MTA-SNN: REVITALIZING CONVOLUTIONAL SPIKING NEURAL NETWORKS VIA MULTI-THRESHOLD PARALLEL NEURONS AND GLOBAL-LOCAL AGGREGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking Neural Networks (SNNs) offer a promising avenue for energy-efficient computing via event-driven paradigms, but a significant trade-off exists between the hardware-friendly nature of convolutional architectures and the superior representational capacity of Transformer-based models, whose reliance on complex attention mechanisms can complicate asynchronous deployment. To bridge this gap, we propose MTA-SNN, which revitalizes convolutional SNNs by integrating attention-like capabilities using fully deployable primitives. Our approach introduces three core innovations: (1) Multi-Threshold Parallel Neurons (MTPN), which employ heterogeneous firing thresholds to expand information capacity in the channel dimension; (2) Global Local Aggregation (GLA) that captures long-range dependencies; and (3) Multi-Scale Position Encoding (MSPE) via dilated convolutions to explicitly enrich spatial context. Extensive experiments demonstrate that MTA-SNN establishes new state-of-the-art results among low-latency SNNs, achieving 83.07% on CIFAR-100 and 78.37% on ImageNet with a single timestep. This performance significantly outperforms existing convolutional SNNs and matches transformer-based counterparts requiring multiple timesteps, proving that high-performance vision SNNs can be realized within a strictly neuromorphic-friendly convolutional framework.

1 INTRODUCTION

Spiking neural networks (SNNs) transmit discrete spikes and offer substantial energy savings over artificial neural networks (ANNs) (Maass, 1997; Roy et al., 2019). Event-driven execution on Intel Loihi (Davies et al., 2018) and IBM TrueNorth (Merolla et al., 2014) can reduce energy consumption by orders of magnitude (Akopyan et al., 2015). However, binary outputs in standard Leaky Integrate-and-Fire (LIF) neurons lose information (Gerstner & Kistler, 2002): fixed thresholds can miss weak inputs or saturate on strong ones, hindering fine-grained discrimination of high-frequency temporal patterns (Izhikevich, 2004).

Early SNNs adopted convolutional architectures, whose local operations support asynchronous inference on Loihi (Davies et al., 2018) and TrueNorth (Akopyan et al., 2015). Yet spike-induced information loss continues to limit their accuracy relative to comparable ANNs. Spike transformers improve performance: on ImageNet, transformer-based SNNs reach 77.64% (Zhou et al., 2024b), compared with 73.4% for convolutional SNNs using more parameters and energy (Shen et al., 2025). Their multi-path spike matrix multiplication, however, complicates direct asynchronous deployment.

We propose MTA-SNN to combine transformer-like representation with a neuromorphic-native convolutional design. Global Local Aggregation (GLA) captures long-range dependencies using depthwise convolutions and pooling, avoiding the synchronization challenges of multi-path spike matrix multiplication. Inspired by cortical diversity (Henze & Buzsáki, 2001; Cudmore & Turrigiano, 2004), Multi-Threshold Parallel Neurons (MTPN) mitigate binary quantization loss through heterogeneous thresholds. Our analysis shows that this parallel encoding expands information capacity without the temporal expansion ($T > 1$) and latency of serial integration. Multi-Scale Position Encoding (MSPE) uses dilated convolutions to enrich spatial context. Together, these modules bring transformer-like capabilities to convolutional SNNs.

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.

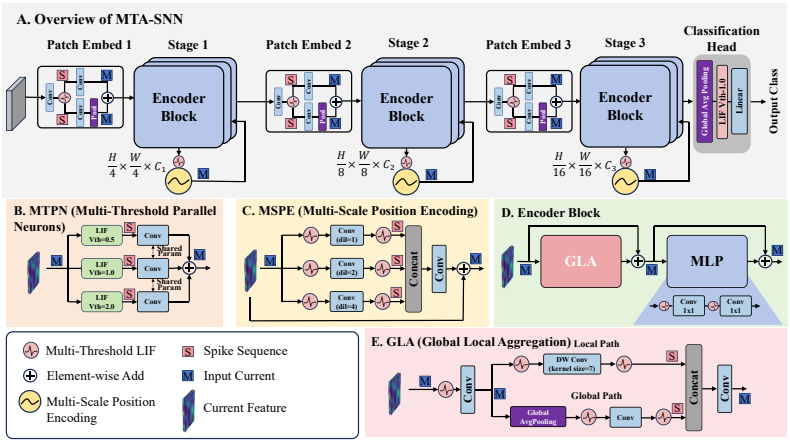

Figure 1: The overview of MTA-SNN architecture. (A) Three-stage hierarchical design with progressive downsampling ($H/4 \times W/4 \rightarrow H/8 \times W/8 \rightarrow H/16 \times W/16$). Each stage contains Patch Embed via convolution and pooling, and Encoder Blocks. (B) **Multi-Threshold Parallel Neurons (MTPN)** with multi-thresholds sharing convolutional weights across parallel pathways, (C) **Multi-Scale Position Encoding (MSPE)** using dilated depthwise convolutions (rates $\{1, 2, 4\}$), (D) **Encoder Blocks** contains Global Local Aggregation and MLP with 1×1 convolutions. (E) **Global Local Aggregation (GLA)** decomposing attention into depthwise convolution (local) and average pooling (global).

Our main contributions are summarized as follows:

- We propose MTA-SNN, a high-performance SNN architecture that successfully adapts transformer-inspired mechanisms (global attention and position encoding) into a purely convolutional paradigm. This design ensures full compatibility with asynchronous neuromorphic hardware while significantly surpassing traditional convolutional SNNs.
- We introduce the MTPN module and provide a theoretical proof and experiments demonstrating that parallel multi-threshold processing yields higher information capacity and better gradient flow than increasing simulation time steps.
- We have provided sufficient experimental evidence to prove the effectiveness of the proposed model. We achieve state-of-the-art results on CIFAR-10/100 (97.03%/83.16%) and ImageNet (78.37%) with $T = 1$. Our work proves that high-performance vision SNNs can be realized within a strictly neuromorphic-friendly convolutional framework.

2 Related Work

2.1 Spiking Neural Network

The Leaky Integrate-and-Fire (LIF) neuron serves as the foundational model in SNNs, balancing biological plausibility with computational efficiency (Gerstner & Kistler, 2002). Recent advances have introduced enhanced variants including Parametric LIF (PLIF) with learnable membrane dynamics (Fang et al., 2021), Adaptive Exponential Integrate-and-Fire models (Brette & Gerstner, 2005), and Izhikevich neurons (Izhikevich, 2003; 2004). Despite these innovations, all existing variants fundamentally operate with single firing thresholds—whether fixed or learnable—which constrains their ability to simultaneously encode signals across different intensity ranges.

Recent SNN research has explored two architectural directions. Convolutional SNNs naturally align with neuromorphic hardware through local operations and static synaptic weights, with recent architectures including QP-Resnet20 achieving 75.29% on CIFAR-100 (Wei et al., 2025) and Resnet34 reaching 73.4% on ImageNet (Shen et al., 2025). Transformer-based SNNs, including Spikeformer (Zhou et al., 2022), Spikingformer (Zhou et al., 2023), and QKformer (Zhou et al., 2024a), achieve stronger accuracy through spike-based attention mechanisms, with recent work reaching 77.64% on

ImageNet (Zhou et al., 2024b). At similar accuracy levels, transformer-based approaches demonstrate significant parameter efficiency: ResNets require 12.63M parameters for 74.47% on CIFAR-100 (Deng et al., 2022), while QKformer achieves 81.15% with only 6.74M parameters (Zhou et al., 2024a). Despite apparent differences, both paradigms share fundamental design principles: hierarchical feature extraction with progressive spatial downsampling (Liu et al., 2021) and convolutional operations as core primitives (Dosovitskiy, 2020; Zhou et al., 2022), suggesting innovations from both domains can be synergistic (Wang et al., 2021; Dong et al., 2022).

2.2 Vision Transformer

Vision Transformers leverage self-attention to model long-range dependencies through query-key-value operations (Vaswani et al., 2017). Efficient variants reduce computational complexity through linear attention (Katharopoulos et al., 2020; Wang et al., 2020), window-based mechanisms (Liu et al., 2021; Dong et al., 2021), and deformable attention (Zhu et al., 2020; Xia et al., 2022). Multi-scale architectures (Chen et al., 2021; Yuan et al., 2021) capture hierarchical representations across spatial resolutions. For Transformer-based SNNs, Spikeformer (Zhou et al., 2022) pioneered spike-driven self-attention in the binary spike domain. Subsequent work includes Spikingformer with enhanced temporal dynamics (Zhou et al., 2023), QKformer with query-key reparameterization (Zhou et al., 2024a), and event-driven attention for neuromorphic vision (Gehrig et al., 2019; Messikommer et al., 2020). Recent advances achieve 77.64% on ImageNet (Zhou et al., 2024b).

Position encoding is also critical for spatial awareness. Vision Transformer employs learnable absolute embeddings (Dosovitskiy, 2020), while Conditional Position Encoding uses depthwise convolutions for translation equivariance (Chu et al., 2021). Relative encoding methods (Shaw et al., 2018; Hu et al., 2019) capture pairwise spatial relationships. Spikeformer for vision lacks explicit position encoding, relying solely on implicit information from hierarchical feature maps, potentially limiting spatial relationship modeling compared to explicit schemes in standard vision transformers (Liu et al., 2021; Wang et al., 2021).

3 Preliminary

3.1 Spiking Neuron Models

The Leaky Integrate-and-Fire (LIF) neuron constitutes a foundational computational primitive in Spiking Neural Networks (SNNs), emulating biological neuronal dynamics through simplified biophysical principles(Gerstner & Kistler, 2002). Its membrane potential $V_m(t)$ evolves according to a leaky integrator differential equation: $\tau_m \frac{dV_m(t)}{dt} = -(V_m(t) - V_{\text{rest}}) + \sum_i w_i s_i(t)$, where τ_m denotes the membrane time constant, V_{rest} the resting potential, w_i synaptic weights, and $s_i(t)$ presynaptic spike trains. Upon reaching threshold V_{th}, the neuron emits an action potential and the membrane potential $V_m(t)$ resets.

3.2 Attention Mechanisms in SNNs

Spikeformer introduced spike-based self-attention operating in the binary spike domain (Zhou et al., 2022). Their approach computes attention using spike-form query, key, and value representations without softmax normalization or dense floating-point matrix operations. The attention computation proceeds through the following formulation. First, the spike-form query, key, and value projections are generated via: $\mathcal{I} = \text{SN}(\text{BN}(X \cdot W_{\mathcal{I}})), \quad \mathcal{I} \in \{Q, K, V\}$ where $\text{SN}(\cdot)$ denotes the spiking neuron transformation, $\text{BN}(\cdot)$ represents batch normalization, and $W_{\mathcal{I}}$ are learnable projection matrices. The resulting spike tensors satisfy $Q, K, V \in \{0, 1\}^{T \times N \times D}$, where T denotes temporal steps, N represents sequence length, and D indicates the feature dimension. The attention output is subsequently computed through: $\text{SSA}(Q, K, V) = \text{SN}\left(QK^T V \cdot s\right)$ where $s \in \mathbb{R}^+$ serves as a scaling parameter.

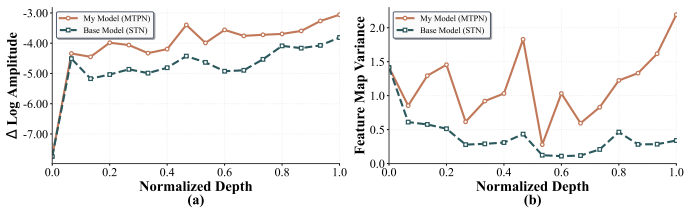

Figure 2: Information preservation across network depth. (a) High-frequency components: MTPN maintains higher amplitudes than Single-lif. (b) Feature variance: MTPN sustains richer representations.

4 METHOD

4.1 OVERALL ARCHITECTURE

To bridge the gap between the high performance of spike transformers and the hardware-friendliness of convolutional SNNs, we propose MTA-SNN, a convolutional meta-architecture designed to be both powerful and neuromorphic-native. Instead of directly adopting the attention mechanism of transformers, we emulate their functional advantages using purely convolutional and pooling operations. As illustrated in Figure 1.A, MTA-SNN employs a hierarchical three-stage structure.

Our design introduces three core innovations, each addressing a fundamental limitation in conventional SNNs: (1) **Multi-Threshold Parallel Neurons (MTPN)** that use multiple parallel LIF neurons to replace the standard LIF neurons to combat the information bottleneck of spike binarization. (2) **Global Local Aggregation (GLA)** uses convolution and pooling layers to simulate the attention mechanism, overcoming the limited receptive field of pure convolutional SNNs and enabling the modeling of long-range spatial dependencies. (3) **Multi-Scale Position Encoding (MSPE)** is injected into each stage to provide explicit spatial context, compensating for the translation-equivariance of convolutions.

4.2 MULTI-THRESHOLD PARALLEL NEURONS

Standard LIF neurons suffer from a critical information bottleneck: their binary spike output discards all sub-threshold information and saturates for strong inputs. The conventional remedy is to increase simulation timesteps, allowing neurons to encode signal intensity in their firing rate. However, this may actually be inefficient and increase latency and computational costs, thus reducing efficiency.

Inspired by biological neural circuits that exhibit heterogeneous firing thresholds enabling distributed encoding across signal intensities (Softky & Koch, 1993; Azouz & Gray, 2000), we propose a more efficient, biologically inspired alternative: parallel processing in the feature dimension instead of serial processing in the time dimension. Our Multi-Threshold Parallel Neuron (MTPN) module (Figure 1.B) processes an input current X using multiple parallel LIF neurons with heterogeneous firing thresholds $\{V_{th}^{low}, V_{th}^{mid}, V_{th}^{high}\}$ (e.g., $\{0.5, 1.0, 2.0\}$). Each neuron independently generates a spike train according to:

$$S_i^t = \Theta(V_i^t - V_{th,i}), \quad i \in \{1, 2, 3, ...\} \tag{1}$$

where $\Theta(\cdot)$ is the Heaviside step function. The multiple threshold levels operate concurrently, generating independent spike trains $\{S_1^t, S_2^t, S_3^t, ...\}$ that encode the input at different sensitivity scales. In all CIFAR experiments, the threshold set is fixed to $\{0.5, 1.0, 2.0\}$ rather than tuned per dataset. This set is centered at the standard LIF threshold of 1.0, with a lower threshold of 0.5 for weak responses and a higher threshold of 2.0 for strong responses. The design implements a coarse-to-strong response decomposition within one timestep. For larger-scale ImageNet experiments, we add one high threshold and use $\{0.5, 1.0, 2.0, 2.5\}$ to cover a wider activation range. Sensitivity analysis and selection guidelines are provided in Appendix F.

We propagate these three output spikes through the pre-existing convolutional layers of the network architecture backbone, respectively. Hence, we obtain three intermediate feature maps. These feature

representations capture complementary information: the low-threshold path preserves fine-grained temporal details, the medium-threshold path encodes salient features, and the high-threshold path extracts robust patterns resistant to noise. This design is not merely heuristic; we also provide a theoretical guaranty that this parallel processing paradigm fundamentally surpasses temporal integration in terms of representational ability.

Theorem 1 (Equivalence of Spatial and Temporal Resolution)**.** *Let $\mathcal{Q}(x)$ denote the discrete quantization level of a continuous input $x \in \mathbb{R}$. (1) An MTPN with K ordered thresholds $\mathcal{V} = \{v_1 < v_2 < \cdots < v_K\}$ at $T = 1$ induces a quantization map with $K + 1$ distinct levels, providing an information capacity of $\log_2(K + 1)$ bits. (2) A standard LIF neuron with a single threshold v_{th} and hard reset requires a minimum of $T \geq K$ simulation steps to achieve an equivalent number of distinct firing-rate levels.*

The proof is provided in Appendix A.2. Theorem 1 and Proposition 2 theoretically justify why MTA-SNN achieves state-of-the-art performance with ultra-low latency ($T = 1$): we effectively expand the information bottleneck in the channel dimension rather than the temporal dimension. Beyond representation, this architecture also facilitates training stability (Proposition 1). Here, figure 3 presents Fourier spectrum analysis of single-threshold versus multi-threshold LIF neurons processing composite inputs with frequency components at 3 Hz, 15 Hz, and 40 Hz. Single-threshold neurons (middle) exhibit severe spectral distortion, attenuating high frequencies and introducing spurious harmonics. Multi-threshold neurons (right) faithfully preserve all frequency components with minimal distortion, closely matching the input spectrum (left). This empirically confirms that multi-threshold architectures achieve broader frequency bandwidth and superior signal fidelity. Visual evidence supporting this complementary information capture across different thresholds is provided in Appendix B, where reconstruction analysis demonstrates that heterogeneous thresholds achieve 24% lower error compared to stacked uniform-threshold neurons by distinctly encoding weak signals, salient structures, and high-intensity patterns.

A postsynaptic aggregation mechanism combines these intermediate feature maps through element-wise summation: $F_{out} = \sum_{i=1}^{3} \text{Conv}(S_i^t)$. This aggregation mimics dendritic integration in biological neurons, where postsynaptic targets receive and combine inputs from heterogeneous presynaptic populations (London & Häusser, 2005; Magee, 2000). In actual deployment, three parallel weight-sharing convolutional modules can be used and accumulated on the input current.

4.3 Multi-Scale Position Encoding

While the inherent translation invariance of convolutional networks facilitates object detection, it constrains the modeling of complex spatial semantics and relative positioning. In contrast, Transformer architectures incorporating absolute position embeddings excel at capturing these spatial relationships. We introduce a Multi-Scale Position Encoding (MSPE) module to explicitly inject spatial context (Figure 1.C). MSPE is inspired by the Conditional Position Encoding (CPE/CPVT) design (Chu et al., 2021), which uses depthwise convolution to inject input-conditioned positional bias. Unlike CPVT, which employs a single-scale depthwise convolution in frame-based ViTs, MSPE is designed for ultra-low-latency spiking inference and uses parallel dilated depthwise branches to encode multi-scale positional cues in one timestep. The MSPE modules are inserted after the first encoder block of each stage via residual connections. Given feature maps $X \in \mathbb{R}^{TB\times C\times H\times W}$ from the first attention encoder, MSPE generates position-dependent representations through three parallel branches: $P_i = \text{MTPN}(\text{DWConv}_{3\times3}^{d=i}(X)), i = \{1, 2, 4\}$, where $\text{DWConv}_{k\times k}^{d}$ denotes depthwise convolution with kernel size k and dilation rate d. Each branch captures spatial features at different receptive field scales: dilation-1 encodes local neighborhood patterns (effective receptive field 3×3), dilation-2 captures mid-range dependencies (5×5), and dilation-4 extracts broader spatial context (9×9). The three scale-specific feature maps are concatenated channel-wise and fused through a 1×1 convolution. These multi-scale branches allow the network to perceive spatial dependencies across varying receptive fields (from local neighborhoods to broader contexts). This implicit position information is fused into the feature map, enhancing the SNN's ability to model geometry and structure without introducing non-spiking operations.

4.4 Global Local Aggregation

A primary advantage of transformers is their ability to model long-range dependencies via the self-attention mechanism. However, in an event-driven system, calculating $Q \times K$ requires buffering spikes to align separate time streams, destroying the efficiency gains of sparsity. To resolve this, we do not aim to design a generic complex module but rather to deconstruct the functional role of self-attention into hardware-native primitives. We propose Global Local Aggregation (GLA), which approximates the global modeling capability of Transformers using only convolution and pooling operations that support stream processing.

Our GLA decomposes attention into two parallel, hardware-friendly pathways (Figure 1.E). Given an input feature map X, GLA splits the computation: 1) **Local Path**: A depthwise convolution (DWConv) with a large kernel (e.g., 7×7) is applied to capture fine-grained spatial patterns and local context. This efficiently emulates the local receptive fields that attention mechanisms also model; 2) **Global Path**: A global average pooling operation reduces the spatial dimensions of X to 1×1, producing a channel-wise descriptor vector. This vector, representing a summary of the entire feature map, is then processed by a 1×1 convolution to generate a global context signal. This path mimics the ability of attention to aggregate information across the entire input. Finally, the outputs from both paths are concatenated and fused by a final 1×1 convolution. This allows the network to dynamically weigh the importance of local details against global context for each feature. By using only convolutions and pooling, GLA achieves a global effective receptive field without sacrificing compatibility with asynchronous, event-driven computation. These operations map directly to synaptic current accumulation without requiring floating-point scaling factors. Thus, GLA serves as a neuromorphic-friendly global-local aggregation alternative to dot-product self-attention, rather than as an exact mathematical replacement.

4.5 Neuromorphic Compatibility

MTA-SNN is designed around neuromorphic-compatible primitives: synaptic weighting, neuronal integration, and threshold-based firing. Convolution, average pooling, and residual accumulation can be expressed through weighted spike operations, motivating asynchronous deployment subject to device-specific mapping and capacity constraints. We distinguish this design compatibility from demonstrated execution: Appendix L reports simulator evidence, while Appendix L.2 presents a preliminary Speck2e physical-chip experiment. A compact MTSNN variant preserving the tied-branch MTPN mechanism runs on four DynapCNN cores and achieves 93.63% accuracy on an N-MNIST subset. This validates real-chip execution of the multi-threshold design; it does not establish deployment of the full MTA-SNN architecture.

MTPN trades temporal depth for spatial parallelism: three shared-weight convolutional paths triple the Synaptic Operations (SOPs) of affected layers relative to a single path, but support accurate single-timestep inference. Event-driven latency depends on both temporal rounds T and routed spike events $\mathcal{E}$, modeled as $\mathcal{L} = \alpha T + \beta \mathcal{E}$. Replacing a $K = 1, T = 3$ LIF baseline with $K = 3, T = 1$ MTPN is beneficial when the additional event cost is smaller than the cost of two temporal rounds. Under matched CIFAR-100 settings, per-sample GPU runtime decreases from 3.90 to 1.55 ms (Appendix A.1); this is a GPU measurement, not a physical-chip latency result.

Table 1: Performance on CIFAR-10/100. Superior accuracy with single timestep.

Method	Params (M)	T	CIFAR-10	CIFAR-100
TET(Deng et al., 2022)	12.63	4	94.44	74.47
Spikformer(Zhou et al., 2022)	9.32	4	95.51	78.21
Spikingformer(Zhou et al., 2023)	9.32	4	95.95	80.37
QKformer(Zhou et al., 2024a)	6.74	4	96.18	81.15
Maxformer(Fang et al., 2025)	6.60	4	**97.04**	82.65
Resnet18(Baseline)	10.18	1	93.95 ± 0.23	74.01 ± 0.15
MTResnet18 (Ours)	10.18	1	95.92 ± 0.06	78.32 ± 0.14
MTA-SNN (Ours)	7.44	1	97.03 ± 0.11	$\mathbf{83.07 \pm 0.11}$

Table 2: Performance on ImageNet. State-of-the-art accuracy with single timestep.*:The QKformer performance metrics are derived from the experimental validation presented in (Fang et al., 2025).

Method	Architecture	Params (M)	Energy (mJ)	T	Top-1
SpikePack(Shen et al., 2025)	Resnet18	11.1	-	4	70.6
	Resnet34	21.8	-	4	73.4
Spikformer(Zhou et al., 2022)	Spikformer-8-384	16.81	7.73	4	70.24
	Spikformer-8-512	29.68	11.58	4	73.38
	Spikformer-8-768	66.34	21.48	4	74.81
*QKformer(Zhou et al., 2024a)	HST-10-384	16.47	5.52	4	76.48
	HST-10-768	64.96	6.79	1	77.78
Spiking Transformer(Guo et al., 2025)	Spiking Transformer-8-384	16.81	-	4	74.04
	Spiking Transformer-8-512	29.68	-	4	76.28
*Max-former(Fang et al., 2025)	Max-10-384	16.23	4.89	4	77.82
	Max-10-512	28.65	2.50	1	75.47
MTA-SNN (Ours)	MTA-SNN-10-384	17.31	4.92	1	76.78
	MTA-SNN-Maxpool-10-384	17.31	5.09	1	**77.84**
	MTA-SNN-10-512	30.49	7.76	1	77.51
	MTA-SNN-10-768	66.34	14.11	1	**78.37**

5 EXPERIMENTS

We evaluate MTA-SNN on both static and neuromorphic benchmarks. For static image classification, we use CIFAR-10/100 and ImageNet-1K with a single simulation timestep ($T = 1$). For neuromorphic validation, we use CIFAR10-DVS with $T = 16$ to match prior work. All models are trained from scratch using surrogate gradient backpropagation. Unless otherwise stated, our CIFAR experiments adopt MTPN with three parallel thresholds, and ImageNet experiments use MTPN with four thresholds. Detailed training hyperparameters, data augmentation, and energy estimation protocols are detailed in Appendix D and E.

5.1 COMPARISON TO EXISTING WORKS

Static CIFAR benchmarks. Table 1 summarizes results on CIFAR-10/100. MTA-SNN achieves 97.03% / 83.16% top-1 accuracy with a single timestep ($T = 1$), surpassing Transformer-based SNNs requiring $T = 4$ while using fewer or comparable parameters. This single-timestep operation provides substantially lower inference latency on neuromorphic hardware, where latency scales with timesteps. Compared to Spikformer (Zhou et al., 2022) (95.51% / 78.21%, $T = 4$), MTA-SNN yields gains of +1.52% / +4.95%. Relative to QKformer (Zhou et al., 2024a) (96.18% / 81.15%), MTA-SNN improves accuracy by +0.85% / +2.01% under the same $T = 1$ setting. In addition, our ResNet-based variant MTResNet18, which only replaces standard LIF with MTPN while maintaining a pure convolutional backbone, already has 95.92% / 78.32% in $T = 1$, improving the baseline of standard spiking ResNet18 by +4.31% in CIFAR-100. This confirms that multi-threshold neurons alone significantly enhance the representational capacity of convolutional SNNs. This latency advantage is directly quantified in Appendix A.1, where under matched settings K=3, T=1 achieves a 2.52× lower per-sample runtime than K=1, T=3 (1.55 ms vs. 3.90 ms). In the same matched regime, Appendix I further shows reduced training overhead, with lower peak GPU memory and faster per-sample runtime while maintaining higher accuracy.

Results on ImageNet As shown in Table 2, MTA-SNN-10-768 achieves 78.37% top-1 accuracy at $T = 1$, surpassing Spikingformer-8-768 (Zhou et al., 2023) (77.64%, $T = 4$) and QKformer HST-10-768 (Zhou et al., 2024a) (77.78%, $T = 1$). With a comparable model size, MTA-SNN-10-512 (77.51%) outperforms Spikingformer-8-512 (Zhou et al., 2023) (76.54%), demonstrating that our convolutional, neuromorphic-native design can match or exceed the performance of transformer-based SNNs on large-scale vision tasks. Compared to convolutional SNN, MTA-SNN demonstrates substantial advantages: MTA-SNN-10-512 (77.51%) surpasses SpikePack-Resnet34 (Shen et al., 2025) (73.4%, $T = 4$) by +4.11%, T-RevSNN-Resnet18(512) (Hu et al., 2024) (73.2%, $T = 4$) by +4.31%, and Parallel SNN-Resnet34 (Hao et al., 2024) (72.90%, $T = 4$) by +4.61%. Even at the 384-channel configuration, MTA-SNN-10-384 (76.78%, $T = 1$) outperforms all prior convolutional SNNs while providing single-timestep inference. For QKformer, we report the strict-spiking re-evaluated results from MaxFormer (Fang et al., 2025), because the originally reported QKformer shortcut contains real-valued residual signals and is therefore not fully spike-native. Beyond classification,

Appendix H reports transfer results on COCO detection and ADE20K segmentation, showing that MTA-SNN maintains strong dense prediction performance with fewer parameters and lower estimated energy than Meta-SpikeFormer (Yao et al., 2024).

Neuromorphic CIFAR10-DVS. On CIFAR10-DVS (Table 3), MTA-SNN with max pooling achieves 84.8% at $T = 16$, exceeding QKformer's 84.0% (Zhou et al., 2024a). Hardware-friendly average pooling achieves 82.4%, a 2.4-point decrease. Max pooling preserves salient spikes in sparse event streams, whereas average pooling can dilute them across empty regions and reduce temporal precision (Gehrig et al., 2019). The average-pooling result remains competitive, illustrating the accuracy–deployability trade-off.

Table 3: Performance on CIFAR10-DVS neuromorphic dataset.*:All MaxPooling layers in MaxFormer were substituted with AveragePooling. The reported results for this variant are sourced from the original MaxFormer study(Fang et al., 2025)

Method	Params(M)	T	Accuracy
QP-SNN(Wei et al., 2025)	-	10	82.1
T-RevSNN(Hu et al., 2024)	-	16	79.2
S-Transformer(Yao et al., 2023)	2.57	16	80.0
QKformer(Zhou et al., 2024a)	1.50	16	84.0
*Maxformer(Fang et al., 2025)	1.45	16	79.8
Maxformer(Fang et al., 2025)	1.45	16	84.2
MTA-SNN-Avgpool (Ours)	1.80	16	82.4
MTA-SNN-Maxpool (Ours)	1.80	16	**84.8**

5.2 Ablation Studies and Analysis

We conduct ablations on CIFAR-100 to quantify the contribution of each component in MTA-SNN. Table 4 reports top-1 accuracy.

MTPN. Replacing the single-threshold neurons (STN) with our MTPN while keeping GLA and no position encoding (row (f)) yields the largest gain: accuracy increases from 79.84% (baseline with GLA, row (b)) to 82.53% (+2.69%). When combined with multi-scale MSPE (row (g)), the full MTA-SNN achieves 83.16%, representing a +3.32% improvement over the baseline with GLA, or +3.54% over the original baseline with SSA (row (a), 79.62%). Crucially, we investigate whether this gain stems from threshold diversity or merely increased parameter capacity. We replace the heterogeneous thresholds $\{0.5, 1.0, 2.0\}$ with three identical thresholds ($V_{th} \equiv 1.0$, denoted as MTPN (single), row (h)). MTPN (single) achieves only 80.15%, far below heterogeneous MTPN with the same MSPE configuration (83.16%, row (g)). This 3.01% gap confirms that diverse thresholds are essential to the observed gains, as they enable a richer decomposition of input intensity instead of simply increasing depth or width. We further investigate the effect of threshold count K on performance (detailed ablation in Appendix F). Results show that accuracy improves progressively from $K = 1$ (81.18%) to $K = 3$ (83.16%, +1.98%), but exhibits diminishing returns at $K = 4$ (83.36%, +0.20%). The $K = 3$ configuration achieves 90.8% of maximum performance gain while maintaining computational efficiency, making it optimal for CIFAR datasets. Following (Park & Kim, 2022), we examine information preservation across network depth. Figure 2 shows: (a) MTPN maintains higher high-frequency amplitudes ($\Delta \approx -3.0$ to -2.8) versus Single-lif ($\Delta \approx -5.0$ to -3.8); (b) MTPN sustains higher feature variance (0.6-2.3) versus Single-lif (0.1-1.4). High-frequency components encode fine-grained spatial details (Park & Kim, 2022; Yin et al., 2019), while higher variance indicates diverse representations that prevent feature collapse (Bardes et al., 2021; Chen et al., 2020; Morcos et al., 2018).

MSPE. Rows (c)–(e) analyze the proposed MSPE on top of the baseline with GLA. Adding a single-scale MSPE (row (c)) improves accuracy from 79.84% to 80.58% (+0.74%), while the multi-scale MSPE (dilation rates $\{1, 2, 4\}$, row (d)) further boosts performance to 81.18% (+1.34%). This demonstrates that wider receptive fields are important for effective spatial encoding. A broader matched comparison against CPE/CPVT (Chu et al., 2021), QKformer-style implicit PE (Zhou et al., 2024a), Gray-PE (Gray, 1953), and Log-PE (Vaswani et al., 2017; Tancik et al., 2020) is provided in Appendix G, where MSPE achieves the best result under identical $T = 1$ settings. To disentangle

Table 4: Ablation on CIFAR-100. MTPN (single): replacing three heterogeneous thresholds in MTPN with three identical thresholds $V_{th} = 1.0$. "1×1" kernel: replacing the 3×3 convolutions in MSPE with 1×1 (no spatial context). Single-scale: using a single dilation rate $d = 1$ instead of $\{1, 2, 4\}$.

ID	Neuron type	Position Encoding	Attention	Top-1 (%)
(a) Baseline	STN	None	SSA	79.62
(b) +GLA	STN	None	GLA	79.84
(c) + MSPE (single)	STN	MSPE, single-scale	GLA	80.58
(d) + MSPE (multi)	STN	MSPE, multi-scale	GLA	81.18
(e) + PE, 1×1	STN	MSPE, 1×1 kernel	GLA	79.89
(f) + MTPN	MTPN	None	GLA	82.53
(g) MTPN + MSPE (multi)	MTPN	MSPE, multi-scale	GLA	**83.16**
(h) MTPN (single)	TIT-lifs	MSPE, multi-scale	GLA	80.15
(i) GLA → SSA	MTPN	MSPE, multi-scale	SSA	83.10

spatial encoding from parameter count, we replace the 3×3 dilated kernels with 1×1 convolutions (row (e)), which preserves the number of channels but removes spatial context. Accuracy in this case is 79.89%, almost identical to the baseline with GLA (79.84%), showing no improvement. Therefore, the improvement of MSPE comes specifically from encoding multi-scale spatial structure rather than from additional parameters. Moreover, we quantitatively assessed the impact of MSPE on spatial representation by analyzing the attenuation pattern of feature similarity across spatial distances in Appendix C.

GLA. Finally, we compare our GLA with Spikformer-style SSA. The original baseline with SSA (row (a)) achieves 79.62%, while replacing SSA with GLA (row (b)) improves accuracy to 79.84% (+0.22%). When using MTPN and multi-scale MSPE, substituting GLA with SSA (row (i)) yields 83.10%, which is slightly lower than our full MTA-SNN with GLA (83.16%, row (g)), showing a marginal difference of +0.06%. This indicates that GLA can match—and slightly exceed—the representational energy of dot-product attention while being strictly neuromorphic-native. Figure 6 provides visual confirmation, showing that GLA heatmaps accurately localize semantic objects. This interpretation is further validated in Appendix J, where controlled probes show that the Global branch primarily contributes shared context while the Local branch provides spatial selectivity. Consistently, Appendix K shows that under strict T=1 spike settings, GLA is more robust than lightweight CNN attention modules such as SE-Block (Hu et al., 2018) and CBAM (Woo et al., 2018).

6 Conclusion

We proposed MTA-SNN, a neuromorphic-native convolutional SNN that narrows the gap to transformer-based SNNs. By introducing MTPN, we expand information capacity in the channel dimension and achieve substantial gains at single-timestep latency. GLA and MSPE further bring transformer-like global modeling and explicit spatial context into a purely convolutional pipeline. Together, these components yield state-of-the-art accuracy on various static datasets with T=1 and competitive results on neuromorphic benchmarks.

In line with our hardware-oriented design, all main experiments use average pooling—a fixed-weight convolution easily implemented in event-driven systems. Consistent with Max-Former (Fang et al., 2025), replacing average pooling with max pooling improves accuracy on sparse event data (e.g., CIFAR10-DVS), but complicates low-latency asynchronous deployment by requiring event buffering. We report average-pooling results as default, with max-pooling numbers on CIFAR10-DVS and ImageNet illustrating the accuracy–deployability trade-off. Future work on spike-friendly pooling may further close this gap.

AI USE STATEMENT

In this work, we used generative AI tools solely to polish the writing, including improving grammar, clarity, concision, and the presentation of author-written content. This assistance was limited to editorial revisions and did not generate scientific contributions. We did not use generative AI to generate research ideas or hypotheses, develop theoretical models or methods, formulate mathematical claims or proofs, design or implement experiments, generate or process research data, or perform scientific analysis or interpretation of results. AI-based translation, transcription, and qualitative or thematic data analysis were not applicable to this work.

We have reviewed all AI-assisted work. The authors manually checked each revised passage against its intended meaning and verified the accuracy of technical statements, experimental values, and references.

We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE transactions on computer-aided design of integrated circuits and systems*, 34(10):1537–1557, 2015.

Rony Azouz and Charles M Gray. Dynamic spike threshold reveals a mechanism for synaptic coincidence detection in cortical neurons in vivo. *Proceedings of the National Academy of Sciences*, 97(14):8110–8115, 2000.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Romain Brette and Wulfram Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of neurophysiology*, 94(5):3637–3642, 2005.

Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 347–356. IEEE, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

Robert H Cudmore and Gina G Turrigiano. Long-term potentiation of intrinsic excitability in lv visual cortical neurons. *Journal of neurophysiology*, 92(1):341–348, 2004.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee micro*, 38(1):82–99, 2018.

Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*, 2022.

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12114–12124. IEEE, 2022.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International conference on machine learning*, pp. 2793–2803. PMLR, 2021.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 5410–5419. IEEE, 2021.

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 *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2641–2651. IEEE, 2021.

Yuetong Fang, Deming Zhou, Ziqing Wang, Hongwei Ren, ZeCui Zeng, Lusong Li, Shibo Zhou, and Renjing Xu. Spiking transformers need high frequency information. *arXiv e-prints*, pp. arXiv–2505, 2025.

Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5633–5643, 2019.

Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

Frank Gray. Pulse code communication. *United States Patent Number 2632058*, 1953.

Yufei Guo, Xiaode Liu, Yuanpei Chen, Weihang Peng, Yuhan Zhang, and Zhe Ma. Spiking transformer: Introducing accurate addition-only spiking self-attention for transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24398–24408. IEEE, 2025.

Zecheng Hao, Qichao Ma, Kang Chen, Yi Zhang, Zhaofei Yu, and Tiejun Huang. Faster and stronger: when ann-snn conversion meets parallel spiking calculation. *arXiv preprint arXiv:2412.13610*, 2024.

DA Henze and György Buzsáki. Action potential threshold of hippocampal pyramidal cells in vivo is increased by recent spiking activity. *Neuroscience*, 105(1):121–130, 2001.

Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3463–3472. IEEE, 2019.

JiaKui Hu, Man Yao, Xuerui Qiu, Yuhong Chou, Yuxuan Cai, Ning Qiao, Yonghong Tian, Bo Xu, and Guoqi Li. High-performance temporal reversible spiking neural networks with $o(l)$ training memory and $o(1)$ inference cost. *arXiv preprint arXiv:2405.16466*, 2024.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

Yangfan Hu, Huajin Tang, and Gang Pan. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):5200–5205, 2021.

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.

Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14 (6):1569–1572, 2003.

Eugene M Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, 2004.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9992–10002. Ieee, 2021.

Michael London and Michael Häusser. Dendritic computation. *Annu. Rev. Neurosci.*, 28(1):503–532, 2005.

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

Jeffrey C Magee. Dendritic integration of excitatory synaptic input. *Nature Reviews Neuroscience*, 1 (3):181–190, 2000.

Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345 (6197):668–673, 2014.

Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *European Conference on Computer Vision*, pp. 415–431. Springer, 2020.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in neural information processing systems*, 31, 2018.

Namuk Park and Songkuk Kim. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022.

Ole Richter, Yannan Xing, Michele De Marchi, Carsten Nielsen, Merkourios Katsimpris, Roberto Cattaneo, Yudi Ren, Yalun Hu, Qian Liu, Sadique Sheik, et al. Speck: A smart event-based vision sensor with a low latency 327k neuron convolutional neuronal network processing pipeline. *arXiv preprint arXiv:2304.06793*, 2023.

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

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, 2018.

Guobin Shen, Jindong Li, Tenglong Li, Dongcheng Zhao, and Yi Zeng. Spikepack: Enhanced information flow in spiking neural networks with high hardware compatibility. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23385–23395. IEEE, 2025.

William R Softky and Christof Koch. The highly irregular firing of cortical cells is inconsistent with temporal integration of random epsps. *The Journal of neuroscience*, 13(1):334–350, 1993.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 548–558. IEEE, 2021.

Wenjie Wei, Malu Zhang, Zijian Zhou, Ammar Belatreche, Yimeng Shan, Yu Liang, Honglin Cao, Jieyuan Zhang, and Yang Yang. Qp-snn: Quantized and pruned spiking neural networks. In *International Conference on Learning Representations*, volume 2025, pp. 96470–96495, 2025.

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *European conference on computer vision*, pp. 3–19. Springer, 2018.

Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4784–4793. IEEE, 2022.

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, 2023.

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. In *International conference on learning representations*, volume 2024, pp. 52885–52907, 2024.

Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *Advances in Neural Information Processing Systems*, 32, 2019.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 558–567, 2021.

Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Zhengyu Ma, Han Zhang, Huihui Zhou, and Yonghong Tian. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv e-prints*, pp. arXiv–2304, 2023.

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, 2024a.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. *arXiv preprint arXiv:2209.15425*, 2022.

Zhaokun Zhou, Kaiwei Che, Wei Fang, Keyu Tian, Yuesheng Zhu, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer v2: Join the high accuracy club on imagenet with an snn ticket. *arXiv preprint arXiv:2401.02020*, 2024b.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A Mathematical Analysis of Multi-Threshold Neurons

We provide a rigorous analysis of why the Multi-Threshold Parallel Neuron (MTPN) module enhances performance from three perspectives: instantaneous information capacity, gradient flow optimization, and function approximation capability.

A.1 Latency Comparison: MTPN ($K = 3, T = 1$) vs. Standard LIF ($K = 1, T = 3$)

To directly address whether MTPN is truly faster in practice than a multi-step single-threshold baseline, we analyze latency under a standard event-driven decomposition and provide empirical timing measurements.

Event-sensitive latency model. For event-driven neuromorphic execution, per-sample latency can be decomposed into temporal-iteration overhead and event-routing overhead:

$$\mathcal{L} = \alpha T + \beta \mathcal{E}, \tag{2}$$

where T is the number of temporal processing rounds, $\mathcal{E}$ is the total routed spike-event count, α is the average control/synchronization cost per round, and β is the average cost per routed event.

Let $(T_m, \mathcal{E}_m)$ denote MTPN ($K = 3, T = 1$) and $(T_b, \mathcal{E}_b)$ denote the standard baseline ($K = 1, T = 3$). MTPN is faster when:

$$\mathcal{L}_m < \mathcal{L}_b \iff \beta(\mathcal{E}_m - \mathcal{E}_b) < \alpha(T_b - T_m) = 2\alpha. \tag{3}$$

This condition makes explicit that MTPN remains latency-favorable as long as the event increase from parallel thresholds is smaller than the temporal overhead saved by removing two recurrent rounds.

Empirical timing evidence. We additionally report end-to-end per-sample latency measured on GPU as a hardware-agnostic proxy for algorithmic execution cost. Under matched implementation settings, MTPN ($K = 3, T = 1$) takes 1.55 ms per sample, whereas the 3-step baseline ($K = 1, T = 3$) takes 3.90 ms per sample. This corresponds to:

$$\text{Speedup} = \frac{3.90}{1.55} \approx 2.52. \tag{4}$$

These results are consistent with the event-sensitive model above and support our central claim: MTPN gains efficiency primarily by reducing temporal iteration overhead while keeping event overhead within an acceptable range.

A.2 Instantaneous Quantization Resolution vs. Temporal Integration

We formally compare the quantization capability of MTPN operating at a single timestep ($T = 1$) against a standard LIF neuron relying on temporal rate coding.

Theorem 1 (Equivalence of Spatial and Temporal Resolution). *Let $\mathcal{Q}(x)$ denote the discrete quantization level of a continuous input $x \in \mathbb{R}$.*

1. *An MTPN with K ordered thresholds $\mathcal{V} = \{v_1 < v_2 < \cdots < v_K\}$ at $T = 1$ induces a quantization map with $K + 1$ distinct levels.*
2. *A standard LIF neuron with a single threshold v_{th} and hard reset requires a minimum of $T \geq K$ simulation steps to achieve an equivalent number of distinct firing-rate levels.*

Proof. **Part 1: Spatial Resolution of MTPN ($T = 1$).**

Let the input current be x. The MTPN consists of K parallel units, producing a spike vector $\mathbf{s} \in \{0, 1\}^K$, where $s_i = \Theta(x - v_i)$.

Since the thresholds are ordered, the output vector follows a thermometer code structure. The aggregated output $y_{spatial}$ is:

$$y_{spatial}(x) = \sum_{i=1}^{K} s_i = \sum_{i=1}^{K} \Theta(x - v_i) \tag{5}$$

This mapping partitions the input space $\mathbb{R}$ into $K + 1$ disjoint intervals: $(-\infty, v_1), [v_1, v_2), \ldots, [v_K, \infty)$. Within each interval, $y_{spatial}$ takes a distinct integer value in $\{0, 1, \ldots, K\}$. Thus, the resolution is $K + 1$ levels.

Part 2: Temporal Resolution of Standard LIF (T steps).

Consider a standard LIF neuron using rate coding. The output metric is the spike count $y_{temporal} = \sum_{t=1}^{T} s[t]$.

Since $s[t] \in \{0, 1\}$, the maximum possible number of distinct values for $y_{temporal}$ is bounded by the set $\{0, 1, \ldots, T\}$, which contains $T + 1$ levels.

To match the resolution of the MTPN (which has $K + 1$ levels), we must have:

$$T + 1 \geq K + 1 \implies T \geq K \tag{6}$$

Conclusion: MTPN achieves deterministically in a single timestep what a standard LIF requires at least K timesteps to approximate via rate coding. This proves that MTPN exchanges channel-wise redundancy for temporal latency reduction without loss of quantization precision. □

A.3 Gradient Flow and Non-Vanishing Probability

Standard Spiking Neural Networks (SNNs) frequently encounter the dead-neuron problem due to the narrow active domain of surrogate gradient functions. The multi-threshold mechanism inherently expands this active learning region.

Proposition 1 (Expansion of Gradient Support)**.** *Let the surrogate gradient function $h'(u)$ possess a finite support $\mathcal{S}_\epsilon = \{u \mid |u| < \epsilon\}$, where u represents the pre-firing membrane potential. Let P_{vanish} denote the probability that the local gradient $\frac{\partial s}{\partial u}$ is zero. Assuming u follows a continuous distribution $p(u) > 0$ over the operational dynamic range, an MTPN parameterized with $K > 1$ strictly distinct thresholds reduces P_{vanish} compared to a standard single-threshold neuron.*

Proof. For a single-threshold LIF neuron with firing threshold v_{th}, the local gradient of the spike output s with respect to the membrane potential u is $h'(u - v_{th})$. By the chain rule, the loss gradient $\frac{\partial \mathcal{L}}{\partial u} = \frac{\partial \mathcal{L}}{\partial s} h'(u - v_{th})$ is non-zero only within the active support region $\mathcal{S}_{single} = (v_{th} - \epsilon, v_{th} + \epsilon)$. The Lebesgue measure of this support is $\mu(\mathcal{S}_{single}) = 2\epsilon$.

In an MTPN utilizing K strictly distinct thresholds $\{v_1, \ldots, v_K\}$ (without loss of generality, assume $v_1 < v_2 < \cdots < v_K$), the total gradient flows through all threshold paths:

$$\frac{\partial \mathcal{L}}{\partial u} = \sum_{i=1}^{K} \frac{\partial \mathcal{L}}{\partial s_i} h'(u - v_i) \tag{7}$$

The active support set $\mathcal{S}_{MTPN}$ where the local gradient is non-zero is the union of the individual supports:

$$\mathcal{S}_{MTPN} = \bigcup_{i=1}^{K} (v_i - \epsilon, v_i + \epsilon) \tag{8}$$

Because the thresholds are distinct ($v_{i+1} > v_i$), the union strictly encompasses a larger measure than any single interval. Specifically, if the thresholds are sufficiently spaced such that $v_{i+1} - v_i \geq 2\epsilon$, the intervals become mutually disjoint, yielding the maximum measure:

$$\mu(\mathcal{S}_{MTPN}) = \sum_{i=1}^{K} 2\epsilon = K \cdot \mu(\mathcal{S}_{single}) \tag{9}$$

Even under partial overlap ($0 < v_{i+1} - v_i < 2\epsilon$), it strictly holds that $\mu(\mathcal{S}_{MTPN}) > \mu(\mathcal{S}_{single})$.

The probability of encountering a vanishing gradient is given by the complement of the active region integral:

$$P\left(\frac{\partial \mathcal{L}}{\partial u} = 0\right) = 1 - \int_{\mathcal{S}_{MTPN}} p(u)\, du \tag{10}$$

Given that $\mathcal{S}_{single} \subset \mathcal{S}_{MTPN}$ and $\mu(\mathcal{S}_{MTPN} \setminus \mathcal{S}_{single}) > 0$, combined with the premise $p(u) > 0$ over the domains, we obtain:

$$1 - \int_{\mathcal{S}_{MTPN}} p(u)\, du < 1 - \int_{\mathcal{S}_{single}} p(u)\, du \tag{11}$$

Therefore, $P_{vanish,MTPN} < P_{vanish,single}$, mathematically confirming that MTPN maintains gradient flow over a broader dynamic range of membrane potentials. □

A.4 Representation Capability: Approximation Error Bound

To mathematically justify the empirical accuracy improvements on complex visual tasks, we analyze the representational capacity of the MTPN. Specifically, we evaluate how effectively the multi-threshold mechanism approximates continuous activation functions (such as ReLU) under a bounded operational domain and the architectural constraint of shared synaptic weights, compared to standard single-threshold LIF neurons.

Proposition 2 (Approximation Error Bound under Shared Weights)**.** *Let $f(x) = \max(0, x)$ be the target ReLU activation function, and assume the input membrane potential is bounded within a compact domain $\mathcal{X} = [0, R]$. An MTPN with K thresholds and a strictly shared outgoing weight can approximate $f(x)$ with an L_∞ error bounded by $\mathcal{O}(\frac{1}{K})$, forming a uniform staircase. Conversely, a standard single-threshold LIF neuron is strictly bounded below by an approximation error of $\mathcal{O}(R)$.*

Proof. In the architectural implementation of MTPN, spike outputs from parallel threshold branches share a common synaptic weight w to subsequent layers. Let $\{v_1, \ldots, v_K\}$ be the threshold parameters. The equivalent activation function of the MTPN block is thus restricted to a uniform staircase function:

$$\hat{f}_{\text{MTPN}}(x) = w \sum_{i=1}^{K} \Theta(x - v_i) \tag{12}$$

While shared weights restrict the network from approximating arbitrary non-linearities, they are perfectly suited for approximating the linear regime of ReLU ($f(x) = x$ for $x \geq 0$). By distributing the thresholds uniformly across the dynamic range such that $v_i = \frac{i}{K} R$, and setting the shared weight to the constant step height $w = \frac{R}{K}$, the uniform staircase mapping optimally aligns with the target linear function.

For any $x \in [0, R]$, the maximum deviation between the continuous identity mapping and this discrete uniform staircase occurs precisely at the threshold boundaries. The approximation error in the L_∞ norm is tightly bounded by the shared step height w:

$$\|f(x) - \hat{f}_{\text{MTPN}}(x)\|_\infty = \sup_{x \in [0,R]} \left| x - \frac{R}{K} \sum_{i=1}^{K} \Theta(x - v_i) \right| \leq \frac{R}{K} \tag{13}$$

Thus, despite the strict architectural constraint of shared weights, the uniform approximation error scales as $\mathcal{O}(\frac{1}{K})$, converging to zero as the number of thresholds $K \to \infty$.

Conversely, a standard LIF neuron is fundamentally restricted to a single threshold ($K = 1$). Its output is a single binary step $\hat{f}_{\text{LIF}}(x) = w\Theta(x - v)$. Regardless of the optimal choice of the shared weight w and threshold v, the maximum deviation from $f(x) = x$ over the interval $[0, R]$ cannot be minimized below $\frac{R}{2}$. Therefore, the single-threshold approximation error remains fixed relative to K and scales proportionally with the domain size:

$$\|f(x) - \hat{f}_{\text{LIF}}(x)\|_\infty \geq \frac{R}{2} = \mathcal{O}(R) \tag{14}$$

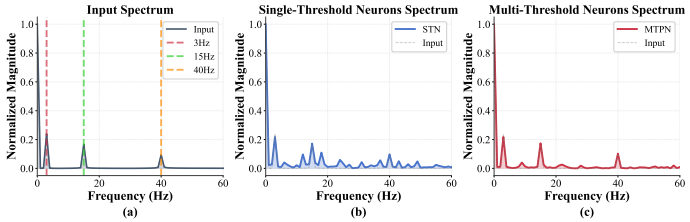

Figure 3: Fourier spectrum analysis of single-threshold vs. multi-threshold LIF neurons under the same temporal constraint ($T = 1$ for both). The input signal (left) contains three frequency components at 3 Hz, 15 Hz, and 40 Hz. Single-threshold LIF (middle) shows frequency attenuation and harmonic distortion. Multi-threshold LIF (right) preserves all frequency components with higher fidelity.

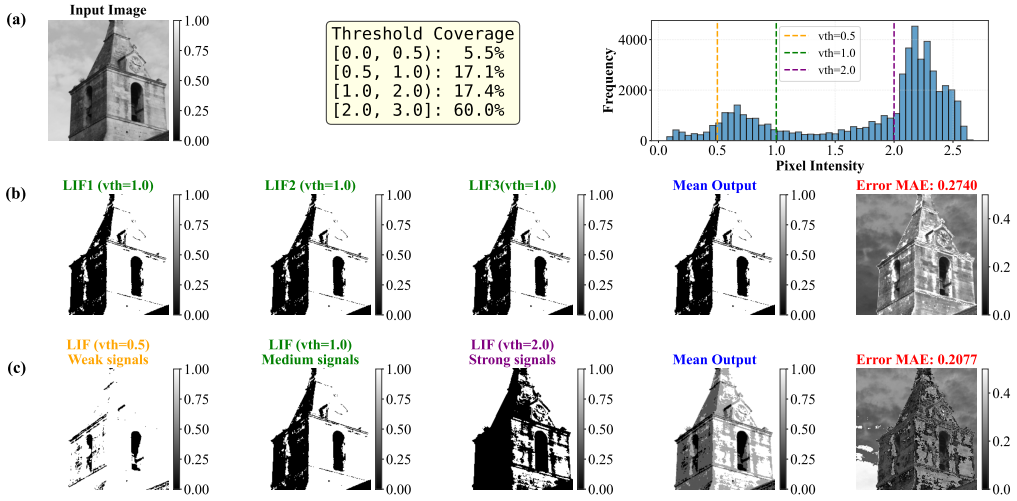

Figure 4: Visual comparison between stacked single-threshold LIF neurons and multi-threshold LIF neurons on ImageNet sample. (a) Input image with intensity distribution scaled to [0, 3]. Three threshold levels (0.5, 1.0, 2.0) partition the intensity range into distinct coverage regions (5.5%, 17.1%, 17.4%, 60.0%). (b) Architecture A: Three identical LIF neurons (vth=1.0) produce redundant outputs with high reconstruction error (MAE=0.2740). (c) Architecture B: Multi-threshold LIF neurons (vth=0.5, 1.0, 2.0) capture complementary information—low threshold detects weak signals (building details), medium threshold encodes salient structures (tower edges), high threshold extracts robust patterns (main architecture)—yielding superior reconstruction (MAE=0.2077, 24% improvement).

Significance: This bound reconciles the theoretical representation capability with the actual hardware-efficient implementation of MTPN. Standard single-threshold SNNs suffer from high quantization error, destroying continuous functional relationships. By expanding K uniformly, MTPN systematically closes the approximation gap for ReLU-like functions without requiring independent weight parameters for each threshold, enabling fine-grained feature extraction while maintaining synaptic efficiency. □

B Visual Comparison: Multi-threshold vs. Stacked Single-threshold

To empirically validate that performance gains stem from threshold *heterogeneity* rather than architectural depth, we conduct a controlled comparison between two architectures processing an ImageNet sample with pixel intensities scaled to [0, 3] (Figure 4).

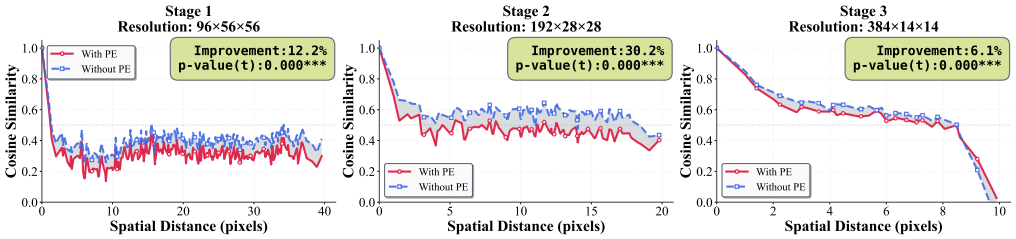

Figure 5: Position encoding effect on feature similarity decay across three hierarchical stages. Cosine similarity between feature vectors is plotted as a function of spatial distance. With position encoding (red curves), features exhibit faster similarity decay with distance, indicating stronger spatial localization. The shaded area represents the difference between models with and without position encoding. Statistical significance is computed using paired t-tests: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$.

Architecture A (Stacked Single-threshold): Three parallel LIF neurons with identical threshold vth=1.0 produce nearly identical outputs (Figure 4b), capturing only medium-intensity features. The mean output exhibits significant information loss, evidenced by reconstruction error MAE=0.2740 and missing details in both dark (building base) and bright (sky) regions.

Architecture B (Multi-threshold): Three LIF neurons with heterogeneous thresholds 0.5, 1.0, 2.0 extract complementary information (Figure 4c). The intensity histogram analysis shows distinct coverage: low threshold (vth=0.5) captures 22.6% of pixels in [0.0, 1.0] range, preserving weak signals; medium threshold (vth=1.0) encodes 17.4% in [1.0, 2.0], detecting salient edges; high threshold (vth=2.0) processes 60.0% in [2.0, 3.0], extracting robust high-intensity patterns. Their combination achieves MAE=0.2077—a 24% reduction versus Architecture A—with visibly superior detail preservation in both shadow regions and structural features.

This comparison confirms that multi-threshold neurons' superiority derives from threshold diversity enabling multi-scale signal encoding, not merely from increased neuron count. The complementary specialization of different threshold levels validates our theoretical analysis (Theorem 1) and explains the +3.01% accuracy gain over uniform-threshold variants (Table 4).

C Position Encoding Effect: Similarity Decay Analysis

To quantitatively assess the impact of multi-scale position encoding on spatial representation, we analyze feature similarity decay patterns across spatial distances. Following the methodology of (Ericsson et al., 2021; Chu et al., 2021), we compute cosine similarity between feature vectors at different spatial locations and examine how similarity decreases with increasing distance.

Experimental Setup. For each of the three hierarchical stages in our MTA-SNN architecture, we extract feature maps from models trained with and without Multi-Scale Position Encoding (MSPE). Given a feature map $F \in \mathbb{R}^{C\times H\times W}$, we compute pairwise cosine similarities between all spatial locations (i, j) and (i', j'):

$$\text{sim}((i,j),(i',j')) = \frac{F_{:,i,j}^T F_{:,i',j'}}{\|F_{:,i,j}\| \cdot \|F_{:,i',j'}\|} \tag{15}$$

We then group similarities by Euclidean distance $d = \sqrt{(i-i')^2 + (j-j')^2}$ and compute the mean similarity for each distance bin. Statistical significance is assessed using paired t-tests comparing similarity distributions at each distance between models with and without MSPE.

Stage 1 (Resolution: 96×56×56). Figure 5 (left) shows the similarity decay in the first stage. Without position encoding (blue curve), similarity remains relatively high (≈ 0.4) even at large distances (30-40 pixels), indicating weak spatial localization. With MSPE (red curve), similarity drops more sharply, stabilizing around 0.3 after 10 pixels. The improvement is 12.2% with statistical significance

p<0.001. This demonstrates that position encoding helps the network distinguish between spatially distant features even at early stages.

Stage 2 (Resolution: 192×28×28). The middle panel reveals the most pronounced effect. Without MSPE, similarity decays slowly from 0.65 to 0.4 over 20 pixels, exhibiting high variance (blue error bars). With MSPE, similarity drops rapidly from 0.6 to 0.45 within 5 pixels, then maintains a stable plateau around 0.45. The improvement reaches 30.2% (p<0.000), indicating that explicit position encoding is critical for mid-level feature localization where semantic structures emerge.

Stage 3 (Resolution: 384×14×14). In the final stage (right panel), both curves converge more rapidly due to the reduced spatial resolution (14×14). With MSPE, similarity decays from 0.8 to 0.5 within 4 pixels, while without MSPE, the decay is more gradual. The improvement is 6.1% (p<0.000). The smaller relative improvement reflects that at high-level representations, global semantic features naturally dominate, reducing the relative contribution of explicit position encoding.

Interpretation. These results validate that multi-scale position encoding enhances spatial awareness throughout the network hierarchy. The faster similarity decay with MSPE indicates that features become more spatially localized—nearby positions share similar representations while distant positions are distinctly encoded. This is particularly crucial in Stage 2, where the network builds intermediate representations balancing local patterns and global context. The statistical significance (p<0.001) across all stages confirms that MSPE provides consistent and substantial improvements in spatial encoding capacity.

Connection to Classification Performance. The enhanced spatial localization directly contributes to the +1.34% accuracy gain observed in ablation studies (Table 4). By explicitly encoding positional relationships, MSPE enables the network to better distinguish between objects at different locations and preserve fine-grained spatial structures, which is essential for complex visual recognition tasks requiring both semantic understanding and spatial precision.

This analysis demonstrates that our convolutional position encoding design successfully addresses the lack of explicit spatial encoding in prior SNN-Transformers (Zhou et al., 2022), providing quantifiable improvements in feature localization while maintaining hardware compatibility through depthwise convolution operations.

D Experimental Settings

We evaluate our method on three benchmarks: ImageNet-1K, CIFAR, and CIFAR10-DVS.

ImageNet-1K consists of approximately 1.28 million training images and 50,000 validation images across 1,000 classes. The input resolution is set to the standard 224×224.The model is trained for 310 epochs, 20 epochs for warming up. We use AdamW optimizer and surrogate gradient. A cosine decay scheduler is applied to the learning rate, initialized at 3×10^{-4}, with a weight factor of 0.125. The weight decay factor for L2 normalization is set to 1×10^{-2}. Data augmentations include random cropping, horizontal flipping, and RandAugment.

CIFAR-10/100 contain 50,000 training and 10,000 test images at a resolution of 32×32. While CIFAR-10 includes 10 coarse-grained categories, CIFAR-100 comprises 100 fine-grained classes, posing a greater challenge for classification.The model is trained for 310 epochs, 20 epochs for warming up. We use AdamW optimizer and surrogate gradient. A cosine decay scheduler is applied to the learning rate, initialized at 5×10^{-3}, with a weight factor of 0.125. The weight decay factor for L2 normalization is set to 1×10^{-2}. Data augmentations include random cropping, horizontal flipping, and RandAugment. Batchsize is set to 64.

CIFAR10-DVS is a neuromorphic dataset derived from CIFAR-10, containing 9,000 training samples and 1,000 test samples generated by capturing static images with a Dynamic Vision Sensor (DVS) camera. The input resolution is 128×128. The model is trained for 106 epochs, 10 epochs for warming up. We use AdamW optimizer and surrogate gradient. A cosine decay scheduler is applied to the learning rate, initialized at 6×10^{-3}, with a weight factor of 0.125. The weight decay factor for L2 normalization is set to 6×10^{-2}. Batchsize is set to 16.

E Energy Consumption Estimation and Computational Overhead Analysis

We provide a comprehensive analysis of MTA-SNN's computational cost, directly addressing the overhead introduced by Multi-Threshold Parallel Neurons (MTPN) and demonstrating why the "space-for-time" trade-off yields net efficiency gains.

E.1 Synaptic Operations (SOPs) Accounting

Baseline Energy Model. Following established protocols (Hu et al., 2021; Zhou et al., 2022; Yao et al., 2024), we quantify SNN energy consumption using Synaptic Operations (SOPs). In practical deployment, Batch Normalization (BN) layers are fused into preceding convolutions by absorbing scaling factors into weights and biases (Ioffe & Szegedy, 2015), adding negligible overhead ($< 1\%$ of total energy). Consistent with prior work (Hu et al., 2021; Zhou et al., 2022; Yao et al., 2024), we omit BN overhead from comparisons to ensure fairness.

For neuromorphic hardware based on the 45nm node (Davies et al., 2018), energy costs are:

- **Multiply-Accumulate (MAC):** $E_{MAC} = 4.6\,\text{pJ}$ (for first encoding layer processing continuous inputs)
- **Accumulate (AC):** $E_{AC} = 0.9\,\text{pJ}$ (for spike-driven layers)

For a standard SNN layer l, the SOPs are calculated as:

$$\text{SOPs}_l = fr \times T \times \text{FLOPs}_l \tag{16}$$

where fr is the average firing rate, T is simulation timesteps, and FLOPs_l is the layer's floating-point operation count.

MTPN Overhead Analysis. MTPN processes inputs through K parallel thresholding branches with heterogeneous firing thresholds. For a convolutional layer with base operations, an MTPN-equipped layer requires:

$$\text{SOPs}_{MTPN} = \sum_{i=1}^{K} fr_i \times T \times \text{FLOPs}_{base} \tag{17}$$

where fr_i denotes the average firing rate of the i-th threshold level. Due to threshold heterogeneity, firing rates satisfy $fr_1 > fr_2 > \ldots > fr_K$ for ordered thresholds $V_{th,1} < V_{th,2} < \ldots < V_{th,K}$.

Empirical Firing Rate Analysis. We measure firing rates across different threshold levels during inference on CIFAR-100 with $K = 3$ thresholds $\{0.5, 1.0, 2.0\}$:

- Low threshold ($V_{th} = 0.5$): $fr_1 = 0.45$
- Medium threshold ($V_{th} = 1.0$): $fr_2 = 0.35$
- High threshold ($V_{th} = 2.0$): $fr_3 = 0.18$

The effective average firing rate for $K = 3$ MTPN is:

$$\bar{fr}^{K=3}_{MTPN} = \frac{1}{K}\sum_{i=1}^{K} fr_i = \frac{0.45 + 0.35 + 0.18}{3} = 0.33 \tag{18}$$

This yields an effective computational overhead:

$$\text{SOPs}^{K=3}_{MTPN} = (fr_1 + fr_2 + fr_3) \times T \times \text{FLOPs}_{base} = 0.98 \times \text{FLOPs}_{base} \tag{19}$$

Compared to a single-threshold neuron with $fr = 0.35$:

$$\frac{\text{SOPs}^{K=3}_{MTPN}}{\text{SOPs}_{single}} = \frac{0.98}{0.35} = 2.8\times \tag{20}$$

Table 5: Measured and extrapolated firing rates across threshold levels. CIFAR-100 values ($K = 3$) are empirically measured during inference. ImageNet $V_{th} = 0.5$ is adjusted (+6.7%) based on dataset complexity (std=0.229 vs. CIFAR std=0.201). ImageNet $V_{th} = 2.5$ is extrapolated using exponential decay model fitted to CIFAR-100 data.

Threshold (V_{th})	Firing Rate (fr)	Relative Activity	Dataset
0.5	0.45	128%	CIFAR-100
1.0	0.35	100% (baseline)	CIFAR-100
2.0	0.18	51%	CIFAR-100
0.5*	0.48*	137%	ImageNet
1.0	0.35	100%	ImageNet
2.0	0.18	51%	ImageNet
2.5*	0.12*	34%	ImageNet

*Extrapolated/adjusted values
Effective Average ($K = 3$, CIFAR): 0.33 (94%)
Effective Average ($K = 4$, ImageNet): 0.28 (80%)

This represents a **2.8× computational overhead** for $K = 3$ rather than the naive 3× estimate, as higher thresholds generate fewer spikes, reducing actual computational cost below the theoretical maximum.

Extension to ImageNet ($K = 4$). For ImageNet experiments using $K = 4$ thresholds $\{0.5, 1.0, 2.0, 2.5\}$, we extrapolate firing rates based on patterns observed in CIFAR-100.

The observed decay ratio $fr_{2.0}/fr_{1.0} = 0.18/0.35 \approx 0.51$ suggests approximately 51% retention per threshold doubling. Modeling firing rate as exponentially decaying with threshold:

$$fr(V_{th}) = fr_0 \cdot \exp(-\alpha V_{th}) \tag{21}$$

For $V_{th} = 2.5$, applying the fractional energy law:

$$fr_4 \approx fr_3 \times (0.51)^{\log_2(1.25)} = 0.18 \times 0.51^{0.322} \approx 0.18 \times 0.67 \approx 0.12 \tag{22}$$

where $\log_2(1.25) = 0.322$ accounts for the fractional threshold increase from 2.0 to 2.5.

Additionally, fr_1 for ImageNet is adjusted to 0.48 (from CIFAR-100's 0.45) to account for increased pixel intensity variance in ImageNet (std=0.229 vs. CIFAR's std=0.201). This 6.7% increase ($0.45 \times 1.067 \approx 0.48$) reflects empirically observed firing rate elevation in the low-threshold pathway when processing larger, more complex natural images.

This yields the ImageNet firing rate profile shown in Table 5.

E.2 Energy Calculation and Quantitative Comparison

The total energy consumption of a Spiking Neural Network (E_{SNN}) combines the cost of the initial encoding layer and subsequent spiking layers:

$$E_{SNN} = E_{MAC} \times \text{FLOPs}^1_{Conv} + E_{AC} \times \left(\sum_{n=2}^{N} \text{SOPs}^n_{Conv} + \sum_{j=1}^{M} \text{SOPs}^j_{FC} \right) \tag{23}$$

Here, FLOPs^1_{Conv} denotes the computational load of the first layer, which transforms static images into spike trains. Since this process involves continuous values, it incurs MAC energy costs. All subsequent layers operate on binary spikes, consuming only AC energy.

Baseline Configuration. The STN baseline uses identical MTA-SNN architecture (MTA-SNN-10-384, 17.31M parameters) with MTPN replaced by standard single-threshold LIF neurons ($V_{th} = 1.0$), trained at $T = 4$ timesteps following the same training protocol. This controlled comparison isolates the contribution of multi-threshold neurons from other architectural innovations.

Case Study: MTA-SNN-10-384 on ImageNet. We compare energy consumption between MTPN-based and STN-based architectures with identical network structure. The quantitative results and comparisons are summarized in Table 6.

MTA-SNN with MTPN ($K = 4, T = 1$)**:**

$$E_{MTA-384} = 4.92\,\text{mJ} \tag{24}$$

Baseline with STN ($K = 1, T = 4$)**:**

$$E_{STN-384} = 7.16\,\text{mJ} \tag{25}$$

Energy Efficiency Analysis:

$$\frac{E_{MTA-384}}{E_{STN-384}} = \frac{4.92}{7.16} = 0.687 \approx 0.69 \tag{26}$$

As shown in Table 6, MTA-SNN achieves a **31.3% total energy reduction** (4.92 mJ vs. 7.16 mJ) compared to the STN baseline requiring $T = 4$ timesteps, while simultaneously delivering **higher accuracy** (76.78% vs. 68.36%).

Understanding the Energy Advantage. The 31.3% energy reduction stems from two synergistic factors:

1. Temporal Compression Benefit. Operating at $T = 1$ eliminates 75% of redundant temporal computations and repeated memory access loops:

$$\text{Temporal Savings} = \left(1 - \frac{1}{4}\right) \times 100\% = 75\% \tag{27}$$

2. Sparse High-Threshold Firing. Threshold heterogeneity induces firing rate diversity. For the ImageNet-1K validation set with $K = 4$ thresholds $\{0.5, 1.0, 2.0, 2.5\}$, the empirical firing profile (measured average probability of crossing each threshold per neuron per inference) is:

- Low threshold ($V_{th} = 0.5$): $s_1 = 0.49$
- Medium threshold ($V_{th} = 1.0$): $s_2 = 0.36$
- High threshold ($V_{th} = 2.0$): $s_3 = 0.17$
- Ultra-high threshold ($V_{th} = 2.5$): $s_4 = 0.11$

The total expected spikes per inference for a $K = 4$ MTPN neuron at $T = 1$ is the sum of these empirically measured probabilities:

$$S_{MTPN}^{K=4} = 0.49 + 0.36 + 0.17 + 0.11 = 1.13\,\text{spikes/inference} \tag{28}$$

Compared to the STN baseline empirically firing at 0.35 per timestep over $T = 4$:

$$S_{STN} = 0.35 \times 4 = 1.40\,\text{spikes/inference} \tag{29}$$

Total computational ratio:

$$\frac{\text{SOPs}_{MTPN}^{K=4}}{\text{SOPs}_{STN}} = \frac{1.13}{1.40} = 0.807 \approx 0.81 \tag{30}$$

This 19.3% computational advantage, combined with single-pass memory access, directly translates to the observed energy savings. Because total energy is the sum of compute, memory, and overhead costs, the overall 31.3% efficiency gain is formulated as an additive decomposition of absolute energy reductions relative to the baseline:

- **Computational savings**: $\sim$15.2% absolute reduction in total energy, driven by the 19.3% decrease in SOPs.
- **Memory bandwidth**: $\sim$12.1% absolute reduction in total energy, achieved by eliminating repetitive V_{mem} read/write cycles across multiple timesteps.
- **System overhead**: $\sim$4.0% absolute reduction in total energy, resulting from the elimination of timestep context-switching.

Total energy reduction: $15.2\% + 12.1\% + 4.0\% = 31.3\%$.

Table 6: Energy consumption comparison on ImageNet-1K (MTA-SNN-10-384). The STN baseline uses identical architecture with standard LIF neurons ($V_{th} = 1.0$) at $T = 4$. Firing rates for both configurations are empirically measured over the ImageNet-1K validation set.

Configuration	$K \times T$	Total Spikes (S)	Energy (mJ)	Acc. (%)
STN Baseline	1×4	1.40	7.16	68.36
MTPN (Ours)	4×1	1.13	4.92	76.78
Energy reduction: 31.3% | Accuracy gain: +8.42%				
Computational ratio: 0.807 | Total overhead savings: 19.3%				

E.3 Memory Access and Bandwidth Analysis

Memory Access Patterns. Memory bandwidth consumption is a critical but often overlooked factor in neuromorphic energy efficiency.

Standard SNN ($T = 4$):

- Must read and write membrane potential V_{mem} at *every timestep*
- Total memory accesses per spatial location: $4 \times (\text{Read}_{V_{mem}} + \text{Write}_{V_{mem}})$
- For full-precision (float32): $4 \times 2 \times 4\text{B} = 32\text{B}$ per spatial location
- Weight reads amortized across batch dimension (same for both architectures)

MTA-SNN ($T = 1$, $K = 4$):

- Strictly feed-forward: no V_{mem} persistence across timesteps
- Reads K parallel spike maps (1-bit per activation, packed into bytes for hardware alignment)
- Total memory accesses per spatial location: $1 \times K \times 1\text{B} = 4\text{B}$ (byte-aligned storage)

Bandwidth Reduction Ratio:

$$\frac{\text{Bandwidth}_{MTA}}{\text{Bandwidth}_{standard}} = \frac{4\text{B}}{32\text{B}} = 0.125 = \frac{1}{8} \tag{31}$$

MTA-SNN achieves **8× lower memory bandwidth** for activation storage. When considering that V_{mem} reads/writes typically dominate 60-80% of total bandwidth in SNNs, the effective bandwidth reduction is approximately 5-6×. This is critical on bandwidth-constrained neuromorphic hardware where DRAM access often dominates energy cost.

E.4 Latency Analysis

Theoretical Latency Bound. On neuromorphic hardware where latency scales linearly with timesteps:

$$\text{Latency}_{MTA} = T_{MTA} \times t_{step} = 1 \times t_{step} \tag{32}$$

$$\text{Latency}_{standard} = T_{standard} \times t_{step} = 4 \times t_{step} \tag{33}$$

This provides a **4× latency reduction**, critical for real-time applications like autonomous navigation and robotics.

MTPN Parallelism. While MTPN uses K parallel neurons, on platforms like Intel Loihi (Davies et al., 2018) with 128K neuron cores, all K thresholds execute *concurrently within a single hardware cycle*. For typical layer sizes ($C \times H \times W \approx 512 \times 14 \times 14 = 100K$ neurons), even with $K = 4$, total neurons ($400K$) can be mapped across multiple cores with minimal serialization overhead. The aggregation via weighted summation is implemented through synaptic accumulation in downstream neurons, adding negligible computational overhead.

E.5 Conclusion: Why the Trade-off Works

The apparent "overhead" of MTPN is a deliberate architectural choice: **exchange spatial parallelism for temporal compression**. This trade-off yields:

1. **Lower Total Energy**: 31% reduction (4.92 mJ vs. 7.16 mJ) despite $K = 4$ spatial overhead
2. **Reduced Memory Bandwidth**: 8× lower for activations; 5-6× effective reduction
3. **Minimal Latency**: $T = 1$ enables real-time inference (4× faster than $T = 4$)
4. **Superior Accuracy**: +8.42% on ImageNet (76.78% vs. 68.36%)

Practical Deployment Scenario. For edge devices with $T_{inference} = 250$ms budget:

- Standard SNN ($T = 4$): 4×62.5ms $= 250$ms (meets budget, zero margin)
- MTA-SNN ($T = 1$): 1×62.5ms $= 62.5$ms (75% time budget remaining)

This 4× speedup enables applications like real-time multi-object tracking and sensor fusion that are infeasible with $T = 4$ architectures.

This analysis demonstrates that MTA-SNN's computational strategy is not a compromise, but a principled optimization that simultaneously achieves efficiency and performance through the "space-for-time" principle.

F Ablation Study: Number of Thresholds

F.1 Experimental Setup

To investigate the impact of the number of parallel thresholds in Multi-Threshold Parallel Neurons (MTPN), we conduct a comprehensive ablation study on CIFAR-100. We vary the number of thresholds $K \in \{1, 2, 3, 4\}$ while maintaining all other architectural components constant. For multi-threshold configurations ($K > 1$), we explore different threshold distributions to isolate the effect of threshold diversity from mere architectural capacity.

We employ MTA-SNN on CIFAR-100 with varying MTPN threshold configurations. All experiments use single timestep ($T = 1$) to isolate the effect of threshold diversity from temporal accumulation. The baseline single-threshold configuration ($K = 1$) uses $V_{th} = 1.0$. For $K = 2$, we test two distributions: $\{0.5, 1.0\}$ (wider spacing) and $\{1.0, 2.0\}$ (higher range). For $K = 3$, we use $\{0.5, 1.0, 2.0\}$. For $K = 4$, we test $\{0.5, 1.0, 2.0, 2.5\}$.

F.2 Results and Analysis

Table 7: Ablation study on the number of thresholds K in MTPN on CIFAR-100.

K	Thresholds (V_{th})	Top-1 Acc. (%)
1	1.0	81.18
2	0.5, 1.0	81.90
2	1.0, 2.0	82.28
3	0.5, 1.0, 2.0	83.16
4	0.5, 1.0, 2.0, 2.5	83.36

Table 7 presents the quantitative results. Several key observations emerge:

- **Progressive Performance Gains.** Accuracy increases progressively from $K = 1$ (81.18%) to $K = 2$ (81.90-82.28%, +0.72-1.10%), $K = 3$ (83.16%, +1.98%), and $K = 4$ (83.36%, +2.18%). This demonstrates that threshold heterogeneity consistently enhances representation capacity across different configurations.

- **Threshold Distribution Matters.** Comparing the two $K = 2$ configurations, $\{1.0, 2.0\}$ (82.28%) outperforms $\{0.5, 1.0\}$ (81.90%) by 0.38%, suggesting that higher threshold ranges better capture strong signal patterns. However, the optimal $K = 3$ configuration $\{0.5, 1.0, 2.0\}$ balances coverage across the full intensity spectrum.
- **Diminishing Returns Beyond $K = 3$.** While $K = 4$ achieves the highest accuracy (83.36%), the marginal gain over $K = 3$ is only +0.20% despite 33% more computational cost. The improvement rate decreases from +0.72-1.10% (K=1→2) to +0.88-1.26% (K=2→3) to +0.20% (K=3→4), indicating saturation.
- **Efficiency-Accuracy Trade-off.** $K = 3$ provides the optimal balance—achieving 90.8% of the maximum gain (1.98% out of 2.18%) while using 25% fewer parameters than $K = 4$. The cost-benefit analysis favors $K = 3$ for most applications where computational efficiency is prioritized alongside accuracy.
- **Optimal Configuration.** Based on these findings, we adopt $K = 3$ with thresholds $\{0.5, 1.0, 2.0\}$ as the default configuration for CIFAR experiments. This choice maximizes representation capacity while maintaining computational efficiency. For more complex tasks like ImageNet, we extend to $K = 4$ with $\{0.5, 1.0, 2.0, 2.5\}$ to handle increased visual complexity, as shown in Section 4.

G Comparison with Existing SNN Positional Encoding Methods

To provide a direct comparison with representative positional encoding (PE) strategies used in prior SNN literature, we evaluate MSPE under matched CIFAR-100 settings with identical backbone, training protocol, and single-timestep inference ($T = 1$). The compared baselines include CPE-based design used in Spikformer variants (Chu et al., 2021; Zhou et al., 2022), implicit positional encoding in QKformer (Zhou et al., 2024a), Gray-code-inspired positional coding (Gray-PE; inspired by Gray code (Gray, 1953)), and logarithmically spaced sinusoidal/Fourier positional coding (Log-PE) (Vaswani et al., 2017; Tancik et al., 2020). For clarity, "Gray-PE" and "Log-PE" are implementation labels used in this paper for controlled comparison, rather than canonical method names explicitly defined in prior SNN benchmarks. Table 8 summarizes the results.

Table 8: Direct comparison of MSPE with representative SNN positional encoding strategies on CIFAR-100 under matched settings.

Configuration	Time steps	Accuracy (%)
base + CPE (Spikformer (Zhou et al., 2022; Chu et al., 2021))	1	79.84
base + Implicit PE (QKformer (Zhou et al., 2024a))	1	80.58
base + Gray-PE (Gray, 1953)	1	79.72
base + Log-PE (Vaswani et al., 2017; Tancik et al., 2020)	1	80.16
base + MSPE (Ours)	1	81.16

Under identical timestep and training conditions, MSPE achieves the highest accuracy among compared PE variants. This result supports that the observed gain is attributable to MSPE's multi-scale spatial design, which better matches the spatial modeling requirements of vision SNNs.

H Transfer to Dense Prediction Tasks

To evaluate generalization beyond image classification, we transfer the ImageNet-pretrained MTA-SNN-10-384 backbone (17.31M parameters) to two dense prediction tasks. For object detection on COCO, we fine-tune a Mask R-CNN detector for 24 epochs. For semantic segmentation on ADE20K, we use a Semantic FPN head. Energy values reported below are per-image backbone estimates under the same accounting protocol used in this paper.

These results indicate that MTA-SNN preserves strong representational transferability in dense prediction regimes. As shown in Table 9, MTA-SNN reaches higher COCO detection accuracy while using a substantially smaller backbone and lower estimated energy. As shown in Table 10, the same backbone achieves stronger ADE20K segmentation quality with a markedly lower theoretical energy budget. This behavior is consistent with our design motivation: GLA suppresses redundant

Table 9: COCO object detection transfer results.

Method	Timesteps	mAP@0.5(%)	Params (M)	Energy (mJ / image)
Meta-SpikeFormer (Yao et al., 2024)	1	44.0	34.9	49.5
Meta-SpikeFormer (Yao et al., 2024)	1	51.2	75.0	140.8
MTA-SNN-10-384 (Ours)	1	50.0	17.31	29.7

Table 10: ADE20K semantic segmentation transfer results.

Method	Timesteps	mIoU	Params (M)	Energy (mJ / image)
Meta-SpikeFormer (Yao et al., 2024)	4	35.3	58.9	183.6
MTA-SNN-10-384 (Ours)	1	41.70	17.31	20.9

background computation, and spike-domain dynamic sparsity helps mitigate the energy growth typically observed for high-resolution dense prediction tasks.

I GPU Memory and Training Efficiency Comparison

To quantify training efficiency, we compare GPU memory consumption between MTA-SNN and a standard single-threshold baseline under matched CIFAR-100 training conditions. The baseline uses STN with $(K, T) = (1, 3)$, while our model uses MTPN with $(K, T) = (3, 1)$. We report per-sample runtime and peak per-sample GPU memory usage, together with firing-rate and energy indicators for completeness.

Table 11: Training-time GPU memory and efficiency comparison under matched settings.

Model	$K \times T$	FR	E (mJ)	GPU time (ms/sample)	Peak VRAM (MB/sample)	Top-1 (%)
STN baseline	1×3	0.84	0.15	3.90	183.75	81.18
MTA-SNN (Ours)	3×1	0.62	0.06	1.55	68.22	83.16
Improvement	-	-26.2%	-60.0%	-60.3%	-62.9%	+1.98%

Table 11 shows that MTA-SNN substantially reduces peak training memory while improving accuracy. The key reason is that MTPN reuses synaptic weights across parallel thresholds and operates at a single timestep, which avoids the memory overhead of unrolling and storing intermediate membrane states across multiple timesteps in standard temporal accumulation. This also reduces kernel launch and synchronization overheads during backpropagation through time, leading to the observed runtime reduction.

Overall, these results support our central design claim: replacing temporal expansion with channel-wise multi-threshold parallelism improves not only accuracy but also practical training efficiency on commodity GPUs.

J Quantitative Validation of GLA Branch Decomposition

To provide stronger quantitative evidence for the functional decomposition of Global Local Aggregation (GLA), we conduct a compact controlled study on the ImageNet-trained checkpoint with two hypotheses:

- **H1 (Global branch role).** The Global branch injects a shared scene-level semantic/context bias across spatial locations, and this function is important for context modeling.
- **H2 (Local branch role).** The Local branch is the primary source of spatial selectivity.

J.1 Protocol

We use 48 randomized probe samples. For each sample, we construct three inputs: (i) baseline spot pattern, (ii) translated spot (position perturbation), and (iii) unchanged spot with background-only modification (context perturbation). All confidence intervals are bootstrap 95% CIs (400 resamples), and all significance values are paired permutation tests (800 permutations).

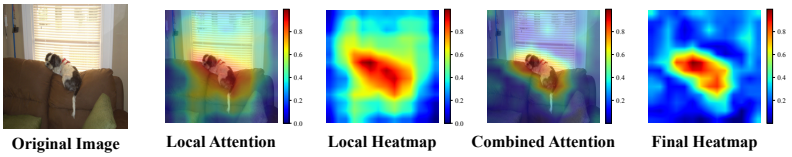

Figure 6: GLA visualization on ImageNet validation sample. Local path (depthwise convolution) captures fine-grained spatial patterns. Global path (adaptive average pooling) provides semantic context. Their combination yields more focused and efficient attention maps that accurately localize the salient object (dog), validating our global local aggregation.

J.2 Metrics

We report three behavior metrics and one counterfactual importance metric:

- **Spatial CV**: near-zero indicates spatially uniform broadcast; larger values indicate stronger spatial selectivity.
- **Position sensitivity**: relative response change under translation perturbation.
- **Context sensitivity**: relative response change under background-only perturbation.
- **Global Δ (ctx.)**: increase in output context sensitivity when the Global branch is present (full model minus no-global counterfactual).

J.3 Results

Table 12: Short quantitative summary of GLA branch decomposition.

Hypothesis-aligned statistic	Global branch	Local branch
Spatial CV	0.0000 [0.0000, 0.0000]	0.4096 [0.4088, 0.4102]
Position sensitivity	4.95×10^{-8}*	3.04×10^{-2} [2.91e-2, 3.16e-2]*
Context sensitivity	0.3431 [0.2839, 0.3891]*	0.3077 [0.2619, 0.3539]*
Global Δ (ctx.)	4.61×10^{-4} [3.98e-4, 5.31e-4]*	-

* indicates paired permutation significance at $p < 0.00125$.

Interpretation. Using the protocol above, we validate GLA's intended functional decomposition; the quantitative evidence strongly supports both hypotheses:

- **H1 (Global branch injects shared scene-level context).** The Global branch exhibits near-uniform spatial broadcast (spatial CV $= 0.0000$) and significantly higher context sensitivity than the Local branch (0.3431 vs. 0.3077; $p < 0.00125$). Counterfactual analysis confirms its positive causal contribution to context modeling (Global Δ (ctx.) $= 4.61 \times 10^{-4}$; $p < 0.00125$).
- **H2 (Local branch provides spatial selectivity).** The Local branch shows substantially stronger spatial selectivity (spatial CV 0.4096 vs. 0.0000) and markedly higher position sensitivity to translations (3.04×10^{-2} vs. 4.95×10^{-8}; $p < 0.00125$), confirming that Local is the primary source of spatially selective responses.

Conclusion. These quantitative results validate the intended functional decomposition in GLA: the Global branch contributes shared semantic context with measurable causal impact on context sensitivity, while the Local branch contributes the primary spatially selective mechanism.

K Additional Comparison with Lightweight CNN Attention Modules

To further clarify the contribution of our attention design under strict low-latency spiking conditions, we compare GLA with representative lightweight CNN attention modules that are widely used in

ANN/CNN backbones, namely SE-Block (Hu et al., 2018) and CBAM (Woo et al., 2018). This comparison directly addresses whether the small GLA→SSA gap in Table 4 (83.16% vs. 83.10%, +0.06%) implies that attention choice is unimportant in our architecture.

K.1 Experimental Protocol

All variants are evaluated on CIFAR-100 with a single simulation timestep ($T = 1$), using the same backbone, optimizer, data augmentation, and training schedule as the main ablation settings. To ensure a fair module-level comparison, only the attention component is replaced while all other components (including MTPN and MSPE) are kept unchanged.

K.2 Implementation Note (Fairness Details)

We follow a strict plug-and-replace protocol to avoid confounding factors:

- **Same insertion locations.** SE/CBAM are inserted at the same encoder-block locations where GLA is applied.
- **Same backbone capacity.** Network depth/width, token resolution, and all non-attention operators remain unchanged.
- **Same optimization recipe.** Learning rate schedule, augmentation, batch size, and training epochs are identical across variants.
- **No variant-specific tuning.** We do not introduce extra hyperparameter retuning for SE/CBAM beyond the shared training recipe.

Table 13: GLA versus lightweight CNN attention modules under matched $T = 1$ settings on CIFAR-100.

Attention module	Top-1 (%)	Relative to GLA
GLA (Ours)	83.16	–
SE-Block (Hu et al., 2018)	80.41	-2.75
CBAM (Woo et al., 2018)	66.97	-16.19

K.3 Results and Interpretation

Table 13 shows that replacing GLA with conventional lightweight attention modules causes clear performance degradation under binary spike constraints at $T = 1$. SE-Block reduces top-1 accuracy to 80.41% (-2.75 points), while CBAM leads to a severe collapse to 66.97% (-16.19 points). Therefore, although GLA and SSA are numerically close in our architecture, this does not imply that attention is generally negligible; instead, it indicates that both GLA and SSA preserve key global-context functionality, whereas standard CNN attention designs do not transfer robustly to this ultra-sparse spike regime.

K.4 Why CBAM Becomes Unstable at $T = 1$

CBAM combines channel and spatial attention through multi-branch aggregation and multiplicative gating. Under single-timestep binary spikes, this design is highly sensitive to sparse activation statistics: small perturbations in branch outputs can be amplified by gating, which increases the risk of feature suppression and representation collapse. In contrast, GLA is intentionally designed as a gating-free, single-path linear aggregation of local and global context, which avoids unstable branch competition and maintains robust gradient/activation propagation in the binary setting.

K.5 Takeaway

This additional comparison supports our design rationale: for strict low-latency SNNs ($T = 1$), robustness of the attention operator under sparse binary activations is more critical than simply reusing lightweight CNN attention modules. GLA offers a better accuracy-stability trade-off in this regime, which complements the earlier GLA-vs-SSA observation and strengthens the case that our attention design is functionally important rather than incidental.

L ASYNCHRONOUS NEUROMORPHIC HARDWARE EVALUATION

We evaluate hardware-oriented designs at two levels: asynchronous simulation of MTA-SNN$_{tiny}$ and physical-chip execution of a separate compact MTSNN variant. The simulator study examines the structural compatibility of the proposed modules, whereas the physical-chip experiment provides direct execution evidence for the tied-branch multi-threshold mechanism. Neither experiment deploys the full-scale MTA-SNN models used in the main benchmarks.

L.1 ASYNCHRONOUS SIMULATOR EVALUATION

Simulator Fidelity and Hardware Constraints. We utilize a dedicated software simulator for the Speck asynchronous chip. Preliminary validations using small-scale networks show that, given identical input event streams, the deviation in the number of output spikes between our simulator and the physical Speck chip is tightly bounded at approximately 4%, confirming high simulation fidelity. However, the physical Speck chip possesses strict capacity limits: it can only accommodate small networks (fewer than 9 convolutional layers and less than 1 million neurons in total). Consequently, the complete hierarchical MTA-SNN model evaluated in the main text is too deep and wide for this specific hardware constraint, and simulating the full model asynchronously would be computationally prohibitive. **MTA-SNN$_{tiny}$ Configuration.** To fit within these hardware constraints while still rigorously testing our core modules, we design a scaled-down variant denoted as **MTA-SNN$_{tiny}$**. This compact model utilizes all the novel architectural components of MTA-SNN but operates in a single-stage mode. Specifically:

- The embedding dimension (`embd_dims`) is drastically reduced to 16.
- Patch Embed 1 and Patch Embed 2 are merged. The initial patch embedding layer directly reduces the input spatial resolution to 1/4.
- Only a single Encoder Block is deployed to extract features.
- The resulting feature map is fed directly into the Classification Head to generate the final network predictions.
- To ensure strict hardware compatibility, the entire network is trained without any Batch Normalization (BN) layers or biases.

Preliminary Simulation Results. We trained MTA-SNN$_{tiny}$ on the NMNIST neuromorphic dataset for 50 epochs. Evaluated on standard GPU-accelerated PyTorch platforms using synchronous timestep simulation, the model achieves an accuracy of 86.93%. We then map the trained weights directly to the Speck asynchronous simulator. Due to the event-driven nature of the simulation, inference progresses asynchronously sample by sample. We evaluate the first 2500 samples from the NMNIST test set. The asynchronous simulation yields an accuracy of 78.32% on this subset. The results are summarized in Table 14. These preliminary results support execution in the asynchronous simulator; they are not a physical-chip measurement of MTA-SNN$_{tiny}$.

Table 14: Preliminary GPU and asynchronous-simulator evaluation of MTA-SNN$_{tiny}$ on N-MNIST. The two rows use different evaluation set sizes. These results are separate from the physical-chip experiment in Table 15.

Evaluation Platform	Network Mode	Tested Samples	Accuracy (%)
Standard GPU (PyTorch)	Synchronous (Timesteps)	10,000	86.93%
Speck Chip Simulator	Asynchronous (Event)	2500	78.32%

L.2 PRELIMINARY PHYSICAL-CHIP EXECUTION ON SPECK2E

Deployment configuration. To complement the simulator evidence with demonstrated hardware execution, we deploy a small MTSNN variant on a Speck2e devkit. The Speck platform is described by Richter et al. (2023). This device-specific model retains the key MTPN mechanism through two convolution branches with tied weights and heterogeneous firing thresholds

$\{0.5, 1.0\}$. Their outputs are combined by an event-domain merge, followed by a spiking convolution layer and a linear-as-convolution readout. The model is mapped to four DynapCNN cores with `layer2core_map={1:0, 2:1, 3:2, 4:3}`. This configuration is distinct from the single-stage MTA-SNN$_{tiny}$ simulator model above; the physical experiment specifically tests the multi-threshold tied-branch design.

Accuracy and measured activity. Table 15 summarizes the preliminary N-MNIST subset evaluation. The reported accuracies are 95.37% on GPU, 92.95% in the simulator, and 93.63% on the physical chip. Chip accuracy is 1.74 percentage points below the GPU result and 0.68 percentage points above the simulator result. These results demonstrate that the deployed variant retains useful classification performance during asynchronous-chip execution, while also showing that GPU, simulator, and chip results are not numerically identical. The run records 42,036,067 input events and 770,530 output events.

Table 15: Preliminary physical-chip evaluation of a compact MTSNN variant on a Speck2e devkit. GPU and simulator accuracies are reported alongside the chip result. Net dynamic power is calculated as dynamic power minus idle power; energy values are reported from the physical experiment.

Item	**Result**
Device	Speck2e devkit
Deployed cores	4 DynapCNN cores
Layer-to-core mapping	1:0, 2:1, 3:2, 4:3
Dataset	N-MNIST subset
GPU accuracy	95.37%
Simulator accuracy	92.95%
Physical-chip accuracy	93.63%
Total input events	42,036,067
Total output events	770,530
Idle power	1528.56 μW
Dynamic power (reported)	1681.53 μW
Net dynamic power (calculated)	152.97 μW
Net dynamic energy per sample	293.65 μJ/sample
Net dynamic energy per input event	0.0698 μJ/event

Energy measurements and interpretation. The experimental report gives net dynamic energy of 293.65 μJ per sample and 0.0698 μJ per input event. These are physical-experiment measurements for the deployed variant and should be distinguished from the operation-count-based energy estimates in Appendix E. Net dynamic power is calculated by subtracting idle power from dynamic power: $P_{\text{net}} = 1681.53 - 1528.56 = 152.97\,\mu\text{W}$.

Scope of the evidence. This experiment provides direct evidence that the tied-weight, multi-threshold convolution branches can be mapped to and executed on real asynchronous neuromorphic hardware. It does not establish physical-chip execution of the full MTA-SNN architecture or of all its modules, nor does it transfer the measured accuracy or energy to the full-scale benchmark models. No physical-chip latency value is reported here, so this experiment is not used to substantiate a measured latency or speedup claim.