DELTA SNN: LEARNING TIME-VARIANT SYNAPTIC WEIGHTS IN SPIKING NEURAL NETWORKS FOR EVENT VISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs) naturally align with event cameras through sparse, event-driven computation, yet effectively modeling the spatiotemporal dynamics of event streams remains challenging. Typically, SNNs fully share synaptic weights across timesteps to maintain parameter and computational efficiency. However, event distributions and neuronal states evolve over time, causing different timesteps to exhibit heterogeneous features and potentially conflicting optimization demands, while a fixed transformation must serve them all. To address this issue, we propose Delta SNN, a time-variant synaptic framework that relaxes complete temporal weight sharing. Delta SNN augments a shared synaptic backbone with compact temporal modulation, enabling adaptive transformations while retaining cross-time knowledge sharing. We further introduce binary synaptic routing to realize temporal modulation with sparse, addition-based inference. Theoretically, we show that time-variant weights substantially enhance the expressive capacity of SNNs. Experiments across three challenging event-vision tasks consistently improve strong baselines with only marginal overhead.

1 INTRODUCTION

Event cameras asynchronously capture changes in pixel brightness, producing sparse event streams with high temporal resolution, reduced motion blur, and high dynamic range (Gallego et al., 2020). These properties make them well suited to dynamic perception in applications such as autonomous driving and robotics (Gehrig & Scaramuzza, 2024). Spiking neural networks (SNNs) provide a natural computational counterpart: their event-driven communication and intrinsic temporal dynamics align closely with event streams, offering a promising foundation for efficient neuromorphic vision (Roy et al., 2019; Hagenaars et al., 2021).

Despite this compatibility, effectively extracting the rich spatiotemporal structure of event streams remains challenging for SNNs, resulting in a non-trivial performance gap with ANNs (Xu et al., 2025a). ANN-based approaches typically combine attention with recurrent aggregation or hierarchical memory to capture spatial relationships and temporal dependencies (Gehrig & Scaramuzza, 2023; Hamaguchi et al., 2023). Directly mirroring these modules in SNNs, however, would introduce dense and extensive temporal computation, compromising the sparse, addition-based computation that underlies their efficiency (Yao et al., 2023b). Recent SNN studies have sought to narrow this gap through spike-friendly architectures and training strategies for improved spatiotemporal modeling (Fan et al., 2024; Luo et al., 2024; Yao et al., 2025). Yet these efforts largely focus on macro-level designs, while the underlying neuronal dynamics remain relatively rigid and limited in modeling complex spatiotemporal semantics.

Conventional SNN models capture spatiotemporal semantics in event streams through the fundamental neuronal dynamics. Neurons integrate information over time, while synaptic transformations extract spatial features. Conventional SNNs reuse the same synaptic weights across timesteps, providing a parameter-efficient inductive bias (Wu et al., 2018). However, such temporal weight sharing implicitly assumes that features across timesteps follow similar distributions and can be effectively processed by a common transformation. In practice, neuronal context evolves as information accumulates, making features at early and late timesteps increasingly heterogeneous. This shift is particularly pronounced

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.

under highly dynamic event inputs. A fixed synaptic transformation can therefore limit fine-grained discrimination across temporal contexts and force shared weights to reconcile conflicting optimization directions, leading to suboptimal spatiotemporal representations. An intuitive solution is to assign independent synaptic weights to each timestep, but doing so substantially increases parameter and training costs, undermining the efficiency advantages of SNNs.

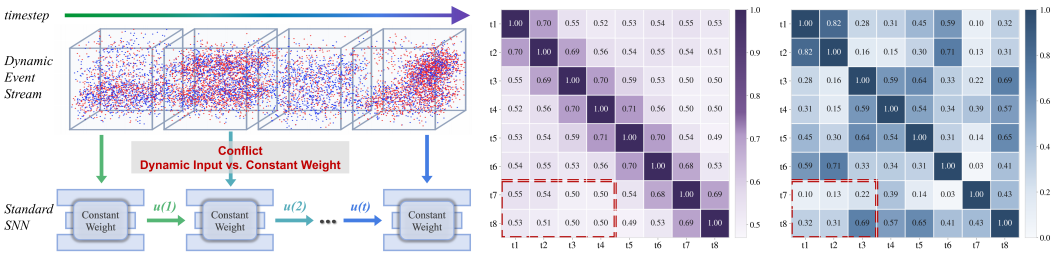
In this work, we explore how to enhance the spatiotemporal modeling capacity of SNNs for event vision through temporal synaptic heterogeneity while retaining their inherent efficiency. We propose Delta SNN, a versatile framework that equips conventional SNNs with time-variant synaptic weights. Rather than assigning independent weights to individual timesteps, Delta SNN retains a shared synaptic backbone and introduces compact temporal modulation to capture timestep-dependent variations. This design preserves knowledge sharing across time while allowing effective synaptic transformations for evolving temporal contexts. We further introduce binary synaptic routing, which realizes temporal modulation through lightweight routing and accumulation instead of dense floating-point operations. Together, these designs enable time-variant synaptic transformations without sacrificing the computational efficiency that makes SNNs attractive. We further provide a theoretical analysis characterizing the representational gains of Delta SNN over conventional SNNs. Experiments across multiple challenging event-vision tasks demonstrate consistent improvements in both performance and efficiency. Our contributions are threefold:

1. We identify temporal weight sharing as a key bottleneck and propose Delta SNN, which introduces time-variant synaptic transformations through compact modulation and binary routing while preserving spike-driven efficiency.
2. We establish theoretical guarantees characterizing the capacity gains of Delta SNN. Our analysis shows that time-variant synaptic transformations expand the expressive capacity needed for fine-grained discrimination under temporally dynamic inputs.
3. We conduct extensive experiments on challenging event-vision tasks, including automotive detection, human activity recognition, and video anomaly detection. Comprehensive studies confirm the effectiveness and efficiency of Delta SNN.

2 Related Works

Spiking Neural Networks. SNNs attract growing attention for energy-efficient neuromorphic computing. Their training methods generally fall into three categories: (1) ANN-to-SNN conversion, (2) unsupervised local learning via Spike-Timing-Dependent Plasticity (STDP), and (3) direct supervised learning with surrogate gradients. ANN-to-SNN conversion (O'Connor et al., 2013; Sengupta et al., 2019; Deng & Gu, 2021) benefits from mature ANN models, but rate coding often introduces high latency. Temporally weighted dynamics are explored to improve coding efficiency (Kim et al., 2018; Stöckl & Maass, 2021). Unsupervised STDP methods (Caporale & Dan, 2008; Florian, 2007; Rahman & Yusoff, 2025; Shen et al., 2023) use biologically plausible local rules but often face scalability and generalization challenges on complex tasks. Direct supervised learning uses surrogate gradients to enable end-to-end Spatio-Temporal Backpropagation (Wu et al., 2018; Neftci et al., 2019). While these methods achieve competitive performance on static image benchmarks (Zhou et al., 2024; Lei et al., 2025), they adopt time-invariant network weights. None employ time-variant synaptic weights in modern SNNs and optimize them in an end-to-end supervised manner.

Event-based Vision. Event cameras (or Dynamic Vision Sensors) represent a paradigm shift in visual acquisition, operating asynchronously to record pixel-level log-intensity changes rather than synchronized frames (Rebecq et al., 2018; Chakravarthi et al., 2024). This mechanism eliminates redundancy and motion blur, offering microsecond temporal resolution, high dynamic range, and exceptionally low latency (Gallego et al., 2020; Shariff et al., 2024). Such characteristics align inherently with the sparse, asynchronous processing of SNNs, positioning SNNs as a leading candidate for energy-efficient neuromorphic vision. While SNNs have shown promise in tasks like optical flow estimation (Hagenaars et al., 2021) and gesture recognition (Amir et al., 2017), they still lag behind the state-of-the-art ANNs on complex event-based benchmarks (Xu et al., 2025a; Yao et al., 2025). A critical bottleneck is the reliance on time-invariant weights. This simple formulation fails to capture the temporal context shift under event input, leading to suboptimal feature extraction in highly dynamic scenarios (De Tournemire et al., 2020; Wang et al., 2024b).



(a) Processing event stream with standard SNN (b) Temporal feature similarity (c) Temporal gradient similarity

Figure 1: Motivation for time-variant synaptic weights. (a) Event streams and neuronal states evolve over time, yet conventional SNNs use fixed weights. (b) On HAR-DVS ($T = 8$), intermediate features of a Spike-Driven Transformer show low similarity across distant timesteps. Representative low-similarity regions are highlighted by red boxes. (c) Such decoupled features induce low gradient similarity at the same layer, indicating conflicting timestep-wise optimization directions.

3 METHOD

We first revisit the LIF neuron, then introduce *Delta SNN*, a flexible framework that enhances spatiotemporal modeling with minimal computational overhead.

3.1 PRELIMINARY

Leaky Integrate-and-Fire (LIF) neuron (Maass, 1997) is the most widely adopted neuron in SNNs. At each time step t, LIF neuron performs the bio-inspired dynamics defined as:

$$\begin{aligned} s(t) &= H\big(\beta u(t-1) + \mathbf{W}x(t) + b - \vartheta\big), \\ u(t) &= \beta u(t-1) + \mathbf{W}x(t) + b - \vartheta s(t). \end{aligned} \tag{1}$$

Here, $s(t) \in \{0,1\}^N$ and $u(t) \in \mathbb{R}^N$ denote the binary spike output and membrane potential at time t, respectively. The neuronal dynamics are characterized by a three-stage evolution. First, the neuron integrates the input $x(t)$ via the synaptic weight $\mathbf{W}$ and bias b into the leaked prior state $\beta u(t-1)$, where $\beta \in (0,1)$ governs the membrane decay. Second, a spike is triggered through the Heaviside step function $H(\cdot)$ when the accumulated potential surpasses the firing threshold ϑ. Finally, a soft reset mechanism is applied, where the term $-\vartheta s(t)$ subtracts the threshold value to regulate the membrane potential for the subsequent timestep.

While the above mathematical abstraction enables efficient implementation and computation, its reliance on a time-invariant weight matrix $\mathbf{W}$ oversimplifies the dynamically evolving nature of biological synapses (Harris et al., 2002; Jackman et al., 2016). In practice, this limits the model's capacity to handle complex temporal computing tasks, especially under sparse spike-driven computation.

3.2 DELTA SNN

Temporal Rigidity under Weight Sharing. Conventional SNNs reuse the same synaptic weights across timesteps, providing an efficient and parameter-economical inductive bias but limiting temporal adaptability. As illustrated in Figure 1(a), event streams exhibit continuously changing spatial distributions, while the same synaptic transformation is applied throughout the sequence. Beyond this representational mismatch, we further identify *temporal optimization interference*: different temporal contexts may favor distinct parameter updates, forcing shared weights to compromise among them during optimization. From this perspective, an SNN unrolled over T timesteps can be viewed as a hard-parameter-sharing temporal multi-task system. Specifically, we introduce virtual copies $\{W_t\}_{t=1}^T$ of a synaptic weight and write the sequence-level objective as $\mathcal{L}(W_1, \ldots, W_T)$. A conventional SNN imposes the constraint $W_1 = \cdots = W_T = W$, under which

$$\nabla_W \mathcal{L} = \sum_{t=1}^{T} g_t, \qquad g_t = \left.\frac{\partial \mathcal{L}}{\partial W_t}\right|_{W_1=\cdots=W_T=W},$$

where g_t represents the update induced by the temporal context at timestep t. To examine this effect, we measure timestep-wise similarities of intermediate features and weight gradients in a spike-driven

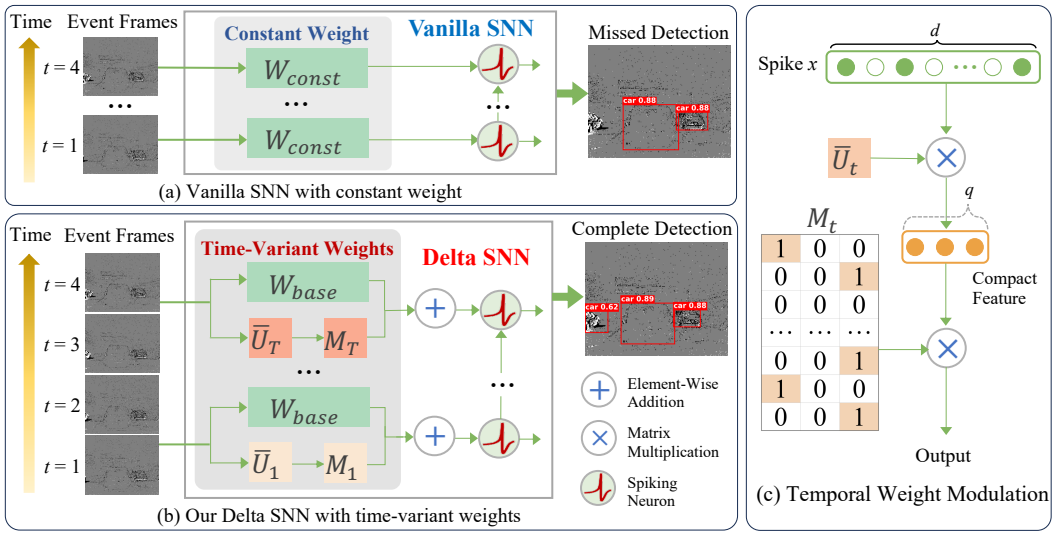


Figure 2: Overview of Delta SNN Framework. (a) Vanilla SNNs are constrained by constant synaptic weights, which limits their ability to capture the complex spatiotemporal dynamics of dynamic event data. (b) The proposed Delta SNN incorporates time-variant weights modulated by learnable matrices, significantly enhancing performance for event vision tasks. (c) The modulation is instantiated through compact temporal weight modulation and binary synaptic routing, enabling adaptive temporal representation while preserving sparse, addition-based inference.

transformer trained on HAR-DVS (Yao et al., 2025; Wang et al., 2024b). As shown in Figure 1(b), representations remain strongly correlated at nearby timesteps but become increasingly decoupled over longer temporal intervals. More importantly, the gradients show markedly lower alignment across timesteps (Figure 1(c)), indicating that temporally heterogeneous representations also induce different optimization demands on the shared weights. When such gradients are weakly aligned, their joint update can interfere, limiting the ability of a single shared transformation to adapt across temporal contexts. This motivates our design for relaxing complete temporal parameter sharing.

Unlocking Time-Variant Connectivity. To address the mismatch between dynamic temporal context and fixed model parameters, we generalize the standard LIF connectivity and propose the Delta LIF neuron, which introduces temporal heterogeneity into synaptic weights. Specifically, the conventional time-invariant synaptic weight is extended to a time-dependent sequence $\{\mathbf{W}_t\}_{t=1}^{T}$. Instead of transforming inputs from all timesteps with a single shared weight, Delta LIF enables timestep-specific synaptic transformations that better accommodate the temporal characteristics of each input. Formally, the refined neuronal dynamics are given by:

$$\begin{aligned} s(t) &= H\big(\beta u(t-1) + \mathbf{W}_t x(t) + b - \vartheta\big), \\ u(t) &= \beta u(t-1) + \mathbf{W}_t x(t) + b - \vartheta s(t). \end{aligned} \tag{2}$$

Delta LIF preserves the computational integrity of standard neuronal dynamics and can be readily integrated into existing LIF-based SNN architectures. Based on this formulation, we construct *Delta SNN*, a time-variant framework for unleashing temporal modeling capacity. However, realizing this potential depends critically on how the time-variant weight $\mathbf{W}_t$ is designed. Specifically, $\mathbf{W}_t$ should be expressive enough to capture temporal dynamics while still preserving the spike-driven computation paradigm. We address this challenge through two effective designs.

Compact Temporal Weight Modulation. A straightforward way to parameterize $\mathbf{W}_t$ is to assign an independent weight matrix to each timestep. However, this fully decoupled design is poorly suited to SNNs: sparse spikes and timestep-local surrogate gradients provide only fragmented supervision for each temporal weight, while the parameter costs grow linearly with T. To address these issues, Delta SNN introduces a compact temporal modulation around a shared synaptic backbone:

$$\begin{aligned} \mathbf{W}_t &= \mathbf{W}_{\text{base}} + \Delta\mathbf{W}_t, \\ [\Delta\mathbf{W}_1, \ldots, \Delta\mathbf{W}_T] &= [\widehat{\mathbf{M}}_1\mathbf{U}_1, \ldots, \widehat{\mathbf{M}}_T\mathbf{U}_T]. \end{aligned} \tag{3}$$

Here, $\mathbf{W}_{\text{base}} \in \mathbb{R}^{m\times d}$ serves as a shared synaptic backbone, providing a stable connectivity prior that couples all timesteps and regularizes temporal learning. The modulation $\Delta\mathbf{W}_t$ is compactly parameterized by a projection matrix $\mathbf{U}_t \in \mathbb{R}^{q\times d}$ and a routing matrix $\widehat{\mathbf{M}}_t \in \mathbb{R}^{m\times q}$, where q denotes the modulation width. Here, $\widehat{\mathbf{M}}_t$ is denoted as a general matrix, which is later constrained into a scaled binary form for spike-driven inference. This formulation introduces only $\mathcal{O}(q/m + q/d)$ relative computational overhead over the backbone, which becomes negligible when $q \ll \min(m, d)$. Therefore, Delta SNN improves temporal adaptability with marginal additional cost. To characterize the underlying temporal dynamics, Delta SNN jointly learns the base weight and time-specific modulations from scratch in an end-to-end manner, guided directly by the optimization objective.

Remark 3.1 (Biological Motivation)**.** Temporal synaptic heterogeneity serves as a fundamental mechanism in biological neural systems, enabling downstream neurons to integrate pre-synaptic spikes with evolving efficacy (Abbott & Regehr, 2004; Wang et al., 2006; Chabrol et al., 2015). This dynamic process supports diverse temporal representations which robustly encode rhythmic signals and periodic patterns. Meanwhile, such dynamics exhibit transient modulations that evolve smoothly over a stable structural backbone rather than independent rewiring at every instant (Harris et al., 2002). Motivated by this principle, Delta SNN translates fine-grained, time-specific synaptic dynamics into a concrete, machine-learning-oriented architecture.

Addition-Preserving Binary Synaptic Routing. Although the compact structure reduces computational complexity, directly computing the temporal modulation introduces floating-point matrix multiplications between $\widehat{\mathbf{M}}_t$ and $\mathbf{U}_t x(t)$. This breaks the desired sparse and addition-only computation pattern (Luo et al., 2024; Yao et al., 2024), potentially weakening the energy efficiency advantage of SNNs. To preserve the spike-driven inference paradigm, we introduce *binary synaptic routing*, where the routing factor is instantiated as a calibrated binary matrix $\widehat{\mathbf{M}}_t \in \{0, \lambda\}^{m\times q}$. Specifically, we use a latent full-precision proxy $\mathbf{Z}_t$ during training and obtain the binary routing matrix as:

$$\mathbf{M}_t = H(\mathbf{Z}_t - \kappa)\ ,\ \widehat{\mathbf{M}}_t = \lambda\mathbf{M}_t, \tag{4}$$

where $\mathbf{M}_t \in \{0, 1\}^{m\times q}$, κ is the routing threshold, and λ is a learnable layer-wise calibration scale. Gradients through the Heaviside step function $H(\cdot)$ w.r.t. $\mathbf{Z}_t$ are estimated using the straight-through estimator (STE). After training, we absorb the calibration scale λ into $\mathbf{U}_t$ so that the routing matrix used at inference becomes purely binary:

$$\widehat{\mathbf{M}}_t\mathbf{U}_t = (\lambda\mathbf{M}_t)\mathbf{U}_t = \mathbf{M}_t(\lambda\mathbf{U}_t) = \mathbf{M}_t\overline{\mathbf{U}}_t\ ,\ \ \overline{\mathbf{U}}_t = \lambda\mathbf{U}_t. \tag{5}$$

Since $\mathbf{M}_t$ is a 0/1 matrix, it can be stored in a 1-bit format, making its storage overhead negligible. Importantly, the timestep-specific synaptic transformation during inference becomes:

$$\mathbf{W}_t x(t) = \mathbf{W}_{\text{base}} x(t) + \mathbf{M}_t\big(\overline{\mathbf{U}}_t x(t)\big). \tag{6}$$

Since $x(t)$ is spike-valued, both $\mathbf{W}_{\text{base}}x(t)$ and $\overline{\mathbf{U}}_t x(t)$ can be computed by accumulating weights over active spikes. The binary mask $\mathbf{M}_t$ then performs lightweight routing through selection and accumulation. Therefore, the original dense floating-point multiplications are replaced by mask-based routing and accumulation operations, preserving the low-power spike-driven sparse inference paradigm and maintaining compatibility with neuromorphic chips.

4 Theoretical Analysis

In this section, we formally characterize how Delta SNN enhances network expressivity, using input space partitioning as an indicator for expressive power. We further provide principled guidance for selecting the modulation width q.

4.1 Input Space Partitioning in SNNs

Input-space partitioning serves as a classic metric for measuring the representational complexity of neural networks (Pascanu et al., 2013; Montúfar et al., 2014; Nguyen et al., 2025). In networks comprising linear transformations followed by thresholding activations (e.g., SNNs), each neuron defines a family of hyperplanes that partition the input space into polyhedral regions. Within each region, the network's mapping remains constant; thus, the total number of such regions provides a rigorous upper bound on the model's expressivity.

For LIF neurons defined in Equation 1, the decision hyperplane at time step t is governed by the following hyperplane equation:

$$\mathbf{W}x + b + \frac{\beta^t u(0) - \vartheta\left(1 + \sum_{\tau=1}^{t-1} \beta^\tau s(t-\tau)\right)}{\sum_{\tau=0}^{t-1} \beta^\tau} = 0. \tag{7}$$

Expressivity Bottleneck of Conventional SNNs. Since the synaptic weight $\mathbf{W}$ is time-invariant in vanilla SNNs, the normal vectors of the hyperplanes across different timesteps $t \in \{1, \ldots, T\}$ remain identical, rendering these hyperplanes strictly parallel (see Figure 6a). This structural constraint limits the granularity of input-space partitioning.

4.2 Expressivity Gain via Time-Variant Weights

We now formalize the expressivity gain brought by the compact temporal modulation in Delta SNN.

Theorem 4.1 (Layer-wise Input Partitioning)**.** *Consider a Delta SNN layer with n neurons processing d-dimensional dynamic inputs $x(t) = x + y_t$, where y_t denotes the time-dependent perturbation. Under compact temporal modulation with modulation width q, the number of polyhedral regions N_{layer} induced by the layer over T timesteps is bounded by:*

$$N_{\text{layer}}(T, q) \leq \prod_{k=1}^{n} \left(\sum_{j=0}^{\min(d, 1+q)} \binom{M(T)}{j} \right), \tag{8}$$

where $M(T) = (T^2 + T)/2$ and T denotes the number of timesteps.

Remark 4.2. Theorem 4.1 establishes the expressivity gain of Delta SNN, with the detailed proof provided in Appendix A. To quantify the scaling gap, we compare the partition complexity N_{Delta} of Delta SNN with the theoretical upper bound N_{Const} of conventional constant-weight SNNs:

$$N_{\text{Const}} \leq M(T) + 1 \approx \mathcal{O}(T^2)\,, \quad N_{\text{Delta}} \leq \sum_{j=0}^{\min(d, 1+q)} \binom{M(T)}{j} \approx \mathcal{O}\left(T^{2(1+q)}\right). \tag{9}$$

This comparison shows that vanilla SNNs exhibit only polynomial growth in T due to the parallel-hyperplane constraint. In contrast, Delta SNN breaks this geometric rigidity through time-variant weights, allowing hyperplanes from different timesteps to intersect and thereby form a more expressive partition of the input space (see Figure 6b). The resulting shift from $\mathcal{O}(T^2)$ to $\mathcal{O}(T^{2(1+q)})$ confirms that compact temporal modulation can substantially enrich the representational capacity of SNNs.

4.3 Theoretical Guidance for Modulation Width Selection

A key question in Delta SNN is how to choose the modulation width q to balance expressivity and parameter efficiency. Our analysis reveals an expressivity saturation phenomenon, which provides theoretical guidance for selecting an effective compact modulation configuration.

Corollary 4.3 (Expressivity Saturation)**.** *Given T and dimension $d > 1+q$, let $s = \min(d, 1+q)$ and $S_s(T) = \sum_{j=0}^{s} \binom{M(T)}{j}$. The sequence $\{S_s(T)\}$, which measures the input-partitioning expressivity, is non-decreasing and converges to $2^{M(T)}$. For any $\varepsilon \in (0, 1)$, there exists a threshold $s_0(T, \varepsilon)$ such that for all $s \geq s_0$, the partition count reaches at least $(1 - \varepsilon)$ of the theoretical maximum.*

Example. For $T = 4$, we have $M(4) = 11$ and the maximal partition count is $2^{11} = 2048$. In this case, $S_5 = 1024, \quad S_8 = 1981, \quad S_{11} = 2048$, showing that expressivity rapidly approaches saturation. For example, with $\varepsilon = 0.05$, the saturation threshold is $s_0 = 8$, implying that moderate modulation width already captures most of the combinatorial expressivity.

Strategic Modulation Width Selection. Guided by this insight, we set the modulation width using a compact reduction ratio, i.e., $q = \lfloor d/64 \rfloor$. This configuration allows the effective dimension $s = \min(d, 1 + q)$ to enter the saturation regime, achieving strong representational capacity with minimal additional parameters. The ablation in section 5.4 further confirms that increasing the width beyond this setting brings only marginal gains.

Table 1: Performance of automotive detection on event-based Gen1 dataset. Following the format in Luo et al. (2024), T denotes temporal bins for ANNs and forward steps for SNNs, while D is the activation upper bound of SpikeYOLO during training (with $D = 1$ being the default for other SNNs and ANNs). "-" indicates that the corresponding metric is unavailable.

Model	Architecture	$T \times D$	#Params.(M)	Energy(mJ) ↓	mAP@50(%) ↑	mAP@50:95(%) ↑
YOLOv3-tiny (Redmon & Farhadi, 2018)	ANN	1	10.2	5.1	44.5	-
RED (Perot et al., 2020)	ANN	5	24.1	-	-	40.0
RVT (Gehrig & Scaramuzza, 2023)	ANN	10	18.5	116.7	70.1	47.2
SAST (Peng et al., 2024)	ANN	-	18.9	70.0	-	48.2
Hybrid (Ahmed et al., 2025)	ANN	10	6.6	8.4	61.0	35.0
SpikingYolo (Kim et al., 2020)	SNN	500	7.9	102.3	44.2	-
MobileNet+SSD (Cordone et al., 2022)	SNN	1	24.3	5.7	35.0	14.7
EMS-YOLO (Su et al., 2023)	SNN	5	14.4	3.4	59.0	31.0
SpikeFPN (Zhang et al., 2024)	SNN	3	22.0	2.6	47.7	22.3
SpikingCenterNet (Bodden et al., 2024)	SNN	5	13.0	5.0	-	22.9
SFOD (Fan et al., 2024)	SNN	5	11.9	7.3	59.3	32.1
EAS-SNN (Wang et al., 2024c)	SNN	3	25.3	28.1	69.9	37.5
SpikingViT (Yu et al., 2024)	SNN	64	21.5	11.9	61.6	39.4
CREST (Mao et al., 2025)	SNN	5	7.6	6.3	63.2	36.0
MSD (Li et al., 2025)	SNN	5	7.8	6.5	66.3	38.9
SpikeYOLO (Luo et al., 2024)	SNN	2×2	23.1	7.8	66.1	39.0
***Delta SNN* (Ours)**	SNN	2×2	24.0	**6.5 (-1.3)**	**67.7 (+1.6)**	**41.2 (+2.2)**
SpikeYOLO (Luo et al., 2024)	SNN	2×4	23.1	7.1	67.0	40.1
***Delta SNN* (Ours)**	SNN	2×4	24.0	**6.3 (-0.8)**	**69.1 (+2.1)**	**41.9 (+1.8)**
SpikeYOLO (Luo et al., 2024)	SNN	4×1	23.1	14.8	66.0	38.4
***Delta SNN* (Ours)**	SNN	4×1	24.9	**11.5 (-3.3)**	**68.9 (+2.9)**	**41.3 (+2.9)**
SpikeYOLO (Luo et al., 2024)	SNN	4×2	23.1	12.9	67.2	40.4
***Delta SNN* (Ours)**	SNN	4×2	24.9	**12.0 (-0.9)**	**69.6 (+2.4)**	**43.3 (+2.9)**

5 EXPERIMENTS

To validate the effectiveness of Delta SNN, we conduct experiments across three representative event vision tasks. Detailed dataset descriptions and training strategies are provided in Appendix B.

5.1 EVENT-BASED OBJECT DETECTION ON GEN1

The Gen1 dataset (De Tournemire et al., 2020) is a large-scale neuromorphic object detection benchmark. It includes 39 hours of real-world driving data captured by an ATIS event camera, with over 255,000 manually annotated bounding boxes for pedestrians and vehicles.

Our experiments adhere to the SpikeYOLO pipeline (Luo et al., 2024), which leverages Integer-LIF neurons to enable integer-activated training while converting activations to spike sequences during inference. We integrate Delta SNN into all convolutional and linear modules. For each layer with a time-invariant base weight $\mathbf{W}_{\text{base}} \in \mathbb{R}^{m \times d}$, the modulation width q is set by a uniform reduction ratio of $q/\min(m, d) = 1/64$. For energy estimation, we adopt the standard protocol in the SNN literature (Roy et al., 2019; Yin et al., 2021), with the detailed methodology provided in Appendix F.

As shown in Table 1, Delta SNN consistently improves detection accuracy across different $T \times D$ settings. Under $T \times D = 4 \times 1$, Delta SNN surpasses SpikeYOLO by 2.9% in both mAP@50 and mAP@50:95. It also achieves lower inference energy, likely because time-variant synaptic modulation improves information transmission under sparse firing. With $T \times D = 4 \times 2$, Delta SNN reaches state-of-the-art performance among SNN-based methods on Gen1. Compared with strong ANN detectors such as RVT (Gehrig & Scaramuzza, 2023) and SAST (Peng et al., 2024), our method significantly narrows the accuracy gap while consuming less than 20% of their estimated energy.

5.2 HUMAN ACTIVITY RECOGNITION ON HAR-DVS

HAR-DVS (Wang et al., 2024b) is a large-scale event-based human action recognition benchmark with 300 categories and 107,646 samples captured by a DAVIS346 camera.

Our experiments are built upon the transformer-based E-SpikeFormer (Yao et al., 2025), which employs Integer-LIF neurons for precise encoding. We integrate the proposed Delta SNN into all

Table 2: Performance of human action recognition on event-based HAR-DVS dataset.

Model	Type	$T \times D$	Acc.(%) ↑
ACTION-Net (Wang et al., 2021)	ANN	8	46.9
TimeSformer (Bertasius et al., 2021)	ANN	8	50.8
Video-SwinTrans (Liu et al., 2022)	ANN	8	51.9
ESTF (Wang et al., 2024b)	ANN	8	51.2
Res-SNN-34 (Fang et al., 2021a)	SNN	8	46.1
ASA-SNN (Yao et al., 2023a)	SNN	8	47.1
Meta-SpikeFormer (Yao et al., 2024)	SNN	8	47.5
E-SpikeFormer (Yao et al., 2025)	SNN	8×4	48.9
***Delta SNN* (Ours)**	SNN	8×4	**51.0 (+2.1)**
E-SpikeFormer (Yao et al., 2025)	SNN	8×8	49.2
***Delta SNN* (Ours)**	SNN	8×8	**51.4 (+2.2)**

Table 3: Performance of video anomaly detection on event-based UCF-Crime-DVS dataset.

Model	Type	AUC(%) ↑	FAR(%) ↓
MIL (Sultani et al., 2018)	ANN	55.6	8.7
3C-Net (Narayan et al., 2019)	ANN	59.2	9.5
AR-Net (Wan et al., 2020)	ANN	60.7	8.5
HL-Net (Wu et al., 2020)	ANN	58.6	34.4
RTFM (Tian et al., 2021)	ANN	52.7	13.2
TSA (Joo et al., 2023)	ANN	51.9	22.4
SEW-ResNet (Fang et al., 2021a)	SNN	54.0	11.8
PLIF (Fang et al., 2021b)	SNN	54.7	9.2
Spikingformer (Zhou et al., 2023a)	SNN	62.8	11.5
MSF (Qian et al., 2025)	SNN	65.0	3.3
***Delta SNN* (Ours)**	SNN	**67.7 (+2.7)**	**0.1 (-3.2)**

learnable weights of the model. Considering the large number of time steps ($T = 8$), we employ a window-based strategy to mitigate excessive parameter growth and memory overhead, while exploiting the local temporal correlation among event bins. Specifically, we define a window size of 2 time steps and apply distinct modulation matrices within each window.

As reported in Table 2, the E-SpikeFormer equipped with Delta SNN consistently achieves a marked improvement in Top-1 accuracy over the original baseline. Meanwhile, Delta SNN exhibits a sparse firing pattern, reducing energy consumption by 10% compared to its constant-weight counterpart. Our model outperforms previous SNN baselines and is comparable to advanced ANN models, such as TimeSformer (Bertasius et al., 2021) and Video-SwinTrans (Liu et al., 2022).

5.3 Video Anomaly Detection on UCF-Crime-DVS

UCF-Crime-DVS (Qian et al., 2025) is the first large-scale event-based benchmark for weakly supervised video anomaly detection. Each sample is converted from a long event stream into an event-based video, typically containing over 100 event frames.

Our implementation follows the Multi-Scale Spiking Fusion (MSF) pipeline (Qian et al., 2025). We integrate Delta SNN into the Conv1d and Linear modules of MSF. Following the HAR-DVS setup, we split the temporal sequence into four windows and learn distinct modulation within each window.

Following standard protocols, we report the Area Under the Curve (AUC) and False Alarm Rate (FAR) for performance comparison. As illustrated in Table 3, our method offers a tangible enhancement in AUC and significantly reduces the FAR to a mere 0.1%. These empirical results validate the superiority of Delta SNN in capturing long-range dependencies in complex temporal video data.

5.4 Analytical Experiments

Effect of Modulation Width To study the effect of the modulation width q in $\mathbf{U}_t \in \mathbb{R}^{q \times d}$ and $\widehat{\mathbf{M}}_t \in \mathbb{R}^{m \times q}$, we conduct ablations on Gen1 with Delta SNN ($T \times D = 2 \times 4$). For each layer with base weight $\mathbf{W}_{\text{base}} \in \mathbb{R}^{m \times d}$, we vary the uniform width ratio $q/\min(d, m)$ across the network, covering ratios from $1/64$ to $1/4$, as well as a minimal setting where $q = 1$ for all layers, denoted as "$1/d$". We also include two boundary cases: ratio 0, corresponding to the constant-weight SNN, and ratio 1, where timestep-specific weights are fully decoupled. Temporal weight modulation consistently improves over the constant-weight baseline (Figure 3). Even with the minimal width $q = 1$, which

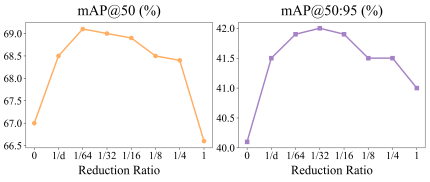


Figure 3: Delta SNN accuracy across modulation width reduction ratios.

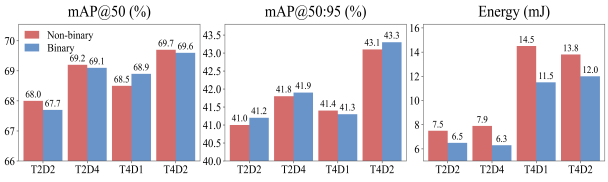
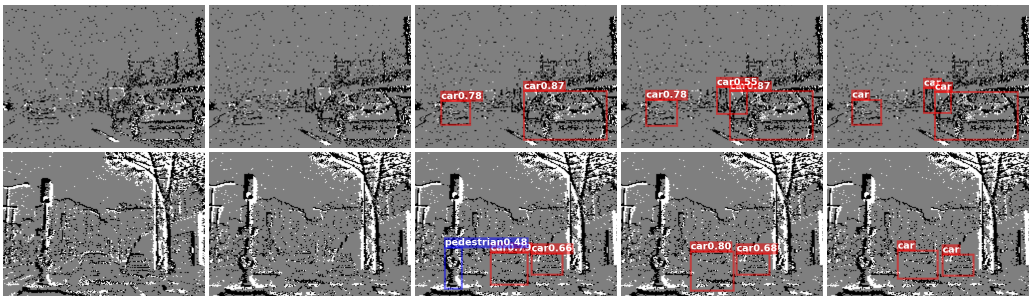


Figure 4: Effect of Binary Synaptic Routing on prediction accuracy and inference efficiency.



(a) Input frame 1 (b) Input frame 2 (c) Constant-Weight SNN (d) Delta SNN (e) Ground truth

Figure 5: Detection results on the Gen1 dataset (More results provided in Appendix H). (a)–(b): input event frames; (c)–(d): model outputs; (e): ground truth. Delta SNN yields more accurate detections.

adds only about 1% training parameters, Delta SNN improves mAP@50 and mAP@50:95 by $+1.5\%$ and $+1.4\%$, respectively. In contrast, fully decoupled weights degrade performance, likely due to harder optimization and overfitting under sparse, noisy supervision. The best performance appears around the $1/64$ ratio, empirically supporting our analysis of modulation width selection.

Ablation of Binary Synaptic Routing. To assess Binary Synaptic Routing, we ablate the binarization of $\mathbf{M}_t$ on Gen1 with Delta SNN. As shown in Fig. 4, binary $\mathbf{M}_t$ yields almost identical mAP@50 and mAP@50:95 across all $T \times D$ settings, indicating that temporal modulation does not require high-precision routing. We attribute this to the role of $\mathbf{M}_t$ in selecting timestep-specific compact bases, while fine-grained synaptic variations are captured by the full-precision $\mathbf{U}_t$. Meanwhile, Binary Routing consistently reduces energy cost, achieving a 20.7% reduction under T4D1 (14.5 mJ $\rightarrow 11.5$ mJ). More importantly, it preserves the sparse, event-driven computation paradigm of SNNs through mask-based AC operations. These results show that Binary Routing improves efficiency while maintaining accuracy and neuromorphic compatibility.

Training and Inference Efficiency. To quantify the overhead of time-variant weights, we benchmark Delta SNN against its constant-weight counterpart on a single NVIDIA H200 GPU (Table 4). On Gen1 ($T = 4$, $D = 1$, batch size 128), Delta SNN increases the parameter count by only 8% and training and inference latency by 5% and 8%, respectively, while keeping training memory nearly unchanged. On HAR-DVS ($T = 8$, $D = 8$, batch size 64), the overheads are similarly modest. These results confirm that compact modulation and binary routing preserve practical efficiency.

Table 4: Computational comparison between vanilla SNN and Delta SNN. Delta SNN introduces only marginal parameter, latency, and memory overhead during both training and inference.

Setting	Params. (M)	Train Lat. (s/epoch)	Train Mem. (GB)	Infer Lat. (ms/sample)	Infer Mem. (GB)
Vanilla SNN on Gen1 (T4D1)	23.12	751	102.5	28.44	1.94
Delta SNN on Gen1 (T4D1)	24.87 (1.08×)	787 (1.05×)	103.0 (1.005×)	30.66 (1.08×)	2.22 (1.14×)
Vanilla SNN on HAR-DVS (T8D8)	18.75	666	75.88	13.50	0.69
Delta SNN on HAR-DVS (T8D8)	20.14 (1.07×)	740 (1.11×)	79.53 (1.05×)	15.31 (1.13×)	0.73 (1.06×)

Prediction Visualization. We compare the predictions of Delta SNN ($T \times D = 2 \times 4$) with those of its constant-weight counterpart in Figure 5. Our method consistently yields more precise predictions, demonstrating its effectiveness in processing dynamic event-based inputs.

6 Conclusion

We present Delta SNN, which addresses the constant weight bottleneck in SNNs and enhances their spatiotemporal modeling capacity for event vision. The time-variant connectivity in Delta SNN is instantiated through compact weight modulation and binary synaptic routing. The effectiveness of our design is justified by both theoretical and empirical evidence. These results suggest that Delta SNN provides a practical and theoretically grounded solution for neuromorphic vision.

AI USE STATEMENT

In this work, we used generative AI tools to interpret results. We have not used generative AI tools to generate synthetic datasets; help develop theoretical models or conceptual frameworks; formulate mathematical claims; provide critical ingredients for proving mathematical claims; assist in the writing of proofs; propose or refine hypotheses; design or provide feedback on research methodology or experiments; implement methods; assist with translation; clean or reformat datasets; or support qualitative or thematic data analysis. Additionally, we used generative AI tools to edit the research paper to improve readability. We have reviewed all AI-assisted work: the authors independently checked AI-assisted interpretations against the original experimental results and verified all AI-assisted edits for accuracy and consistency with the intended technical meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we describe the experimental setups in Section 5, with detailed information on the datasets, data processing procedures, training protocols, hyperparameters, and model configurations provided in Appendix B. The assumptions and derivations underlying our theoretical results are presented in Section 4, and complete statements and proofs are provided in Appendix A. The source code will be made publicly available in the final version.

REFERENCES

Larry F Abbott and Wade G Regehr. Synaptic computation. *Nature*, 431(7010):796–803, 2004.

Soikat Hasan Ahmed, Jan Finkbeiner, and Emre Neftci. Efficient event-based object detection: A hybrid neural network with spatial and temporal attention. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13970–13979, 2025.

Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7243–7252, 2017.

Malyaban Bal and Abhronil Sengupta. P-SpikeSSM: Harnessing probabilistic spiking state space models for long-range dependency tasks. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Sf4ep9Udjf.

Maximilian Baronig, Romain Ferrand, Silvester Sabathiel, and Robert Legenstein. Advancing spatio-temporal processing through adaptation in spiking neural networks. *Nature Communications*, 16: 5776, 2025. doi: 10.1038/s41467-025-60878-z.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 813–824. PMLR, 2021. URL https://proceedings.mlr.press/v139/bertasius21a.html.

Lennard Bodden, Duc Bach Ha, Franziska Schwaiger, Lars Kreuzberg, and Sven Behnke. Spiking centernet: A distillation-boosted spiking neural network for object detection. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. IEEE, 2024.

Siqi Cai, Pengcheng Sun, Tanja Schultz, and Haizhou Li. Low-latency auditory spatial attention detection based on spectro-spatial features from EEG. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, pp. 5812–5815, 2021.

Natalia Caporale and Yang Dan. Spike timing–dependent plasticity: a hebbian learning rule. *Annu. Rev. Neurosci.*, 31(1):25–46, 2008.

François P Chabrol, Alexander Arenz, Martin T Wiechert, Troy W Margrie, and David A DiGregorio. Synaptic diversity enables temporal coding of coincident multisensory inputs in single neurons. *Nature neuroscience*, 18(5):718–727, 2015.

Bharatesh Chakravarthi, Aayush Atul Verma, Kostas Daniilidis, Cornelia Fermuller, and Yezhou Yang. Recent event camera innovations: A survey. In *European Conference on Computer Vision*, pp. 342–376. Springer, 2024.

Ding Chen, Peixi Peng, Tiejun Huang, and Yonghong Tian. Fully spiking actor network with intralayer connections for reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2881–2893, 2025. doi: 10.1109/TNNLS.2024.3352653.

Loïc Cordone, Benoît Miramond, and Philippe Thierion. Object detection with spiking neural networks on automotive event data. In *2022 International Joint Conference on Neural Networks*, pp. 1–8. IEEE, 2022.

Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2744–2757, 2022. doi: 10.1109/TNNLS.2020.3044364.

Neetha Das, Tom Francart, and Alexander Bertrand. Auditory attention detection dataset KULeuven, 2019. URL https://zenodo.org/records/4004271.

Pierre De Tournemire, Davide Nitti, Etienne Perot, Davide Migliore, and Amos Sironi. A large scale event-based detection dataset for automotive. *arXiv preprint arXiv:2001.08499*, 2020.

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

Chaoteng Duan, Jianhao Ding, Shiyan Chen, Zhaofei Yu, and Tiejun Huang. Temporal effective batch normalization in spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/de2ad3ed44ee4e675b3be42aa0b615d0-Abstract-Conference.html.

Cunhang Fan, Hongyu Zhang, Qinke Ni, Jingjing Zhang, Jianhua Tao, Jian Zhou, Jiangyan Yi, Zhao Lv, and Xiaopei Wu. Seeing helps hearing: A multi-modal dataset and a mamba-based dual branch parallel network for auditory attention decoding. *Information Fusion*, 118:102946, 2025. doi: 10.1016/j.inffus.2025.102946.

Yimeng Fan, Wei Zhang, Changsong Liu, Mingyang Li, and Wenrui Lu. Sfod: Spiking fusion object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17191–17200, 2024.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021a.

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

Yuetong Fang, Deming Zhou, Ziqing Wang, Hongwei Ren, Zecui Zeng, Lusong Li, Shibo Zhou, and Renjing Xu. Spiking neural networks need high-frequency information. In *Advances in Neural Information Processing Systems*, volume 38, pp. 103404–103425, 2025. doi: 10.52202/085713-3455.

Răzvan V Florian. Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural computation*, 19(6):1468–1502, 2007.

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

Daniel Gehrig and Davide Scaramuzza. Low-latency automotive vision with event cameras. *Nature*, 629(8014):1034–1040, 2024.

Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13884–13893, 2023.

Jesse Hagenaars, Federico Paredes-Vallés, and Guido De Croon. Self-supervised learning of event-based optical flow with spiking neural networks. *Advances in Neural Information Processing Systems*, 34:7167–7179, 2021.

Ryuhei Hamaguchi, Yasutaka Furukawa, Masaki Onishi, and Ken Sakurada. Hierarchical neural memory network for low latency event processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22867–22876, 2023.

Ilyass Hammouamri, Ismail Khalfaoui-Hassani, and Timothée Masquelier. Learning delays in spiking neural networks using dilated convolutions with learnable spacings. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4r2ybzJnmN.

Kenneth D Harris, Darrell A Henze, Hajime Hirase, Xavier Leinekugel, George Dragoi, Andras Czurkó, and György Buzsáki. Spike train dynamics predicts theta-related phase precession in hippocampal pyramidal cells. *Nature*, 417(6890):738–741, 2002.

Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*, pp. 10–14. IEEE, 2014.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yifan Hu, Lei Deng, Yujie Wu, Man Yao, and Guoqi Li. Advancing spiking neural networks toward deep residual learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2): 2353–2367, 2025. doi: 10.1109/TNNLS.2024.3355393.

Yulong Huang, Zunchang Liu, Changchun Feng, Xiaopeng Lin, Hongwei Ren, Haotian Fu, Yue Zhou, Hong Xing, and Bojun Cheng. PRF: Parallel resonate and fire neuron for long sequence learning in spiking neural networks. *arXiv preprint arXiv:2410.03530*, 2024.

Ziyuan Huang, Shiwei Zhang, Liang Pan, Zhiwu Qing, Mingqian Tang, Ziwei Liu, and Marcelo H. Ang, Jr. TAda! temporally-adaptive convolutions for video understanding. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=izj68lUcBpt.

Sangwoo Hwang, Seunghyun Lee, Dahoon Park, Donghun Lee, and Jaeha Kung. Spike-dattention: Training-free and fully spike-driven transformer-to-snn conversion with winner-oriented spike shift for softmax operation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 67422–67445, 2024. doi: 10.52202/079017-2152. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/7c9341ad0263428b5057d92f4d88dfa0-Paper-Conference.pdf.

Skyler L Jackman, Josef Turecek, Justine E Belinsky, and Wade G Regehr. The calcium sensor synaptotagmin 7 is required for synaptic facilitation. *Nature*, 529(7584):88–91, 2016.

Hyekang Kevin Joo, Khoa Vo, Kashu Yamazaki, and Ngan Le. Clip-tsa: Clip-assisted temporal self-attention for weakly-supervised video anomaly detection. In *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 3230–3234. IEEE, 2023.

Jaehyun Kim, Heesu Kim, Subin Huh, Jinho Lee, and Kiyoung Choi. Deep neural networks with weighted spikes. *Neurocomputing*, 311:373–386, 2018.

Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 11270–11277, 2020.

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

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

Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. CIFAR10-DVS: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 11:309, 2017. doi: 10.3389/fnins.2017.00309.

Ziqi Li, Tao Gao, Yisheng An, Ting Chen, Jing Zhang, Yuanbo Wen, Mengkun Liu, and Qianxi Zhang. Brain-inspired spiking neural networks for energy-efficient object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3552–3562, 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3202–3211, 2022.

Zeyu Liu, Gourav Datta, Anni Li, and Peter Anthony Beerel. LMUFormer: Low complexity yet powerful spiking model with legendre memory units. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oEF7qExD9F.

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

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

Ruixin Mao, Aoyu Shen, Lin Tang, and Jun Zhou. Crest: An efficient conjointly-trained spike-driven framework for event-based object detection exploiting spatiotemporal dynamics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 6072–6080, 2025.

Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *Advances in neural information processing systems*, 27, 2014.

Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8679–8687, 2019.

Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Duc Anh Nguyen, Ernesto Araya, Adalbert Fono, and Gitta Kutyniok. Time to spike? understanding the representational power of spiking neural networks in discrete time. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=aqZKgwf7Cc.

Qinke Ni, Hongyu Zhang, Cunhang Fan, Shengbing Pei, Chang Zhou, and Zhao Lv. DBPNet: Dual-branch parallel network with temporal-frequency fusion for auditory attention detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 3115–3123, 2024. doi: 10.24963/ijcai.2024/345.

Peter O'Connor, Daniel Neil, Shih-Chii Liu, Tobi Delbruck, and Michael Pfeiffer. Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in neuroscience*, 7: 178, 2013.

Peter Orlik and Hiroaki Terao. *Arrangements of hyperplanes*, volume 300 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1992. doi: 10.1007/978-3-662-02772-1.

Razvan Pascanu, Guido Montufar, and Yoshua Bengio. On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv preprint arXiv:1312.6098*, 2013.

Yansong Peng, Hebei Li, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. Scene adaptive sparse transformer for event-based object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16794–16804, 2024.

Etienne Perot, Pierre De Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. *Advances in Neural Information Processing Systems*, 33:16639–16652, 2020.

Yuanbin Qian, Shuhan Ye, Chong Wang, Xiaojie Cai, Jiangbo Qian, and Jiafei Wu. Ucf-crime-dvs: A novel event-based dataset for video anomaly detection with spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 6577–6585, 2025.

Nazeerah Abdul Rahman and Nooraini Yusoff. Modulated spike-time dependent plasticity (stdp)-based learning for spiking neural network (snn): A review. *Neurocomputing*, 618:129170, 2025.

Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pp. 969–982. PMLR, 2018.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

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

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

Waseem Shariff, Mehdi Sefidgar Dilmaghani, Paul Kielty, Mohamed Moustafa, Joe Lemley, and Peter Corcoran. Event cameras in automotive sensing: A review. *IEEE Access*, 12:51275–51306, 2024.

Guobin Shen, Dongcheng Zhao, Yiting Dong, Yang Li, Feifei Zhao, and Yi Zeng. Learning the plasticity: Plasticity-driven learning framework in spiking neural networks. *arXiv preprint arXiv:2308.12063*, 2023.

Society for Machinery Failure Prevention Technology. Fault data sets, 2024. URL `https://mfpt.org/fault-data-sets/`.

Matei-Ioan Stan and Oliver Rhodes. Learning long sequences in spiking neural networks. *Scientific Reports*, 14:21957, 2024. doi: 10.1038/s41598-024-71678-8.

Christoph Stöckl and Wolfgang Maass. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, 3(3):230–238, 2021.

Qiaoyi Su, Yuhong Chou, Yifan Hu, Jianing Li, Shijie Mei, Ziyang Zhang, and Guoqi Li. Deep directly-trained spiking neural networks for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6555–6565, 2023.

Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6479–6488, 2018.

Pengfei Sun, Jibin Wu, Paul Devos, and Dick Botteldooren. Towards parameter-free attentional spiking neural networks. *Neural Networks*, 185:107154, 2025. doi: 10.1016/j.neunet.2025.107154.

Guangzhi Tang, Neelesh Kumar, Raymond Yoo, and Konstantinos Michmizos. Deep reinforcement learning with population-coded spiking neural network for continuous control. In *Proceedings of the Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 2016–2029, 2021.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=qVyeW-grC2k`.

Yu Tian, Guansong Pang, Yuanhong Chen, Rajvinder Singh, Johan W Verjans, and Gustavo Carneiro. Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4975–4986, 2021.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Boyang Wan, Yuming Fang, Xue Xia, and Jiajie Mei. Weakly supervised video anomaly detection via center-guided discriminative learning. In *2020 IEEE international conference on multimedia and expo (ICME)*, pp. 1–6. IEEE, 2020.

H. Wang and Y.-F. Li. Bioinspired membrane learnable spiking neural network for autonomous vehicle sensors fault diagnosis under open environments. *Reliability Engineering & System Safety*, 233:109102, 2023. doi: 10.1016/j.ress.2023.109102.

Jiaqi Wang, Liutao Yu, Liwei Huang, Chenlin Zhou, Han Zhang, Zhenxi Song, Min Zhang, Zhengyu Ma, and Zhiguo Zhang. SpikeSCR: Efficient speech command recognition leveraging spiking neural network and curriculum learning-based knowledge distillation. *arXiv preprint arXiv:2412.12858*, 2024a.

Jiaqi Wang, Zhengyu Ma, Xiongri Shen, Chenlin Zhou, Leilei Zhao, Han Zhang, Yi Zhong, Siqi Cai, Zhenxi Song, and Zhiguo Zhang. S^2M-Former: Spiking symmetric mixing branchformer for brain auditory attention detection. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jiaqi Wang, Liutao Yu, Xiongri Shen, Sihang Guo, Chenlin Zhou, Leilei Zhao, Yi Zhong, Zhiguo Zhang, and Zhengyu Ma. SpikCommander: A high-performance spiking transformer with multi-view learning for efficient speech command recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(3):2119–2127, 2026. doi: 10.1609/aaai.v40i3.37194.

Xiao Wang, Zongzhen Wu, Bo Jiang, Zhimin Bao, Lin Zhu, Guoqi Li, Yaowei Wang, and Yonghong Tian. Hardvs: Revisiting human activity recognition with dynamic vision sensors. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 5615–5623, 2024b.

Yuchen Wang, Kexin Shi, Chengzhuo Lu, Yuguo Liu, Malu Zhang, and Hong Qu. Spatial-temporal self-attention for asynchronous spiking neural networks. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 3085–3093, 2023. doi: 10.24963/ijcai.2023/344.

Yun Wang, Henry Markram, Philip H Goodman, Thomas K Berger, Junying Ma, and Patricia S Goldman-Rakic. Heterogeneity in the pyramidal network of the medial prefrontal cortex. *Nature neuroscience*, 9(4):534–542, 2006.

Zhengwei Wang, Qi She, and Aljosa Smolic. Action-net: Multipath excitation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13214–13223, 2021.

Ziming Wang, Ziling Wang, Huaning Li, Lang Qin, Runhao Jiang, De Ma, and Huajin Tang. Eas-snn: End-to-end adaptive sampling and representation for event-based detection with recurrent spiking neural networks. In *European Conference on Computer Vision*, pp. 310–328. Springer, 2024c.

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

Peng Wu, Jing Liu, Yujia Shi, Yujia Sun, Fangtao Shao, Zhaoyang Wu, and Zhiwei Yang. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pp. 322–339. Springer, 2020.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

Qi Xu, Jie Deng, Jiangrong Shen, Biwu Chen, Huajin Tang, and Gang Pan. Hybrid spiking vision transformer for object detection with event cameras. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=WZKcJZWG3P.

Xiran Xu, Bo Wang, Yujie Yan, Xihong Wu, and Jing Chen. A DenseNet-based method for decoding auditory spatial attention with EEG. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1946–1950, 2024. doi: 10.1109/ICASSP48485.2024.10448013.

Zijie Xu, Tong Bu, Zecheng Hao, Jianhao Ding, and Zhaofei Yu. Proxy target: Bridging the gap between discrete spiking neural networks and continuous control. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-5316.

Yinsong Yan, Yujie Wu, and Jibin Wu. Advancing spatiotemporal representations in spiking neural networks via parametric invertible transformation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0ac46bb0a72a7afe311d9b48b5088df8-Paper-Conference.pdf.

Man Yao, Jiakui Hu, Guangshe Zhao, Yaoyuan Wang, Ziyang Zhang, Bo Xu, and Guoqi Li. Inherent redundancy in spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16924–16934, 2023a.

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, 2023b.

Man Yao, JiaKui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo XU, and Guoqi Li. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1SIBN5Xyw7.

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

Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10):905–913, 2021.

Lixing Yu, Hanqi Chen, Ziming Wang, Shaojie Zhan, Jiankun Shao, Qingjie Liu, and Shu Xu. Spikingvit: a multi-scale spiking vision transformer model for event-based object detection. *IEEE Transactions on Cognitive and Developmental Systems*, 2024.

Dehao Zhang, Malu Zhang, Shuai Wang, Jingya Wang, Wenjie Wei, Zeyu Ma, Guoqing Wang, Yang Yang, and Haizhou Li. Dendritic resonate-and-fire neuron for effective and efficient long sequence modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3587.

Duzhen Zhang, Tielin Zhang, Shuncheng Jia, and Bo Xu. Multi-sacle dynamic coding improved spiking actor network for reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):59–67, 2022. doi: 10.1609/aaai.v36i1.19879.

Hu Zhang, Yanchen Li, Luziwei Leng, Kaiwei Che, Qian Liu, Qinghai Guo, Jianxing Liao, and Ran Cheng. Automotive object detection via learning sparse events by spiking neurons. *IEEE Transactions on Cognitive and Developmental Systems*, 2024.

Minghang Zhao, Shisheng Zhong, Xuyun Fu, Baoping Tang, and Michael G. Pecht. Deep residual shrinkage networks for fault diagnosis. *IEEE Transactions on Industrial Informatics*, 16(7): 4681–4690, 2020. doi: 10.1109/TII.2019.2943898.

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 preprint arXiv:2304.11954*, 2023a.

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

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, 2024.

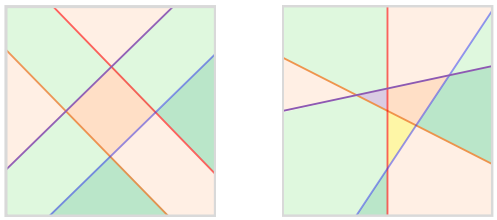
Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. *arXiv preprint arXiv:2209.15425*, 2022.

Lin Zuo, Yongqi Ding, Mengmeng Jing, Kunshan Yang, Biao Chen, and Yunqian Yu. Toward end-to-end bearing fault diagnosis for industrial scenarios with spiking neural networks. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4238–4249, 2025. doi: 10.1145/3711896.3737160.

CONTENTS

1 Introduction . . . **1**
2 Related Works . . . **2**
3 Method . . . **3**
3.1 Preliminary . . . 3
3.2 Delta SNN . . . 3
4 Theoretical Analysis . . . **5**
4.1 Input Space Partitioning in SNNs . . . 5
4.2 Expressivity Gain via Time-Variant Weights . . . 6
4.3 Theoretical Guidance for Modulation Width Selection . . . 6
5 Experiments . . . **7**
5.1 Event-Based Object Detection on Gen1 . . . 7
5.2 Human Activity Recognition on HAR-DVS . . . 7
5.3 Video Anomaly Detection on UCF-Crime-DVS . . . 8
5.4 Analytical Experiments . . . 8
6 Conclusion . . . **9**
AI Use Statement . . . 10
Reproducibility Statement . . . 10
A Theoretical Proof . . . **20**
B Implementation Details . . . **23**
B.1 Object Detection on Gen1 . . . 23
B.2 Human Activity Recognition on HAR-DVS . . . 23
B.3 Video Anomaly Detection on UCF-Crime-DVS . . . 23
C Generalization Beyond the Main Benchmarks . . . **24**
C.1 Additional Event-Based Classification Benchmarks . . . 24
C.2 Generalization to Static Inputs . . . 24
C.3 Generalization Across Temporal Modalities . . . 25
C.4 Long-Range Classification . . . 26
D Temporal Properties and Structure . . . **26**
D.1 Effect of Timestep . . . 26
D.2 Effect of Input Time Horizons . . . 27
D.3 Temporal Structure of Learned Modulations . . . 27
D.4 Modulation Granularity and Accuracy–Efficiency Trade-off . . . 28
D.5 Long-Sequence Scalability . . . 29
E Controlled Comparisons and Method Analysis . . . **30**
E.1 Controlled Comparison with Static Modulation . . . 30
E.2 Comparison with LoRA-Style Adaptation . . . 30
E.3 Comparison with Alternative Temporal Adaptation Mechanisms . . . 31
E.4 Backbone-Controlled Evaluation . . . 32
E.5 Generalization Across Neuron Models . . . 33
F Hardware-Aware Efficiency and Neuromorphic Validation . . . **33**
F.1 Analytical Energy Model . . . 33
F.2 Parameter and Operation Breakdown . . . 34
F.3 Effect of Binary Synaptic Routing . . . 34
F.4 Lava-Based Neuromorphic Functional Validation . . . 35
F.5 Hardware-Aware Memory Access and Latency . . . 35
G Limitation . . . **36**
H Additional Visual Comparison of Prediction Results . . . **37**

A THEORETICAL PROOF



(a) Vanilla SNN (b) Delta SNN

Figure 6: Hyperplane partitioning comparison with two neurons and $T = 2$. Lines denote decision hyperplanes, and colors indicate distinct polyhedral regions. Delta SNN rotates hyperplanes across timesteps via time-variant weights, yielding a richer input-space partition.

Lemma A.1 (Temporal decision boundaries of a single LIF neuron)**.** *(Nguyen et al., 2025) Consider a shallow **discrete-time LIF-SNN** with T time steps and an arbitrary hidden neuron k. Then the spike decisions $\{s_k(t)\}_{t=1}^{T}$ induce a family of at most $\frac{T^2+T}{2} \in O(T^2)$ parallel affine hyperplanes in the input space $\mathbb{R}^{n_{\text{in}}}$. Moreover, this bound is tight: there exist choices of membrane parameters $u_k(0), \beta, \vartheta \in \mathbb{R} \times [0,1] \times \mathbb{R}_+$ for which exactly $H(T)$ such hyperplanes arise.*

Theorem A.2 (Layer-wise input partition under temporal modulation)**.** *Consider a shallow discrete-time LIF-SNN with T time steps and input dimension d. The first hidden layer contains n neurons. For each neuron $k \in \{1, \dots, n\}$, its time-varying synaptic weights $\{w_{k,t}\}_{t=1}^{T} \subset \mathbb{R}^d$ satisfy a compact temporal modulation*

$$w_{k,t} = w_k + \mathbf{U}^\top \widehat{m}_{k,t}, \qquad \mathbf{U} \in \mathbb{R}^{q \times d},\ \widehat{m}_{k,t} \in \mathbb{R}^q,$$

with modulation width q. Let

$$M(T) := \frac{T^2 + T}{2}, \qquad s_k := \min\{d,\ 1 + q\}.$$

Then, for static inputs $x \in \mathbb{R}^d$, the first hidden layer partitions the input space into at most

$$N_{\text{layer}}(T, q) \ \le \ \prod_{k=1}^{n} \left(\sum_{j=0}^{s_k} \binom{M(T)}{j} \right)$$

polyhedral regions, on each of which all spike trains $\big(s_k(t)\big)_{k,t}$ are constant.

Moreover, if all neurons share the same modulation width and $d \ge 1 + q$, so that $s_k = 1 + q$ for all k, this simplifies to

$$N_{\text{layer}}(T, q) \ \le \ \left(\sum_{j=0}^{1+q} \binom{M(T)}{j} \right)^n = O\big(M(T)^{n(1+q)}\big).$$

Proof.

1. Compact structural reduction for each neuron. Fix an arbitrary hidden neuron $k \in \{1, \dots, n\}$. Its time-varying synaptic weights satisfy the compact temporal modulation

$$w_{k,t} = w_k + \mathbf{U}^\top \widehat{m}_{k,t}, \qquad \mathbf{U} \in \mathbb{R}^{q \times d},\ \widehat{m}_{k,t} \in \mathbb{R}^q.$$

All vectors $\{w_{k,t}\}_{t=1}^{T}$ lie in the affine subspace

$$\mathcal{U}_k := w_k + \text{span}(\mathbf{U}^\top) \subset \mathbb{R}^d,$$

whose dimension is at most $\min\{d, 1 + q\}$. Define

$$s_k := \min\{d, 1 + q\}.$$

We introduce the feature map

$$\phi_k(x) := (\langle w_k, x\rangle,\ \mathbf{U}x) \in \mathbb{R}^{1+q},$$

and restrict to the first s_k coordinates if $1 + q > d$. Denote $z_k(x) := \phi_k(x) \in \mathbb{R}^{s_k}$.

For all t,

$$\langle w_{k,t}, x\rangle = \langle w_k + \mathbf{U}^\top \widehat{m}_{k,t}, x\rangle = c_{k,t}^\top z_k(x), \qquad c_{k,t} := (1,\ \widehat{m}_{k,t}^\top)^\top.$$

Thus the LIF dynamics of neuron k depend on the original input x only through the lower-dimensional feature vector $z_k(x)$. There exists a mapping

$$F_k : \mathbb{R}^{s_k} \to \{0, 1\}^T$$

such that

$$(s_k(t))_{t=1}^T = F_k(z_k(x)).$$

2. Temporal partition from Lemma A.1. Lemma A.1 ensures that for a single LIF neuron with T discrete time steps, its input space can be partitioned by at most

$$M(T) := \frac{T^2 + T + 2}{2}$$

parallel hyperplanes into regions on each of which its spike train is constant. Applied to the effective input $z_k(x) \in \mathbb{R}^{s_k}$, we obtain hyperplanes

$$H_{k,1}, \ldots, H_{k,M(T)} \subset \mathbb{R}^{s_k},$$

such that F_k is constant inside every region induced by these hyperplanes.

3. Counting regions for a single neuron. A classical result on hyperplane arrangements states that M hyperplanes in $\mathbb{R}^s$ can create at most

$$\sum_{j=0}^{s} \binom{M}{j}$$

polyhedral regions (Orlik & Terao, 1992). Hence, neuron k alone induces no more than

$$R_k := \sum_{j=0}^{s_k} \binom{M(T)}{j}$$

regions in the original input space $\mathbb{R}^d$, and its full spike train $(s_k(t))_{t=1}^T$ is constant on each region.

4. Intersection of partitions over all neurons. The first hidden layer induces the common refinement of the R_k-region partitions associated with each neuron k. Therefore, the total number of distinct regions satisfies

$$N_{\text{layer}}(T, q)\ \le\ \prod_{k=1}^{n} R_k\ =\ \prod_{k=1}^{n} \left(\sum_{j=0}^{s_k} \binom{M(T)}{j} \right),$$

and on each region all spike trains $\{s_k(t)\}_{k,t}$ are constant.

5. Uniform modulation width and simplified asymptotics. If all neurons share the same modulation width and $d \ge 1 + q$, then $s_k = 1 + q$ for all k, and

$$N_{\text{layer}}(T, q)\ \le\ \left(\sum_{j=0}^{1+q} \binom{M(T)}{j} \right)^n.$$

Since $\sum_{j=0}^{1+q} \binom{M(T)}{j} = O\big(M(T)^{1+q}\big)$ for fixed q, and $M(T) = \Theta(T^2)$, we further obtain

$$N_{\text{layer}}(T, q) = O\big(M(T)^{n(1+q)}\big).$$

This completes the proof. □

Corollary A.3 (Effective saturation of modulation width for fixed temporal horizon)**.** *Let $d > 1 + q$, and T be fixed and define*

$$M(T) := \frac{T^2 + T + 2}{2}, \qquad S_s(T) := \sum_{j=0}^{s} \binom{M(T)}{j} \quad \text{for } s = 0, 1, \ldots, M(T).$$

Then, the sequence $\{S_s(T)\}_{s=0}^{M(T)}$ is non-decreasing in s and satisfies

$$S_{M(T)}(T) = 2^{M(T)}.$$

In particular, for any $\varepsilon \in (0, 1)$ there exists an index $s_0(T, \varepsilon) \in \{0, \ldots, M(T)\}$ such that

$$\frac{S_s(T)}{2^{M(T)}} \;\geq\; 1 - \varepsilon, \qquad \forall\, s \geq s_0(T, \varepsilon).$$

Proof. Note first that for fixed T, $M(T)$ is a finite integer. By definition,

$$S_s(T) = \sum_{j=0}^{s} \binom{M(T)}{j}, \quad s = 0, 1, \ldots, M(T),$$

is a partial sum of the binomial expansion of $(1 + 1)^{M(T)}$. The sequence is non-decreasing in s because each term $\binom{M(T)}{j}$ is non-negative.

Moreover,

$$S_{M(T)}(T) = \sum_{j=0}^{M(T)} \binom{M(T)}{j} = (1 + 1)^{M(T)} = 2^{M(T)}.$$

Since $\{S_s(T)\}_{s=0}^{M(T)}$ is a finite non-decreasing sequence bounded above by $2^{M(T)}$, for any $\varepsilon \in (0, 1)$ there exists at least one index $s_0(T, \varepsilon)$ such that

$$S_{s_0(T,\varepsilon)}(T) \;\geq\; (1 - \varepsilon)\, 2^{M(T)}.$$

By monotonicity, this inequality holds for all $s \geq s_0(T, \varepsilon)$. □

Remark A.4 (Modulation width saturation at $T = 4$)**.** For $T = 4$, $M(4) = 11$ and the theoretical maximum number of regions is $2^{11} = 2048$. The cumulative sums $S_s(4) = \sum_{j=0}^{s} \binom{11}{j}$ yield:

s	5	6	7	8	9	10	11
S_s	1024	1486	1816	1981	2036	2047	2048

Thus $S_8/2048 \approx 96.7\%$ and $S_9/2048 \approx 99.4\%$. For $\varepsilon = 0.05$, $s_0(4, \varepsilon) = 8$. Hence, with input dim. $d \geq 8$, modulation width $q \geq 7$ (so $s_k = \min\{d, 1 + q\} \geq 8$) already achieves near-saturation; increasing q further yields only marginal gains.

Remark A.5 (Comparison with baseline)**.** The recent work (Nguyen et al., 2025) establishes that in a shallow discrete-time LIF-SNN each neuron in the first hidden layer induces at most

$$M(T) = \tfrac{T^2+T+2}{2}$$

parallel hyperplanes across the input space, leading to an overall input partition count scaling as $O(T^2 n_1)$ for a layer of width n_1. In that analysis, the per-time-step weights are allowed to vary arbitrarily.

In contrast, our Theorem A.2 assumes a compact temporal modulation $w_{k,t} = w_k + \mathbf{U}^\top \widehat{m}_{k,t}$, thereby constraining all temporal weight changes to a compact modulation subspace of effective dimension $s_k = \min\{d, 1 + q\}$. Despite this additional constraint, we show that the induced partitioning per neuron can still approach the full combinatorial ceiling $2^{M(T)}$ — more precisely, no more than

$$\sum_{j=0}^{s_k} \binom{M(T)}{j}$$

regions, which for moderate s_k is already near maximal.

Thus our result refines and generalizes in two ways: (i) it accounts for realistic compact temporal modulation parameterizations common in practice, and (ii) provides a quantitative saturation analysis showing that beyond a modest modulation width, expressivity (in terms of input partitioning) saturates — additional modulation width yields only constant-factor benefits.

B Implementation Details

B.1 Object Detection on Gen1

The Gen1 dataset (De Tournemire et al., 2020) is a large-scale neuromorphic object detection benchmark. It includes 39 hours of real-world driving data captured by an ATIS event camera, with over 255,000 manually annotated bounding boxes for pedestrians and vehicles.

We adhere to the training pipeline established in SpikeYOLO (Luo et al., 2024). Each input sample consists of a 250ms event stream preceding the annotation timestamp, temporally partitioned into $T = 2$ or $T = 4$ slices. Regarding the model configuration, we apply a uniform reduction ratio of $q : \min(m, d) = 1 : 64$ across all convolutional and linear layers, where q represents the modulation width and m, d denote the dimensions of each time-invariant base weight matrix.

The model is trained via Stochastic Gradient Descent (SGD) for 50 epochs, with a batch size of 160 and a spatial resolution of 320×320. The learning rate is initialized at 0.02 and decays to 0.004, utilizing a 10-epoch warmup period to stabilize early training. Momentum is set to 0.8, and the bias learning rate is 0.1. All data augmentation strategies are disabled.

B.2 Human Activity Recognition on HAR-DVS

HAR-DVS (Wang et al., 2024b) is a large-scale event-based human activity classification dataset comprising 300 action categories and 107,646 samples, captured using a DAVIS346 camera with a spatial resolution of 346×260.

We conduct direct training on HAR-DVS ($T = 8$), following the data preprocessing pipeline of E-SpikeFormer (Yao et al., 2025). Regarding the Delta SNN configuration, we enforce a uniform reduction ratio of $q : \min(m, d) = 1 : 64$ across learnable weights. As explained in subsection 5.2, we adopt a window-based strategy with a window size of 2, resulting in 4 distinct temporal windows, each modulated by independent compact modulation matrices. The model is optimized using Adam for 50 epochs, with a batch size of 24. We set the initial learning rate to 4×10^{-5} and the weight decay to 10^{-2}.

Under this setting, Delta SNN improves the constant-weight E-SpikeFormer backbone from 49.2% to 51.4% Top-1 accuracy, surpassing prior SNN methods while remaining competitive with strong ANN baselines. The E-SpikeFormer backbone contains 18.75M parameters, whereas Delta SNN contains 20.14M parameters ($1.07\times$), corresponding to an additional 1.39M parameters for temporal modulation.

B.3 Video Anomaly Detection on UCF-Crime-DVS

UCF-Crime-DVS (Qian et al., 2025) is the first large-scale event-based benchmark developed for weakly supervised video anomaly detection. It consists of 1,900 event streams, covering 13 anomaly categories, totaling 128 hours of data with a high resolution of 1280×720. For each sample, the long event stream is sliced and processed into an event-based video, typically containing over 100 event frames.

To ensure a fair comparison, we strictly adhere to the training protocol established in MSF (Qian et al., 2025). Architecturally, we integrate Delta SNNs into the Conv1d and Linear modules of the MSF backbone. The temporal sequence is partitioned into four windows, with distinct compact weight modulation ($q : \min(m, d) = 1 : 64$) applied within each window.

Optimization is performed using the Adam optimizer with a learning rate of 10^{-4} and a weight decay of 5×10^{-4}. We employ a batch size of 60, composed of a balanced mix of normal and abnormal sequences randomly sampled from the training set.

C Generalization Beyond the Main Benchmarks

C.1 Additional Event-Based Classification Benchmarks

To further validate the generalizability of our method across a broader range of event vision tasks, we conduct additional experiments on CIFAR10-DVS (Li et al., 2017) and DVS-Gesture (Amir et al., 2017). Since these datasets are relatively small, we adapt E-SpikeFormer (Yao et al., 2025) into lightweight backbones with approximately 2.6M and 1.3M parameters for CIFAR10-DVS and DVS-Gesture, respectively. We then incorporate our time-variant modulation using a 1:64 reduction ratio and four temporal windows, while keeping the additional parameter overhead small.

Table 5: Results on CIFAR10-DVS.

Method	Top-1 Acc. (%) ↑
Spikformer (Zhou et al., 2022)	80.9
Spikingformer (Zhou et al., 2023a)	81.3
CML (Zhou et al., 2023b)	80.9
Spike-driven Transformer (Yao et al., 2023b)	80.0
QKFormer (Zhou et al., 2024)	84.0
E-SpikeFormer (Baseline) (Yao et al., 2025)	82.3
Delta SNN	**85.2**

Table 6: Results on DVS-Gesture.

Method	Top-1 Acc. (%) ↑
Spikformer (Zhou et al., 2022)	98.3
Spikingformer (Zhou et al., 2023a)	98.3
QKFormer (Zhou et al., 2024)	98.6
CML (Zhou et al., 2023b)	98.6
STSA (Wang et al., 2023)	98.7
E-SpikeFormer (Baseline) (Yao et al., 2025)	97.2
Delta SNN	**99.3**

Delta SNN consistently improves the constant-weight E-SpikeFormer baseline by 2.9% on CIFAR10-DVS and 2.1% on DVS-Gesture. It also surpasses prior competitive models on both datasets. These results further demonstrate that the proposed time-variant modulation generalizes effectively across event-based classification benchmarks.

C.2 Generalization to Static Inputs

To assess whether Delta SNN is beneficial for static, non-event vision tasks, we additionally evaluate it on two static-image benchmarks: CIFAR-100 (Krizhevsky, 2009) classification with E-SpikeFormer (Yao et al., 2025) and COCO (Lin et al., 2014) object detection with SpikeYOLO-N (Luo et al., 2024). For both tasks, we retain the original backbone and training configuration and only introduce the proposed temporal weight modulation.

Table 7: Static-image classification results on CIFAR-100.

Method	Params (M)	Top-1 Acc. (%) ↑
Spikformer (Zhou et al., 2022)	9.3	77.9
QKFormer (Zhou et al., 2024)	6.7	81.2
Max-Former (Fang et al., 2025)	6.6	82.7
E-SpikeFormer (Baseline) (Yao et al., 2025)	8.5	81.9
Delta SNN	**8.8**	**83.2**

Table 8: Static-image object detection results on COCO.

Method	Params (M)	AP (%) ↑	AP_{50} (%) ↑
Spiking-YOLO (Kim et al., 2020)	10.2	25.7	–
EMS-YOLO (Su et al., 2023)	26.9	30.1	50.1
Meta-SpikeFormer (YOLO) (Yao et al., 2024)	16.8	–	50.3
SpikeYOLO-N (Baseline) (Luo et al., 2024)	13.3	43.6	60.4
Delta SNN	**14.4**	**45.0**	**61.8**

Although Delta SNN is primarily designed for dynamic inputs, it also improves static-image classification and detection. We hypothesize that this improvement arises because static inputs still produce temporally evolving membrane potentials and spike representations during multi-step SNN inference, allowing temporal weight modulation to better adapt to the internal dynamics.

C.3 Generalization Across Temporal Modalities

We evaluate Delta SNN on four non-event-vision tasks spanning different temporal modalities: SHD (Cramer et al., 2022) for spoken-digit speech recognition, MFPT (Society for Machinery Failure Prevention Technology, 2024) for bearing-fault diagnosis from sensor time series, KUL-AAD (Das et al., 2019) for EEG-based auditory-attention detection, and HalfCheetah-v4 (Todorov et al., 2012) for continuous robotic control. For each task, we retain the architecture and training configuration of the corresponding constant-weight SNN baseline and only add compact temporal weight modulation.

Table 9: Spoken-digit recognition results on SHD.

Method	Accuracy (%) ↑
DCLS-Delays (Hammouamri et al., 2024)	95.07
SpikeSCR (Wang et al., 2024a)	95.60
SE-adLIF (Baronig et al., 2025)	95.81
PfA-SNN (Sun et al., 2025)	96.26
SpikCommander (Baseline) (Wang et al., 2026)	96.41
Delta SNN	**97.00**

Table 10: Bearing-fault diagnosis results on MFPT.

Method	Accuracy (%) ↑
DRSN (ANN) (Zhao et al., 2020)	88.95 ± 1.52
CBAM (ANN) (Woo et al., 2018)	88.52 ± 0.67
MLR-SNN (Wang & Li, 2023)	88.35 ± 1.22
MS-ResNet (Hu et al., 2025)	86.73 ± 0.72
MRA-SNN (Baseline) (Zuo et al., 2025)	94.57 ± 0.82
Delta SNN	$\mathbf{96.57 \pm 0.31}$

Table 11: Auditory-attention detection results on KUL-AAD.

Method	Accuracy (%) ↑
SSF-CNN (ANN) (Cai et al., 2021)	87.80 ± 9.39
DenseNet-3D (ANN) (Xu et al., 2024)	95.90 ± 3.48
DBPNet (Ni et al., 2024)	93.66 ± 7.88
M-DBPNet (Fan et al., 2025)	93.75 ± 6.34
S^2M-Former (Baseline) (Wang et al., 2025)	94.92 ± 7.01
Delta SNN	$\mathbf{96.49 \pm 3.05}$

Table 12: Continuous-control results on HalfCheetah-v4.

Method	Average Return ↑
MDC-SAN (Zhang et al., 2022)	9147 ± 231
ILC-SAN (Chen et al., 2025)	9222 ± 615
PopSAN (Tang et al., 2021)	9594 ± 689
Proxy-Target PT-LIF (Baseline) (Xu et al., 2025b)	10103 ± 607
Delta SNN	$\mathbf{11889 \pm 589}$

Delta SNN consistently improves the corresponding constant-weight baseline across all four domains: $+0.59\%$ on SHD, $+2.00\%$ on MFPT, $+1.57\%$ on KUL-AAD, and $+1{,}786$ return ($+17.7\%$) on HalfCheetah-v4. It also achieves competitive results relative to prior work across all four benchmarks. These consistent gains demonstrate that compact temporal weight modulation is not specific to event vision and generalizes effectively across diverse temporal modalities.

C.4 Long-Range Classification

To further assess long-range classification, we evaluate Delta SNN on the Long Range Arena (LRA) benchmark (Tay et al., 2021), covering both hierarchical symbolic sequences (ListOps) and serialized images (Image).

Table 13: Classification accuracy on the Long Range Arena benchmark.

Method	ListOps (%) ↑	Image (%) ↑
Spiking LMUFormer (Liu et al., 2024)	37.3	55.7
Binary S4D (Stan & Rhodes, 2024)	54.8	82.0
P-SpikeSSM (Baseline) (Bal & Sengupta, 2025)	58.2	82.4
PRF (Huang et al., 2024)	59.2	84.8
D-RF (Zhang et al., 2025)	60.0	85.3
Delta SNN	**61.1**	**86.8**

Delta SNN improves its constant-weight P-SpikeSSM backbone by 2.9% on ListOps and 4.4% on Image, surpassing prior competitive SNN baselines. These results demonstrate consistent gains across symbolic and serialized-image classification.

D Temporal Properties and Structure

Baseline denotes the corresponding constant-weight backbone without temporal modulation, namely SpikeYOLO on Gen1 and E-SpikeFormer on HAR-DVS.

D.1 Effect of Timestep

To examine the effect of the number of timesteps, we conduct an ablation study on Gen1. We first clarify the related concepts of input time horizon, timestep, and temporal resolution. The input time horizon h is the total duration of the event stream included in one sample. A timestep is one discrete processing step of the SNN, and T denotes the number of such steps. With uniform temporal binning, the horizon is divided into T input frames, each spanning $\Delta t = h/T$. The input temporal resolution is therefore determined by Δt, with a smaller interval corresponding to finer resolution. Increasing T at a fixed horizon refines the temporal resolution, whereas increasing the horizon at a fixed T makes it coarser.

We vary T on Gen1 while fixing the input horizon at 250 ms and $D{=}1$. We use timestep-wise modulation, with one modulation state per timestep.

Table 14: Sensitivity to the number of timesteps on Gen1.

T	SpikeYOLO mAP@50/mAP@50:95 (%)	Delta SNN mAP@50/mAP@50:95 (%)	Gain
2	63.6/36.5	66.2/39.3	$+2.6/+2.8$
4	66.0/38.4	68.9/41.3	$+2.9/+2.9$
8	66.7/39.3	69.7/42.0	$+3.0/+2.7$

Across the evaluated range, increasing T improves Delta SNN from 66.2/39.3 to 69.7/42.0, while its gain over the constant-weight baseline remains stable at 2.6–3.0%. This indicates that, within the evaluated range, Delta SNN remains robust as the number of timesteps increases and the input temporal resolution becomes finer.

D.2 Effect of Input Time Horizons

Each input sample in our Gen1 configuration has a duration of 250 ms, following the protocols of SpikeYOLO (Luo et al., 2024) and MSD (Li et al., 2025) for fair comparison. We compare Delta SNN with Baseline at input horizons of 30 ms, 200 ms, 250 ms, and 500 ms under $T \times D{=}2 \times 4$ and 4×2. The 250 ms results are reproduced from Table 1.

Table 15: Performance of Baseline and Delta SNN on Gen1 across input time horizons. All mAP values are reported as percentages.

Input Horizon	$T \times D$	Baseline		Delta SNN	
		mAP@50	mAP@50:95	mAP@50	mAP@50:95
30 ms	2×4	63.0	36.1	65.2	38.1
30 ms	4×2	61.1	34.4	63.0	36.2
200 ms	2×4	66.6	39.6	68.8	42.4
200 ms	4×2	67.2	40.5	69.7	42.8
250 ms	2×4	67.0	40.1	69.1	41.9
250 ms	4×2	67.2	40.4	69.6	43.3
500 ms	2×4	65.8	39.6	67.5	41.6
500 ms	4×2	67.6	41.3	70.1	43.6

As shown in Table 15, Delta SNN consistently outperforms Baseline on both metrics across all four input horizons and both configurations, with absolute mAP@50:95 gains ranging from 1.8% to 2.9%. At 500 ms, the gains are 2.0% and 2.3% for $T \times D{=}2 \times 4$ and 4×2, respectively. With 2×4, extending the horizon from 200 ms to 500 ms reduces Delta SNN accuracy from 68.8% to 67.5% mAP@50 and from 42.4% to 41.6% mAP@50:95. Baseline also drops from 66.6% to 65.8% mAP@50. This shared decline may arise because each input frame spans a longer duration, compressing temporal information too aggressively and obscuring fine temporal structure. With 4×2, Delta SNN benefits from the longer horizon and reaches 70.1% mAP@50 and 43.6% mAP@50:95 at 500 ms.

Delta SNN remains effective at both coarse and fine input temporal resolutions. At 500 ms with $T{=}2$, each frame aggregates 250 ms of events, yet Delta SNN achieves 41.6% mAP@50:95 compared with 39.6% for Baseline. This shows that its advantage persists even at low temporal resolution. At the short horizon of 30 ms with $T{=}4$, each timestep covers only 7.5 ms of input, and Delta SNN achieves 63.0% mAP@50 and 36.2% mAP@50:95, exceeding Baseline by 1.9% and 1.8%, respectively. These results support the effectiveness of temporal modulation across different frame durations, including brief input windows with limited temporal context.

D.3 Temporal Structure of Learned Modulations

We analyze the learned effective modulation $\Delta\mathbf{W}_t = \widehat{\mathbf{M}}_t\mathbf{U}_t$ at each timestep and group pairwise similarities by temporal gap $\ell = |t - t'|$. For each weight matrix, we extract its leading singular-vector subspace and compute the average squared cosine of the principal angles between two subspaces, where higher values indicate more similar dominant directions. We demonstrate this pattern using a

representative intermediate convolutional layer on Gen1 and a linear layer on HAR-DVS, both with $T{=}8$ and a separate modulation state at each timestep.

Table 16: Temporal alignment of learned modulations.

Gap ℓ	1	2	3	4	5	6	7
Gen1	0.68	0.60	0.56	0.47	0.25	0.24	0.06
HAR-DVS	0.75	0.65	0.52	0.34	0.20	0.13	0.03

In both cases, similarity decreases with temporal distance: principal-subspace alignment drops from 0.68 to 0.06 on Gen1 and from 0.75 to 0.03 on HAR-DVS. On Gen1, this decrease corresponds to a Spearman correlation of $\rho = -1.0$. The trend is also supported by cosine similarity, which decreases from $+0.48$ and becomes negative for distant states, and by relative Frobenius distance, which increases monotonically. The learned weights therefore preserve local temporal continuity while adapting to longer-range feature changes, indicating that neighboring timesteps learn related synaptic transformations.

D.4 Modulation Granularity and Accuracy–Efficiency Trade-off

Assigning a separate modulation to every timestep provides fine temporal expressivity, but its parameter cost grows linearly with the number of timesteps. Sharing a single modulation across the sequence minimizes this cost but removes temporal variation in the synaptic weights. To balance expressivity and parameter overhead, we introduce windowed modulation, allowing neighboring timesteps to share a modulation state. The local temporal similarity observed in Appendix D.3 motivates this sharing.

In windowed modulation, a window is a contiguous group of input timesteps that share the same effective synaptic weight. We denote the number of windows by K and assign one learned modulation state to each window. Thus, K controls how often the synaptic weights can change within a sequence, while T controls the number of input processing steps. For equal windows, each window contains T/K timesteps when T is divisible by K. The case $K{=}1$ uses a single modulation throughout the sequence, $1{<}K{<}T$ shares modulation across neighboring timesteps, and $K{=}T$ assigns a separate modulation to every timestep.

We select the default number of modulation windows using a conservative budget of less than 20% additional training parameters, rather than tuning K independently for each dataset. For a base weight $\mathbf{W}_{\text{base}} \in \mathbb{R}^{m\times d}$, the default modulation width is $q = \min(m, d)/64$. One modulation state contains $q(m + d)$ matrix parameters, counting both the projection and the latent routing proxy used during training. Its overhead relative to the md base-weight parameters is bounded by

$$\frac{q(m+d)}{md} = \frac{q}{m} + \frac{q}{d} \le \frac{2}{64}.$$

With $K{=}4$, the corresponding bound for the modulation matrices is

$$\frac{Kq(m+d)}{md} \le 4 \times \frac{2}{64} = 12.5\% < 20\%.$$

Thus, $K{=}4$ provides multiple temporally distinct synaptic states while keeping the theoretical matrix-parameter overhead within the budget. The empirical sweep below shows that this choice captures most of the gain of finer modulation.

To compare globally fixed, windowed, and per-timestep modulation, we vary K while keeping all other settings unchanged.

Table 17: Sensitivity to the number of modulation states.

K	Gen1 ($T{=}8$, $D{=}1$) mAP@50/mAP@50:95 (%)	HAR-DVS ($T{=}8$, $D{=}8$) Acc. (%)	UCF AUC/FAR (%)
1	65.8/38.6	49.4	64.9/2.50
2	67.9/40.2	50.9	66.9/0.60
4	69.0/41.4	51.4	67.7/0.09
8	69.7/42.0	51.6	68.0/0.05

For Gen1 and HAR-DVS with $T{=}8$, $K{=}1$, $K{=}4$, and $K{=}8$ correspond to globally fixed, windowed, and per-timestep modulation, respectively. UCF-Crime-DVS contains over 100 event frames per sequence, so both $K{=}4$ and $K{=}8$ are windowed settings. Under the default $K{=}4$ setting, HAR-DVS uses $T{=}8$ with two timesteps per window, while UCF-Crime-DVS sequences are split into four windows.

To quantify the cost associated with modulation granularity, we measure the corresponding GPU inference costs on HAR-DVS.

Table 18: GPU inference cost under different K on HAR-DVS.

Setting	Trainable Params (M)	Infer. Lat. (ms/sample)	Infer. Mem. (GB)
Baseline	18.75	13.50	0.69
Delta SNN ($K{=}1$)	19.10 (1.02×)	13.50 (1.00×)	0.69 (1.00×)
Delta SNN ($K{=}2$)	19.45 (1.04×)	14.58 (1.08×)	0.71 (1.03×)
Delta SNN ($K{=}4$)	20.14 (1.07×)	15.31 (1.13×)	0.73 (1.06×)
Delta SNN ($K{=}8$)	21.53 (1.15×)	15.80 (1.17×)	0.78 (1.13×)

The parameter counts are reported before inference-time folding. For $K{=}1$, the globally fixed modulation can be absorbed into the backbone weight, leaving its deployed parameter storage, latency, and memory identical to Baseline.

The finest evaluated modulation performs best, but $K{=}4$ already retains 82% to 98% of the improvement from $K{=}1$ to $K{=}8$. Doubling K from 4 to 8 adds only 0.2% to 0.7% in detection, classification, and AUC scores. Meanwhile, the modulation parameters grow linearly with K. On HAR-DVS, increasing K from 4 to 8 doubles the modulation parameters from 1.39M to 2.78M, increasing total parameters, latency, and memory from $1.07\times$, $1.13\times$, and $1.06\times$ to $1.15\times$, $1.17\times$, and $1.13\times$, respectively, for only a 0.2% accuracy gain. These results support the budget-based default of $K{=}4$ as a favorable accuracy–efficiency trade-off across these event-based datasets.

This trade-off is consistent with the local correlation between adjacent event bins: assigning an independent modulation to every bin may be redundant and can fragment supervision, making optimization more sensitive to sparse or noisy events. A compact K therefore captures the major early, intermediate, and late temporal phases while sharing information among neighboring bins, thereby retaining most gains without per-timestep weights or storage growth with T.

D.5 Long-Sequence Scalability

We evaluate long-sequence efficiency on UCF-Crime-DVS using $K{=}4$ modulation windows. The dataset contains 290 test videos and 69,285 event frames, with sequence lengths up to 6,753.

Table 19: Modulation overhead on UCF-Crime-DVS.

Item	Backbone + Head	Modulation
Complexity	$\mathcal{O}(T^2)$	$\mathcal{O}(T)$
Full-test operations	41.08G	0.689G (1.68%)
Energy/inference	0.669–2.074 mJ	0.0138–0.0367 mJ (1.7–2.0%)
Parameters vs. T	Independent	Independent ($K{=}4$)

With fixed K, modulation parameters scale as $\mathcal{O}(Kq(m+d))$ and remain independent of sequence length. Modulation contributes only 1.68% of operations and 1.7–2.0% of estimated energy, while improving AUC from 65.0% to 67.7% and reducing FAR from 3.3% to 0.1%. Thus, Delta SNN remains effective with limited overhead on sequences containing hundreds or thousands of frames.

To further examine long-sequence modeling beyond event-camera-specific characteristics, we evaluate permuted MNIST. It presents the 784 pixels of each MNIST image sequentially under a fixed permutation, forming a 784-step classification task that requires long-range information integration. Delta SNN uses $K{=}4$ modulation windows, each covering 196 timesteps, so the number of modulation states remains fixed throughout the 784-step sequence.

Table 20: Long-sequence evaluation on 784-step permuted MNIST. Delta SNN uses $K=4$ modulation windows.

Method	Time-varying	Add-only	Acc. (%)
RSNN	No	No	81.07
Delta SNN	**Yes**	**Yes**	**95.18**

Compared with the static variant, Delta SNN improves accuracy by $+14\%$, confirming the benefit of temporal variation in long-sequence modeling. Overall, Delta SNN remains effective on 784-step sequences and event streams containing thousands of frames, while a fixed number of modulation windows and binary routing prevent the modulation matrices from becoming a computational bottleneck.

E Controlled Comparisons and Method Analysis

In the Gen1 comparisons below, Baseline denotes the constant-weight SpikeYOLO backbone without temporal modulation.

E.1 Controlled Comparison with Static Modulation

With the backbone, training protocol, and other factors fixed, we compare Delta SNN against parameter-matched static modulation on Gen1 with $T=2$ and $D=4$.

Table 21: Comparison of Delta SNN with static modulation on Gen1.

Method	Temporal	Width	Extra Params	mAP@50 (%)	mAP@50:95 (%)
Baseline	No	–	–	67.0	40.1
Static Modulation	No	q	$1\times$	65.7	38.9
Static Modulation	No	$2q$	$2\times$	66.6	39.1
Delta SNN	**Yes**	**q**/step	$\mathbf{2}\times$	**69.1**	**41.9**

Despite the same additional parameter budget, static modulation does not outperform the baseline, whereas Delta SNN improves it by 2.1%/1.8%. Thus, the gain arises from temporally varying transformations rather than parameter count alone. A plausible explanation is that time-varying weights enhance input-space partitioning, enabling more adaptive feature extraction. Final task performance depends on representational capacity together with optimization and regularization. Accordingly, this controlled comparison supports the practical relevance of the theoretical capacity gain, while our theoretical result does not claim a direct accuracy bound; rather, it establishes the representational advantage of time-varying weights and provides testable guidance for the modulation design.

E.2 Comparison with LoRA-Style Adaptation

Delta SNN differs from LoRA along four dimensions:

1. **Purpose.** Delta SNN is designed for highly dynamic event-based vision and overcomes the constant-weight bottleneck in SNNs through temporal synaptic adaptation, whereas LoRA (Hu et al., 2022) enables parameter-efficient adaptation of pre-trained large models.
2. **Neuromorphic execution.** LoRA is not spike-oriented, whereas Delta SNN introduces binary routing for sparse, addition-only neuromorphic inference.
3. **Theoretical basis.** LoRA assumes a low-rank update hypothesis; our theory shows that temporal modulation breaks parallel decision boundaries and combinatorially increases SNN expressivity.
4. **Training and inference.** LoRA typically freezes the backbone and yields a static adapted weight for downstream tasks; Delta SNN jointly learns base weights and temporal modulation from scratch and uses timestep-dependent weights at inference.

We construct controlled LoRA-style baselines on SpikeYOLO using Gen1 (T=2, D=4, reduction ratio 1:64). In Table 22, "Frozen" means that the backbone weights are initialized from a pre-trained constant-weight SNN and held fixed, while only the newly introduced low-rank matrices are further optimized; "Trained" means that the backbone and low-rank matrices are jointly optimized from initialization. "Temporal" indicates whether the low-rank matrices vary across timesteps (Yes) or are shared across timesteps (No). "Extra Params" is normalized to B1. For the parameter-matched variants, the rank is increased from q to $2q$ so that the total additional parameter budget matches that of Delta SNN.

Table 22: Comparison of Delta SNN with LoRA-style baselines on Gen1.

ID	Method	Base $\mathbf{W}$	Temporal	Width	Extra Params	mAP@50 (%)	mAP@50:95 (%)
–	Baseline	Trained	No	–	–	67.0	40.1
B1	LoRA-Frozen (same width)	Frozen	No	q	1×	65.1	39.5
B2	LoRA-Frozen (parameter matched)	Frozen	No	$2q$	2×	65.7	39.2
B3	LoRA-Joint (same width)	Trained	No	q	1×	65.7	38.9
B4	LoRA-Joint (parameter matched)	Trained	No	$2q$	2×	66.6	39.1
B5	Delta-Frozen	Frozen	Yes	q/step	2×	65.2	39.2
–	**Delta SNN (Ours)**	**Trained**	**Yes**	$\mathbf{q}$/step	**2×**	**69.1**	**41.9**

All four LoRA-style variants underperform Baseline across different backbone training strategies and rank settings, even with full-precision low-rank matrices. Delta-Frozen (B5) also underperforms the baseline, indicating that temporal variation alone is insufficient when the pre-trained constant-weight backbone remains frozen. Among the LoRA-style variants, B4 achieves the highest mAP@50 (66.6%), while B1 achieves the highest mAP@50:95 (39.5%); Delta SNN exceeds these respective best scores by 2.5% and 2.4%. Within the parameter-matched subset (B2 and B4), B2 has the higher mAP@50:95 (39.2%), which Delta SNN surpasses by 2.7%.

E.3 Comparison with Alternative Temporal Adaptation Mechanisms

We compare Delta SNN with representative temporal adaptation baselines from both SNNs and ANNs on Gen1 with the SpikeYOLO backbone (T=4, D=1). These methods adapt neuronal dynamics, normalization, firing representations, or convolutional weights.

- **PLIF.** Parametric Leaky Integrate-and-Fire neurons (Fang et al., 2021b) learn the membrane time constant jointly with synaptic weights, allowing each layer to adjust the balance between retaining past membrane states and integrating current inputs. The learned time constant is shared across timesteps, so adaptation occurs in neuronal dynamics while the synaptic weights remain constant over time.
- **TEBN (channel-wise).** We use a channel-wise variant of Temporal Effective Batch Normalization (Duan et al., 2022), adding a learnable scaling matrix $\mathbf{P} \in \mathbb{R}^{T\times C}$ at each normalization layer, with one coefficient per timestep and channel while sharing normalization statistics and affine parameters across timesteps. The scaling coefficients are initialized to one and broadcast over the batch and spatial dimensions, extending the channel-shared temporal scaling in the authors' implementation. Across 123 normalization layers, this adds $\sum_l TC_l = 199{,}424$ parameters, accounting for the approximately 0.2M increase in Table 24.
- **PIT.** Parametric Invertible Transformation (Yan et al., 2026) applies a time-varying transformation and its inverse around the firing function g, yielding $\mathbf{s}_t = \mathbf{A}_t g(\mathbf{A}_t^{-1}\mathbf{u}_t)$. The method uses diagonal transformation matrices to adapt firing representations across channels and timesteps, together with initialization based on membrane-potential statistics and a rectified surrogate gradient.
- **TAda-Learnable-T.V.** This baseline follows the learnable, temporally varying calibration variant in Table 6 of Huang et al. (2022), using separate learned calibration coefficients at different temporal locations to modulate a shared convolutional kernel. It takes the form $\mathbf{W}_t = \boldsymbol{\alpha}_t \odot \mathbf{W}_{\text{base}}$, where channel calibration coefficients are broadcast over the kernel and learned directly rather than generated from the current input. This is an ablation variant of TAdaConv, providing multiplicative temporal weight calibration for comparison with the additive modulation in Delta SNN.
- **TAdaConv.** Temporally-Adaptive Convolutions (Huang et al., 2022) use local and global temporal context to generate input-channel calibration coefficients, yielding $\mathbf{W}_t = \boldsymbol{\alpha}_t(\mathbf{x}) \odot \mathbf{W}_{\text{base}}$. Our

implementation uses a RouteFuncMLP with reduction ratio r=4, two temporal convolutions with kernel size 3, and a global-context projection, producing channel coefficients rather than full convolutional kernels. Calibration is applied to 87 convolutions with groups=1, with one base convolution retained at each location; the 38 depthwise convolutions remain unchanged.

Experimental Setup. All experiments use a single NVIDIA H200 GPU, with mAP@50:95 as the primary detection metric. Training settings follow those of Delta SNN under T=4, D=1, except that TEBN uses a batch size of 64 to fit GPU memory. For the training-cost comparison, all methods are measured separately at a common batch size of 64. PIT uses gradient checkpointing in its firing loop to recompute activations during backpropagation, reducing peak memory at the expense of additional computation. We abbreviate TAda-Learnable-T.V. as TAda-TV in the tables.

Table 23: Detection accuracy of alternative temporal adaptation mechanisms on Gen1 (T=4, D=1).

Method	mAP@50:95 (%)	mAP@50 (%)
Baseline	38.4	66.0
PLIF	38.9	66.4
TEBN (channel-wise)	38.3	66.3
PIT	39.0	66.6
TAda-TV	38.9	66.4
TAdaConv	37.3	64.7
Delta SNN	41.3	68.9

Table 24: Training cost on Gen1 (T=4, D=1), measured on a single NVIDIA H200 GPU with batch size 64 for every method. PIT uses gradient checkpointing.

Method	Params. (M)	Peak Train Mem. (GiB)	Time/Epoch (s)
Baseline	23.1	55.0	806
PLIF	23.1	65.1	848
TEBN (channel-wise)	23.3	97.0	1100
PIT	23.3	44.0	1286
TAda-TV	23.3	64.3	926
TAdaConv	70.2	64.7	914
Delta SNN	24.9	56.3	854

TAdaConv's routing networks contribute approximately 47.1M parameters to the 70.2M total in Table 24, including 18.8M in the global $C \to C$ projection and 28.3M in the two temporal projections and their normalization parameters. This overhead comes from the routing networks attached to the calibrated layers, with no duplicate base convolutions included in the parameter count.

As shown in Tables 23 and 24, Delta SNN outperforms all compared baselines on both detection metrics. Its additional parameter, peak-memory, and training-time costs over Baseline remain modest, providing a favorable balance between detection accuracy and computational overhead.

E.4 Backbone-Controlled Evaluation

We use strong open-source baselines and keep their architectures and training settings unchanged. Delta SNN only replaces constant weights with compact time-varying modulation. To test whether the gains depend on the selected backbone, we integrate Delta SNN into E-SpikeFormer (Yao et al., 2025) and the similarly sized SpikingFormer (Zhou et al., 2023a) on HAR-DVS.

Table 25: Backbone-controlled comparisons on HAR-DVS.

Method	Params (M)	Top-1 (%)
E-SpikeFormer (Yao et al., 2025)	18.8	49.2
E-SpikeFormer + Delta SNN	**20.1**	**51.4**
SpikingFormer (Zhou et al., 2023a)	16.6	47.9
SpikingFormer + Delta SNN	**18.0**	**50.1**

Delta SNN improves both E-SpikeFormer and SpikingFormer by 2.2%. The controlled comparison with similarly sized backbones shows that the gains are method-specific rather than backbone-specific.

E.5 Generalization Across Neuron Models

Delta SNN has no implicit dependency on Integer-LIF and is directly applicable to standard LIF neurons. In SpikeYOLO (Luo et al., 2024), D denotes the upper bound of integer-valued activations; when $D{=}1$, Integer-LIF is mathematically equivalent to standard LIF. Therefore, the Gen1 experiments with $T{=}4$ and $D{=}1$ in Table 1 already demonstrate the effectiveness of Delta SNN with standard LIF neurons. Moreover, our video anomaly detection experiments on UCF-Crime-DVS use standard LIF neurons throughout, further demonstrating that the proposed temporal modulation generalizes beyond Integer-LIF. To provide an additional controlled evaluation, we test SpikeYOLO and Delta SNN on Gen1 with $T{=}2$ and $D{=}1$. As shown in Table 26, Delta SNN consistently improves both detection metrics under the standard LIF setting.

Table 26: Performance on Gen1 with $T{=}2$ and $D{=}1$ (standard LIF).

Model	mAP@50 (%)	mAP@50:95 (%)
SpikeYOLO	63.6	36.5
Delta SNN	**66.2** (+2.6%)	**39.3** (+2.8%)

F Hardware-Aware Efficiency and Neuromorphic Validation

F.1 Analytical Energy Model

To provide an architecture-level estimate that is independent of a specific hardware implementation, we retain the standard analytical protocol widely used in the neuromorphic community (Horowitz, 2014; Yao et al., 2023b; 2024; 2025; Luo et al., 2024). This protocol distinguishes Multiply-and-Accumulate (MAC) operations from Accumulate (AC) operations and enables a consistent comparison between ANNs and SNNs.

Following this protocol, the energy consumption of an ANN is calculated as:

$$E_{\mathrm{ANN}} = FL \cdot E_{\mathrm{MAC}}, \tag{10}$$

where FL denotes the total number of floating-point operations (FLOPs), and $E_{\mathrm{MAC}} = 4.6\,\mathrm{pJ}$ represents the energy cost of a single MAC operation in 45 nm CMOS technology (Horowitz, 2014).

For both the baseline constant-weight SNN and Delta SNN, the energy consumption of the l-th layer is formulated as:

$$E_{\mathrm{SNN}}^{(l)} = FL_l \cdot E_{\mathrm{AC}} \cdot fr_l \cdot T, \tag{11}$$

where FL_l is the FLOP count of the l-th layer, fr_l is the average firing rate, and T is the number of time steps. $E_{\mathrm{AC}} = 0.9\,\mathrm{pJ}$ denotes the energy cost of an AC operation (Horowitz, 2014). The total energy consumption is obtained by summing $E_{\mathrm{SNN}}^{(l)}$ across all layers.

Table 27: Per-operation energy constants used by the analytical model.

Operation	Energy	Technology
MAC	4.6 pJ	45 nm CMOS
AC	0.9 pJ	45 nm CMOS

To quantify FL, the operation counts for convolutional and linear layers are defined as:

$$FL_{\text{Conv}} = \frac{k^2 \cdot h_{\text{out}} \cdot w_{\text{out}} \cdot c_{\text{in}} \cdot c_{\text{out}}}{g}, \tag{12}$$

where k is the kernel size, $(h_{\text{out}}, w_{\text{out}})$ are the spatial dimensions of the output feature map, c_{in} and c_{out} are the input and output channels, and g is the group number. For a linear layer:

$$FL_{\text{Linear}} = d_{\text{in}} \cdot d_{\text{out}}, \tag{13}$$

where d_{in} and d_{out} are the input and output dimensions. This analytical model captures the compute-side advantage of sparse, event-driven execution and serves as the basis for the operation-level breakdown below.

F.2 Parameter and Operation Breakdown

To directly quantify the overhead introduced by temporal modulation, we decompose the parameters, MACs, ACs, and energy of Delta SNN on Gen1 with T=4 and D=1. The spike-driven backbone, the full-precision projection $\overline{\mathbf{U}}_t$, and the binary routing matrix $\mathbf{M}_t$ are reported separately, using the inference-time reparameterization in Eq. (5).

Table 28: Parameter and energy breakdown on Gen1 (T=4, D=1).

Component	Params (M)	Param. Share	Energy (mJ)	Energy Share
Spike-driven backbone (Conv+Linear)	23.12	93.0%	11.51	99.8%
Projection $\overline{\mathbf{U}}_t$	0.755	3.0%	0.023	0.20%
Binary routing $\mathbf{M}_t$	0.995	4.00%	0.0002	0.002%
Delta SNN total	**24.87**	**100%**	**11.53**	**100%**

After binary routing is applied, the deployed modulation adds only 0.20% inference energy, introduces no MACs, and preserves fully addition-only execution. The spike-driven backbone remains responsible for 99.8% of the estimated energy, while the projection and binary routing components together contribute only a small fraction of the total. This residual overhead is measured within the binary-routing model; the energy saved by replacing a separate dense float-routing variant is analyzed in subsection F.3.

F.3 Effect of Binary Synaptic Routing

Although compact temporal modulation reduces parameter complexity, a floating-point routing matrix would still introduce dense matrix multiplications. We therefore ablate the binarization of $\mathbf{M}_t$ to isolate its effect on routing operations and energy. The float-routing variant uses dense routing, whereas the inference form of Delta SNN uses the sparse 1-bit mask $\mathbf{M}_t$ for mask-based selection and accumulation.

Table 29: Effect of routing binarization.

Routing $\mathbf{M}_t$	Operation Type	Routing Energy (mJ)	Total Energy (mJ)
Float (dense)	MAC (0.63G)	2.88	14.5
Binary (ours)	**AC (mask-add)**	**0.0002**	**11.5** (-20.7%)

Binarization improves efficiency in two ways. First, it replaces 0.63G dense floating-point MACs with sparse mask-add operations. Second, the 1-bit routing mask is highly sparse, so only a few paths are activated, reducing routing energy from 2.88 mJ to 2×10^{-4} mJ. Replacing 32-bit routing weights with 1-bit representations also reduces bandwidth and enables bitwise routing. Relative to the float-routing Delta SNN (14.5 mJ), binarization reduces total energy by 20.7% without accuracy loss. In contrast, the 0.20% reported in subsection F.2 is the residual modulation overhead after binarization; the two figures therefore use different reference models and are consistent. This result is also consistent with the accuracy ablation in Fig. 4.

F.4 Lava-Based Neuromorphic Functional Validation

To provide a more neuromorphically compatible simulation and cost analysis, we implement the Delta convolutional and linear modules in Intel's Lava framework and run a Loihi-2-compatible functional simulation.

Each Delta convolutional or linear module is implemented as a native Lava *Process*, with a custom *ProcessModel* specifying spike-driven, addition-only execution. We use *Loihi2SimCfg* as the *RunConfig* to select the Loihi-2-compatible simulation model and compare the outputs directly with the corresponding PyTorch modules.

Table 30: Numerical equivalence between PyTorch and Lava implementations.

Delta Module	Maximum Absolute Error
Convolution	5.1×10^{-7}
Linear	1.2×10^{-7}

The Lava outputs match their PyTorch counterparts to floating-point precision. This functional equivalence verifies that Delta modulation can be faithfully represented by a Loihi-2-compatible execution graph through binary routing and spike-triggered accumulation without introducing MAC operations. These results establish functional simulation compatibility.

F.5 Hardware-Aware Memory Access and Latency

Because compute-only estimates omit the dominant cost of data movement and storage access, we further measure routing cost, storage-access overhead, compute, and latency for the Lava simulation. The hardware-aware estimates use a bit-width-scaled 45-nm energy model, with the memory read/write energy model following SpikedAttention (Hwang et al., 2024). Energy results are reported for a representative convolutional module, while complete-network GPU latency is measured on Gen1 with T=4 and D=1.

Table 31: Energy and latency of a representative Delta-Conv module in Lava simulation.

Metric	Baseline Conv	Delta-Conv	Overhead
Compute – synaptic AC (μJ)	754.85	756.21	+0.18%
Data movement (μJ)	131.62	131.86	+0.18%
Storage access – weight R/W (μJ)	4193.61	4200.90	+0.17%
Routing cost (μJ)	0	0.009	–
LIF neuron update – shared (μJ)	60.19	60.19	+0%
Total energy (μJ)	5140.27	5149.15	1.002×
Algorithmic timesteps	4	4	1.00×
Pipeline depth	2	3	+1

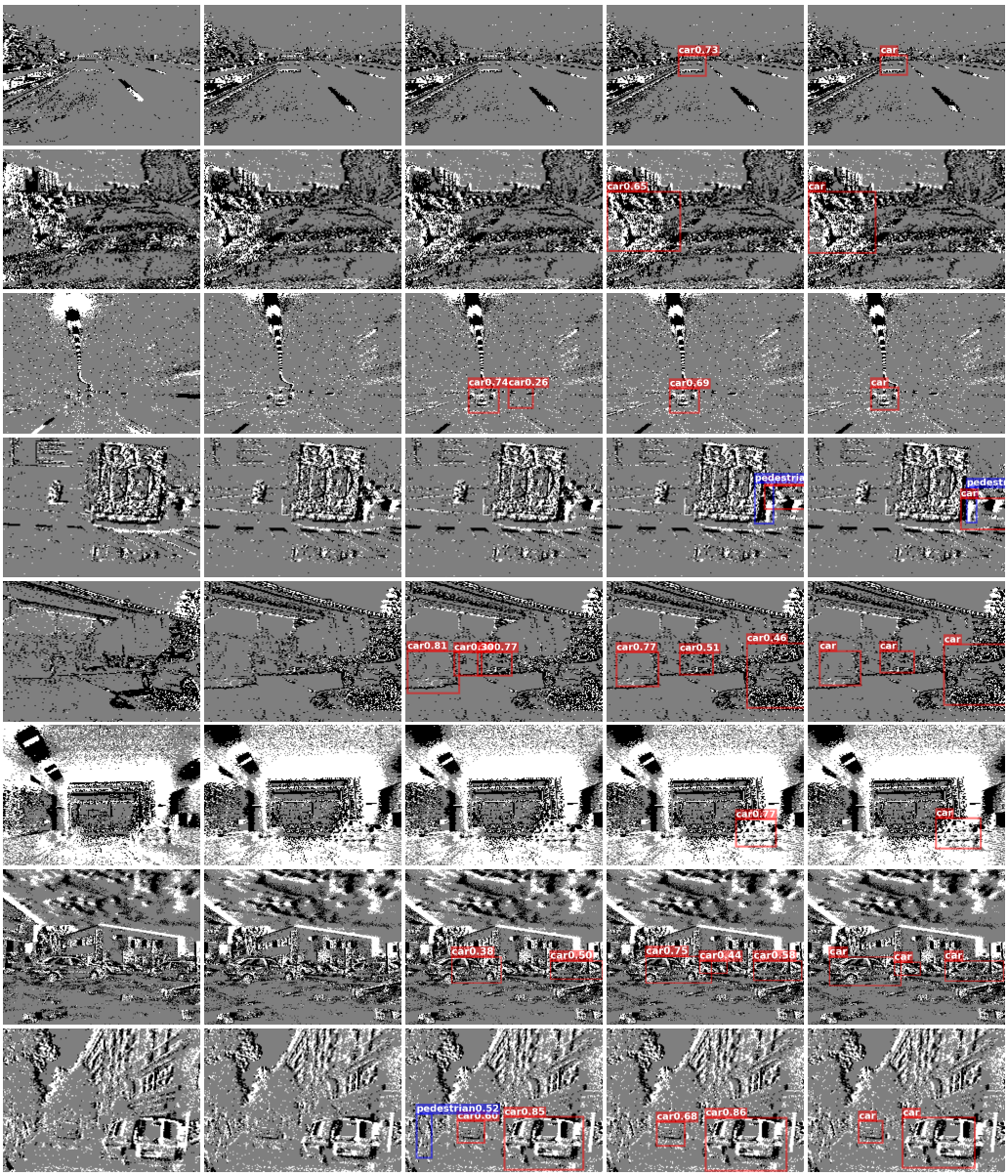
Storage access dominates at approximately 82% of total energy, yet Delta adds only 0.17% to this cost. The routing matrix $\mathbf{M}_t$ is 1-bit, 32× cheaper to read than a 32-bit weight, and sparse; consequently, its routing cost is only 0.009 μJ. Lava-reported algorithmic latency remains unchanged at T=4. The projection $\overline{\mathbf{U}}_t$ increases pipeline depth from 2 to 3, introducing only a minor startup delay without affecting steady-state throughput.

The representative Delta-Linear module similarly incurs only 1.04% memory overhead and $1.010\times$ total energy, confirming low hardware-aware overhead for both convolutional and linear layers. At the complete-network level, Gen1 GPU inference latency increases from 28.44 ms to 30.66 ms ($1.08\times$), with approximately 0.3 GB additional memory.

G LIMITATION

Due to limited resources, we leave application of our method to language models and other domains for future work. While current deployments are limited to GPUs, we anticipate that its advantages will be more pronounced when implemented on neuromorphic hardware.

H ADDITIONAL VISUAL COMPARISON OF PREDICTION RESULTS



(a) Input frame 1 (b) Input frame 2 (c) Constant-Weight SNN (d) Delta SNN (e) Ground truth

Figure 7: Additional qualitative comparison of detection results on the event-based Gen1 dataset. In challenging scenarios such as occlusion, high dynamic range, small objects, and fast motion, Delta SNN consistently achieves more precise detection results with significantly fewer false detections.