |
| QAT and CRT pretraining | 8 × |

Table 8: GPU resource requirements in our experiments.

| Model | T | HellaSwag | PIQA | WinoGrande | CRT Avg. (%) | QAT Avg. (%) |
|---|---|---|---|---|---|---|
| NILIF (WD-SpikingFormer-0.4B) | 4 | 25.9 | 53.4 | 50.2 | 43.2 | 28.4 |
| NASN (WD-SpikingFormer-0.4B) | 4 | 26.0 (+0.1) | 54.6 (+1.2) | 49.4 (-0.8) | 43.3 (+0.1) | 29.4 (+1.0) |
| NILIF (WD-SpikingFormer-1.0B*) | 4 | 26.4 | 55.7 | 50.3 | 44.1 | 31.9 |
| NASN (WD-SpikingFormer-1.0B*) | 4 | 26.4 (+0.0) | 58.0 (+2.3) | 51.0 (+0.7) | 45.1 (+1.0) | 35.5 (+3.6) |

Table 9: Detailed performance comparison.

## D Energy Consumption

Spiking Neural Networks (SNNs) replace conventional multiply-accumulate (MAC) operations with energy-efficient accumulate (AC) operations. In contrast, the energy consumption of Artificial Neural Networks (ANNs) is commonly evaluated by their MACs. Specifically, for a linear layer with input dimension $m$ and output dimension $n$, the energy consumption is given by:

$$E_{\text{Linear}}^{\text{Ann}} = m \times n \times E_{MAC}, \tag{21}$$

Spiking Neural Networks (SNNs) convert MAC-based matrix multiplications into accumulation (AC) operations. For SNNs, considering the same setting as in the ANN case, the theoretical energy consumption of the linear layer can be computed as:

$$E_{\text{Linear}}^{\text{Snn}} = m \times n \times E_{AC} \times fr \times T, \tag{22}$$

where $fr$ is the layer firing rate and $T$ is the number of simulation time steps. This estimate follows prior work (Kundu et al., 2021; Zhou et al., 2023b; 2026b; Panda et al., 2020; Yao et al., 2023b). We assume 45-nm hardware (Horowitz, 2014), with $E_{MAC} = 4.6\,\text{pJ}$ and $E_{AC} = 0.9\,\text{pJ}$. For language models, we estimate energy consumption for inference over a sequence of 512 tokens.

## E Further Performance Analysis

This section provides a more detailed analysis of the experimental results. Across the two language task groups, i.e., question-answering tasks (QAT) and commonsense reasoning tasks (CRT), the base model equipped with NASN consistently achieves higher overall performance than its NILIF counterpart.

The gains from NASN vary across datasets and scales. In smaller settings, the learnable shifts have less opportunity to adapt to diverse activation distributions, so NASN may behave similarly to NILIF. Nevertheless, NASN improves ImageNet top-1 accuracy by 0.12 percentage points on a 0.016B-parameter vision model, the mean GLUE score by 1.2 points on a 0.1B-parameter language model, and QAT accuracy by 1.0 and 3.6 percentage points on the 0.4B- and 1.0B-parameter language models, respectively. These results suggest that the advantage of NASN becomes more pronounced as model size and pretraining-data scale increase, particularly for language tasks.

As reported in Table 9, a few language subtasks show slight degradation. At limited model and pretraining-data scales, the average improvement from NASN is modest, and individual results may be affected by task-specific variance and optimization noise. Minor drops can therefore occur even when average performance improves. With larger models and more pretraining data, the gains become more consistent across both aggregate and per-task results, further suggesting that NASN benefits from scaling.

## F Further Analysis on NASN Scaling

We analyze NASN scaling in WD-SpikingFormer with respect to model size and pretraining-data scale.

**Parameter scaling.** Panels (a) and (b) of Figure 3 show that increasing the model size from 0.4B to 1.0B parameters improves QAT accuracy from 29.4% to 29.8% and CRT accuracy from 43.3% to 43.5%.

**Pretraining data scaling.** Panels (c) and (d) of Figure 3 show the effect of increasing the number of pretraining tokens for NASN (1.0B) on QAT and CRT. Increasing the pretraining corpus from 0.1B to 0.5B tokens improves NASN (1.0B) from 29.8% to 35.5% on QAT and from 43.5% to 45.1% on CRT. The 5.7-percentage-point QAT gain indicates that NASN can benefit from larger-scale pretraining.

## G Time-Step Ablation and Training Overhead

As shown in Table 10, increasing the number of time steps further improves the GLUE score of WE-SpikingFormer with NASN.

In addition, since the learnable neuron parameters are defined at the layer level, the overall parameter overhead is minimal. As a result, the training time remains nearly identical to that of the ILIF family, with only a negligible increase.

## H DVS Dataset Results

Table 11 reports results on DVS128 Gesture. NASN also outperforms NILIF on this event-based dataset.

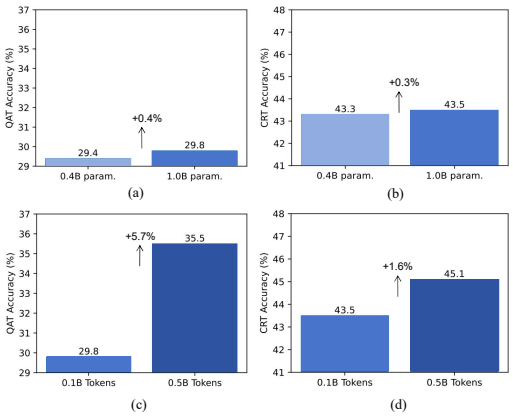


Figure 3: Parameter and pretraining-data scaling of WD-SpikingFormer with NASN. Panels (a) and (b) show QAT and CRT performance as model size increases; panels (c) and (d) show performance as the number of pretraining tokens increases.

| WE-SpikingFormer | GLUE Acc (%) |
|---|---|
| NILIF ($1 \times 4$) | 66.3 |
| NASN ($1 \times 4$) | 67.5 |
| NASN ($2 \times 4$) | 69.3 |

Table 10: Ablation study for time steps in NASN.

| Tasks | LIF($\times$1) | NILIF($\times$4) | NASN($\times$4) |
|---|---|---|---|
| T=10 | 96.2 | 96.9 | 97.2 (+0.3) |

Table 11: DVS128 results using Spikingformer backbone.

| Method | CIFAR-10 | | | CIFAR-100 | | |
|---|---|---|---|---|---|---|
| | Param | T | Acc | Param | T | Acc |
| Spikformer | 5.76 | 4 | 94.80 | 5.76 | 4 | 76.95 |
| Spikformer | 9.32 | 4 | 95.51 | 9.32 | 4 | 78.21 |
| SD-Transformer | 10.28 | 4 | 95.60 | 10.28 | 4 | 78.4 |
| Spikingformer | 9.32 | 4 | 95.81 | 9.32 | 4 | 79.21 |
| Spikingformer$^\dagger$ | 9.32 | 4 | 95.95 | 9.32 | 4 | 80.37 |
| Transformer(ANN) | 9.32 | 1 | 96.73 | 9.32 | 1 | 81.02 |
| **Ours** | 9.32 | 4 | **96.27** | 9.32 | 4 | **81.11** |

Table 12: Comparison on CIFAR-10 and CIFAR-100. "Param" denotes the number of parameters in millions, "Acc" denotes top-1 accuracy (%), and "T" denotes the number of time steps. Our model uses the Spikingformer$^\dagger$ architecture.

## I CIFAR Dataset Results

To further validate the effectiveness and generalization of our method, we conduct experiments on the CIFAR-10 and CIFAR-100 datasets Krizhevsky (2009). We compare our method with Spikformer Zhou et al. (2023b), SD-Transformer Yao et al. (2023a), and Spikingformer Zhou et al. (2026b). Table 12 reports the results. For a controlled comparison, our model uses the same architecture and experimental configuration as the Spikingformer$^\dagger$ baseline. Our model achieves 96.27% accuracy on CIFAR-10 and 81.11% on CIFAR-100, exceeding Spikingformer$^\dagger$ by 0.32 and 0.74 percentage points, respectively.