EAS-SNN: End-to-End Adaptive Sampling and Representation for Event-based Detection with Recurrent Spiking Neural Networks

Ziming Wang[1], Ziling Wang[1], Huaning Li[1], Lang Qin[1], Runhao Jiang[1], De Ma[1,2] ✉, and Huajin Tang[1,2] ✉

[1] College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
[2] State Key Laboratory of Brain-Machine Intelligence, Zhejiang University, Hangzhou 310027, China. ✉ Corresponding author.
{zi_ming_wang,wzzero.hhh,lhnkevin1014,qinl,RhJiang,made,htang}@zju.edu.cn

Abstract. Event cameras, with their high dynamic range and temporal resolution, are ideally suited for object detection in scenarios with motion blur and challenging lighting conditions. However, while most existing approaches prioritize optimizing spatiotemporal representations with advanced detection backbones and early aggregation functions, the crucial issue of adaptive event sampling remains largely unaddressed. Spiking Neural Networks (SNNs), operating on an event-driven paradigm, align closely with the behavior of an ideal temporal event sampler. Motivated by this, we propose a novel adaptive sampling module that leverages recurrent convolutional SNNs enhanced with temporal memory, facilitating a fully end-to-end learnable framework for event-based detection. Additionally, we introduce Residual Potential Dropout (RPD) and Spike-Aware Training (SAT) to regulate potential distribution and address performance degradation encountered in spike-based sampling modules. Empirical evaluation on neuromorphic detection datasets demonstrates that our approach outperforms existing state-of-the-art spike-based methods with significantly fewer parameters and time steps. For instance, our method yields a 4.4% mAP improvement on the Gen1 dataset, while requiring 38% fewer parameters and only three time steps. Moreover, the applicability and effectiveness of our adaptive sampling methodology extend beyond SNNs, as demonstrated through further validation on conventional non-spiking models. Code is available at Github.

Keywords: Event-based Vision · Spiking Neural Network · Neuromorphic Computing

1 Introduction

Event cameras, inspired by biological vision systems, present a transformative approach to dynamic visual sensing. Diverging from the conventional imaging model, where cameras capture scene illumination synchronously via a global

shutter, event cameras operate on an asynchronous basis, logging logarithmic intensity changes at the pixel level. Each significant change in this context is referred to as an "event". This distinct operational paradigm grants event cameras extraordinary capabilities: exceptional temporal resolution (>10k fps), high dynamic range (>120 dB), and remarkably low power consumption (<10 mW) [17, 18, 40, 55]. These attributes render event cameras exceptionally suited for object detection under challenging environments characterized by rapid motion, extreme lighting conditions, or stringent energy constraints.

However, the inherent asynchronous and sparse characteristics of data captured by event cameras present significant challenges for traditional dense neural networks, which are primarily designed to process synchronous, uniformly structured data. To bridge the asynchronous event stream with the dense tensor formats utilized in ANNs, most existing frameworks incorporate a sampling-aggregation mechanism, also referred to as event representation [4, 20, 21, 33, 41] or event embedding. This integration effectively alleviates the computational demands engendered by the high temporal resolution of event data. Building upon this foundation, researchers have delved into exploiting the unique spatiotemporal attributes of event data through refining event representations [9,34,35,37,57] and enhancing the design of detection architectures [21, 25, 35, 44, 50, 54, 58], thereby boosting network efficacy. Additionally, the aggregation module is integrated with downstream feature extraction components into a unified end-to-end training framework [20]. Regarding event-based sampling, several studies have endeavored to enhance the sampling module from a rule-based or search-based perspective [6, 35, 41, 75]. Nonetheless, these initiatives often suffer from the lack of high-level semantic information or are hindered by artificial constraints and suboptimal search efficiency. So far, the quest for a differentiable, adaptive sampling module that facilitates end-to-end optimization within event-based detection frameworks remains unfulfilled.

On the other hand, SNNs, composed of biologically plausible spiking neurons, support asynchronous event-driven computation on neuromorphic hardware [1, 10, 48] and so present a high potential for fast inference and low power consumption. This inherent compatibility with event data makes SNNs ideally suited for event-based vision tasks. Moreover, a spectrum of low-level representations [20,34,57] can be encapsulated within a unified SNN layer [31]. Nevertheless, the challenge of training deep SNNs under long time steps persists, despite rapid advancements in spike-based learning algorithm [16, 38, 56, 60, 64, 68, 74]. This makes it difficult to align the discretization window of SNNs with event response time, culminating in temporal information loss. Consequently, the current event-based detection models based on SNNs exhibit a notable performance gap when compared to their ANN counterparts.

To address these challenges, this work introduces ARSNN, a learnable adaptive sampling methodology with recurrent convolutional SNNs (Fig. 1a), enabling a comprehensive end-to-end learning paradigm for event representation and feature extraction. This methodology utilizes pixel-wise spiking neurons in the sampling module to capture the dynamic information delivered by events

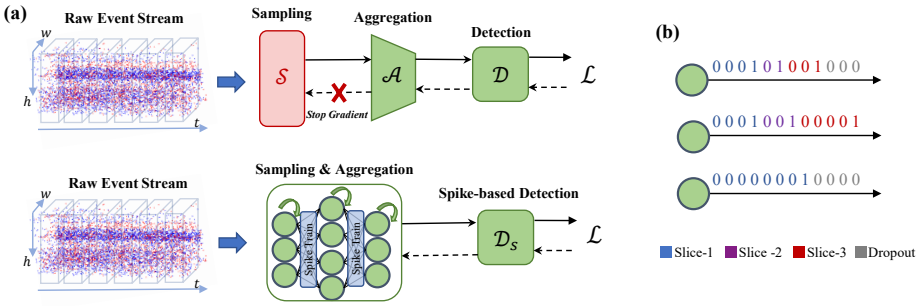


Fig. 1: *Left:* Implementing the end-to-end optimization for event-based sampling via recurrent SNNs. The typical rule-based methods truncate backpropagation, whereas our adaptive sampling module ensures continuous gradient flow. *Right:* The variability and heterogeneity in sampling time and count among different spiking neurons.

across receptive fields. Upon the information integration surpassing a predefined threshold, the events between the last and the current spike firing times are sampled as shown in Fig. 1b. By optimizing the spike firing time and membrane potential distribution, the sampling module can be trained and coordinated with downstream spike-based detectors, thereby greatly mitigating the temporal alignment loss between the raw event stream and SNNs. The main contributions are outlined as follows:

1. We first propose a learnable adaptive sampling method and implement an end-to-end event-based detection framework via SNNs, integrating the processes of sampling, aggregation, and feature extraction.
2. We develop the techniques of Residual Potential Dropout (RPD) and Spike-Aware Training (SAT) to mitigate performance degradation in SNN-based sampling modules with the recurrent convolutional architecture.
3. Empirical evaluations on neuromorphic datasets (N-Caltech 101, Gen1, and 1Mpx) demonstrate that the proposed method outperforms state-of-the-art spike-based methods with 5.85× less energy consumption compared to ANNs. Furthermore, the effectiveness of our adaptive sampling method is verified across dense detection models.

2 Related Work

2.1 Spiking Neural Networks

Spiking Neural Networks (SNNs) have raised substantial interest due to their asynchronous computation paradigm and promising energy efficiency. However, optimizing SNNs poses challenges due to the binary nature of spike firing. The advancements in backpropagation techniques for SNNs, including time-based [45, 46, 69, 70, 73] and activation-based methods [8, 23, 38, 56, 60, 64, 68], estimate gradients for firing time and spike generation successfully. Additionally,

various spiking neuron models [7,16,24,27,39,66] have been proposed to enhance network heterogeneity and performance. To maintain gradient flow in deep SNNs, specialized network modules [13, 16, 71] and architectures [15, 30, 58, 65, 72] have been developed considering temporal dynamics in spiking neurons. These collective efforts are narrowing the performance gap between SNNs and ANNs in various tasks, including object detection [58,61,67], action recognition [12,22,42], point cloud recognition [53], and automatic speech recognition [63].

2.2 Event-based Detection

In event-based detection, methodologies are bifurcated into two principal categories: dense and sparse representations. Dense representations typically convert event data into frame-based formats, subsequently processed by conventional detection algorithms [4, 5, 21, 33, 41]. This approach, while facilitating the use of pre-trained models within an event-based context [43], compress the inherent spatiotemporal richness of events, necessitating sophisticated event-to-frame conversion techniques. To extract the spatiotemporal information more effectively, researchers have integrated the recurrent connections [21, 35, 50] and attention mechanisms [36, 59]. Recent advancements [21, 25, 41] strive to minimize detection latency without substantial performance degradation, through the strategic redesign of network backbones and event aggregation modules.

Conversely, sparse representations, embodied by Graph Neural Networks (GNNs) and SNNs, offer substantial energy efficiency due to their inherent sparse computational models. GNN models [37, 54] construct dynamic spatio-temporal graphs from event stream, updating node and edge states with incoming events. In parallel, SNN methods [9, 58, 67], characterized by the internal temporal dynamics and spike-driven computation, naturally complement the asynchronous nature of event stream. However, given the high temporal resolution of event cameras, event-based sampling is imperative to mitigate computational demands and enhance network generalization capabilities, even for sparse representation.

For event-based sampling, some works have emerged to refine rule-based strategies by considering the instantaneous rate of event generation [32, 35, 41]. Nevertheless, these strategies frequently encounter challenges associated with hyperparameter sensitivity and the difficulty of maintaining an optimal balance between sparse and ambiguous textures in complex scenarios. Nikola *et al.* [75] explored the optimization of sampling windows based on the Gromov-Wasserstein discrepancy from raw event data to their representations. Nonetheless, this technique has yet not fully integrated high-level semantic guidance. Cao *et al.* [6] proposed using SNNs for global event triggering, advocating for a feedback-update mechanism that employs pseudo labels generated by downstream detectors to guide SNN training. Despite these advancements, the integration of ANNs and SNNs remains disjointed, bridged solely by pseudo labels, which introduces artificial constraints on the optimization landscape and compromises search efficiency. In this work, we investigate an adaptive event sampling strategy leveraging learnable SNNs, marking a step towards achieving comprehensive end-to-end event-based representation learning.

3 Methodology

3.1 Event-based Detection

Consider an event stream denoted as $\mathbf{E} = \{(x_i, y_i, p_i, t_i)\}_{i\in\mathbb{N}}$, where each event e_i is characterized by its spatial coordinates $x_i \in [1, W]$ and $y_i \in [1, H]$, a polarity value $p_i \in \{-1, 1\}$ indicating the change in pixel intensity, and a timestamp $t_i \in [0, +\infty)$ marking the occurrence time of events (typically measured in microseconds). In event-based detection, annotations at a specific target time t include object categories l_j and associated ground truth bounding boxes (x_j, y_j, w_j, h_j), encapsulated as $\mathbf{B}_t = \{(x_j, y_j, w_j, h_j, l_j, t)\}_{j\in\mathbb{N}}$. Ideally, a real-time detector $\mathcal{D}$ is capable of delivering predictions at the timestamp level:

$$\mathcal{D}(\mathbf{E}) = \{\mathcal{D}(\mathbf{E}, t)\}_{t\geq 0} = \left\{\mathcal{D}\left(\{e_i\}_{t_i<t}\right)\right\}_{t\geq 0} \tag{1}$$

Nonetheless, given the high temporal resolution of event cameras, it is challenging for event-based object detectors to process and interpret $10^4 - 10^5$ times per second. Moreover, the necessity for detecting moving objects at microsecond intervals is often not critical, as object motion typically occurs at a slower pace than the pixel response rate of event cameras [50]. Consequently, most works [21, 35, 58, 67] frequently incorporate a sampling operation $\mathcal{S}$ and an aggregation operation $\mathcal{A}$ to reduce the volume of sparse and asynchronous event data, thereby facilitating more efficient processing prior to the detection phase:

$$\hat{\mathcal{D}}\left(\mathbf{E}, t\right) = (\mathcal{D} \circ \mathcal{A} \circ \mathcal{S})(\mathbf{E}, t) = (\mathcal{D} \circ \mathcal{A} \circ \mathcal{S})(\{e_i\}_{t_i<t}) \tag{2}$$

By employing fixed window sampling as an example, $\mathcal{S}$ can be expressed as:

$$\mathcal{S}_{\Delta t,T}(\{e_i\}_{t_i<t}) = \Omega_{\Delta t,T} = \left\{\{e_i\}_{t-(j+1)\Delta t\leq t_i<t-j\Delta t}\right\}_{j\in[0,T/\Delta t]} \tag{3}$$

where T denotes the global window for sampling, and Δt represents the interval of sampling slices. We designate the index set that meets the criteria for fixed window sampling within the j-th step as $\mathbf{S}_j$. For the j-th step sampling, a typical aggregation function, $\mathcal{A}^j$, based on event count histograms, is defined as follows:

$$\mathcal{A}^j(\{e_i\}_{i\in\mathbf{S}_j}) = f_j(x, y, p) = \sum_{i\in\mathbf{S}_j} \delta\left(x - x_i, y - y_i, p - p_i\right) \tag{4}$$

Here, $\delta(x, y, z)$ represents the Dirac delta function, which equals 1 only when $x = y = z = 0$. By employing this approach, the asynchronous and sparse event data are effectively down-sampled into a compact form that significantly alleviates the computational burden on the detector. However, existing sampling techniques often operate independently of downstream object detectors , necessitating extensive hyperparameter tuning and showing limited adaptability.

3.2 Reccurent Convolutional SNNs for Adaptive Event Sampling

In this study, our goal is to develop a learnable sampling module (Fig. 2), denoted as $\mathcal{S}_{\text{SNN}}$, utilizing SNNs to address the challenge of non-differentiable, end-to-end learning that constrains rule-based sampling methods.

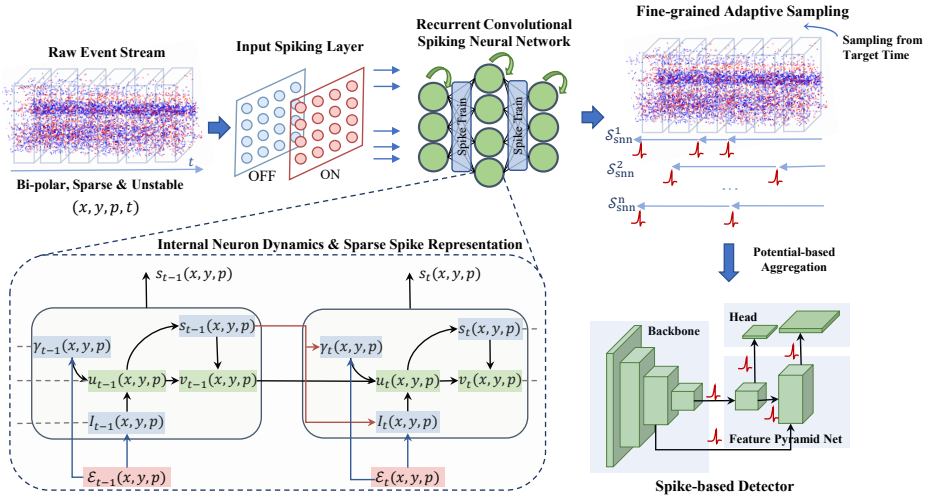


Fig. 2: Illustration of the overall pipeline. The adaptive sampling module, augmented with recurrent synaptic connections for improved memory mechanism, bridges the input event stream with the downstream spike-based detector. Within the unrolled computation graph, internal neuron dynamics are indicated by solid black lines, while blue and red solid lines depict the feedforward and recurrent pathways, respectively.

The Leaky-Integrate-and-Fire (LIF) model, popular in neuromorphic computing due to its minimal computational requirements. The discrete LIF equations, via the first-order Euler method with a time slice of Δt_s, are given by:

$$
\begin{aligned}
u_i^\ell[t] &= g\left(v_i^\ell[t-1], I_i^\ell[t]\right) = \left(1 - \frac{\Delta t_s}{\tau_m}\right) v_i^\ell[t-1] + \frac{\Delta t_s}{\tau_m} I_i^\ell[t] \\
s_i^\ell[t] &= \Theta\left(u_i^\ell[t] - \theta\right) = \begin{cases} 1, & u_i^\ell[t] \geq \theta, \\ 0, & u_i^\ell[t] < \theta \end{cases} \\
v_i^\ell[t] &= \begin{cases} u_i^\ell[t] - \theta \cdot s_i^\ell[t], & \text{soft reset} \\ u_i^\ell[t] \cdot \left(1 - s_i^\ell[t]\right) + u_{\text{reset}} \cdot s_i^\ell[t], & \text{hard reset} \end{cases}
\end{aligned}
\tag{5}
$$

Here, $u_i^\ell[t]$ and $I_i^\ell[t]$ denote the membrane potential and input current of the i-th neuron in the l-th layer, respectively, with τ_m as the membrane time constant. Upon receiving spike inputs, a neuron accumulates the input current $I_i^\ell[t]$ into its membrane potential $u_i^\ell[t]$, which decays at a constant rate. When the membrane potential $u_i^\ell[t]$ exceeds the threshold θ, a spike is emitted, resetting $u_i^\ell[t]$ to $v_i^\ell[t]$. The LIF dynamic closely aligns with the desired properties of event-driven sampling, where events should be selectively sampled as the accumulated information from the asynchronous event stream reaches a threshold. Consequently, we propose to associate spike firing in SNNs with event-based sampling, integrating this functionality into an end-to-end optimization framework.

Firstly, restricted by the poor support for sparse spike-driven computation on GPUs, we perform early aggregation using event count as shown in Eq. (4)

with a small time slice Δt_m and obtain the actual T_m-step inputs for $\mathcal{S}_{\text{SNN}}$. Following early aggregation, a distinct spiking neuron is allocated to each spatial position (x, y) and polarity p, with each neuron exhibiting unique behavior by accumulating its spatially localized input over time:

$$\begin{aligned} I_t(x,y,p) &= (W_{\text{conv}} * f_t)(x,y,p) + b_{\text{conv}}(p) \\ u_t(x,y,p) &= g(I_t(x,y,p), v_t(x,y,p)) = \gamma \cdot v_t(x,y,p) + I_t(x,y,p) \end{aligned} \tag{6}$$

where $\gamma = 1 - \frac{\Delta t_s}{\tau_m}$ denotes a constant decay factor. Our empirical findings suggest that an effective sampler must possess an advanced temporal credit assignment capability to extract informative and discriminative substreams from the inherently noisy and fluctuating spatiotemporal event data. Therefore, we incorporate recurrent synaptic connections, as depicted in the bottom left portion of Fig. 2, to enhance the temporal representational capacity of SNNs:

$$\begin{aligned} u_t(x,y,p) &= \gamma_t(x,y,p) \cdot v_{t-1}(x,y,p) + I_t(x,y,p) \\ I_t(x,y,p) &= (W_I^f * f_t)(x,y,p) + (W_I^s * s_{t-1})(x,y,p) + b_I(p) \\ \gamma_t(x,y,p) &= \sigma\left((W_\gamma^f * f_t)(x,y,p) + (W_\gamma^s * s_{t-1})(x,y,p) + b_\gamma(p)\right) \end{aligned} \tag{7}$$

Here, the spike firing $s_{t-1}(x, y, p)$ at the last time step contributes to the input current $I_t(x, y, p)$ and decay factor $\gamma_t(x, y, p)$ at the current time step directly through recurrent synapse W_I^s and W_γ^s. σ denotes the sigmoid function which normalizes the decay factor γ_t. We refer to SNNs after this modification as "RSNN". When the spike is emitted with over-threshold membrane potential, the event slice between the last spike firing time $t^{k-1}(x, y, p)$ and the current spike firing time $t^k(x, y, p)$ is sampled, so as to achieve local sampling:

$$\begin{aligned} \Omega^k(x,y,p) = \left\{(x_i, y_i, p_i, t_i) \mid t_i \in (t^{k-1}(x,y,p), t^k(x,y,p)] \& [x,y,p] = [x_i, y_i, p_i]\right\} \\ \text{where} \quad s_{t^k}(x,y,p) = 1 \ \& \ t^{k-1}(x,y,p) < t^k(x,y,p) \end{aligned} \tag{8}$$

This approach optimizes event sampling by converting it into learning proper firing times across spiking neurons at distinct spatial coordinates. After event-based sampling, the remained challenge is to effectively integrate each sampled set into a coherent embedding. Empirically, we find that decoupling the sampling and the aggregation modules can hinder training due to unstable gradient flow. Thus, we unify the sampling and the aggregation within a single SNN, using the potential accumulation within the specified sampling window $(t^{k-1}(x, y, p), t^k(x, y, p)]$ as the aggregation input for the downstream network:

$$\hat{f}_k(x,y,p) = \sum_{(x_i,y_i,p_i,t_i) \in \Omega^k(x,y,p)} u_{t_i}(x_i, y_i, p_i) \tag{9}$$

Notably, we adopt the hard reset to clear the membrane potential from the previous sample. Such a model incorporating the spike-driven sampling mechanism and the synaptic recurrent connection is referred to as the 'ARSNN' for the latter discussion.

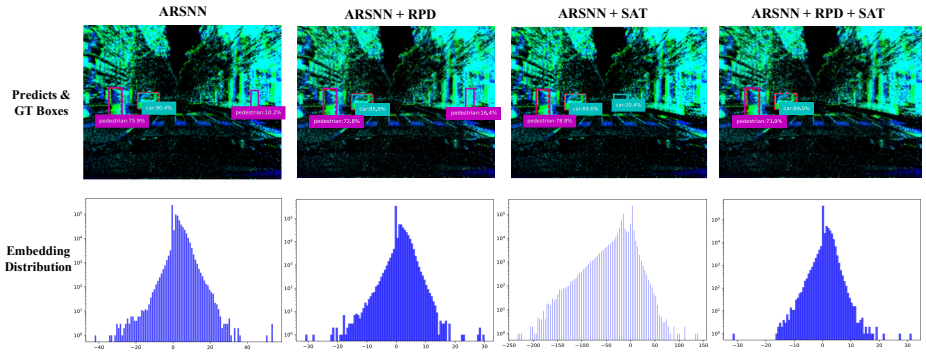


Fig. 3: Visualizing the detection results and corresponding embedding distributions of different models in ablation study. The red dashed line highlights the ground truth.

3.3 Mitigating Degradation of Spike-based Sampling

In light of the adaptive sampling, it can be integrated with the downstream detector for end-to-end training via backpropagation through time (BPTT) [62] with surrogate gradient functions [16, 56, 64]. However, this aggregation method in Eq. (9) implicitly optimizes the sampling module by relying solely on the aggregation of membrane potentials, which lacks information from the firing threshold for optimizing the spike firing timing. Formally, we leverage the gradients from the activation-based technique in SNN training to estimate the update of firing time t^k (detailed derivation provided in the Supplementary Material):

$$\Delta t^k(x,y,p) \propto -\frac{\partial \mathcal{L}((\mathcal{D}\circ\mathcal{A}\circ\mathcal{S})(\mathbf{E},t),\mathbf{B}_t)}{\partial \hat{f}_k(x,y,p)}(u_{t^k}(x,y,p)-u_{t^k-1}(x,y,p)) \quad (10)$$

This reveals that the adjustment in firing time t^k depends solely on the gradient of the membrane potential $u_{t^k}(x,y,p)$ and the instantaneous change in potential at the firing moment. It disregards the relationship between the firing time update Δt^k and the firing threshold θ. To address this issue, we introduce Spike-aware Training (SAT), which incorporates the firing spike $s_{t^k}(x,y,p)$ into the aggregation function as follows:

$$\hat{f}_k(x,y,p) = s_{t^k}(x,y,p) \sum_{t'=t^{k-1}(x,y,p)+1}^{t^k(x,y,p)} u_{t'}(x,y,p) \quad (11)$$

Given $s_{t^k}=1$, this modification does not alter the forward computation of the aggregation function. However, it effectively rewires the gradient flow:

$$\Delta t^k \propto -\frac{\partial \mathcal{L}}{\partial \hat{f}_k}(1+\sum_{t'} u_{t'} h_\alpha(u_{t^k}-\theta))(u_{t^k}-u_{t^k-1}), \quad \forall\ (x,y,p) \quad (12)$$

where $h_\alpha(x)$ is the surrogate gradient function used to circumvent the non-differential problem of the Heaviside function in Eq. (5). Here, $1+u_{t^k}h_\alpha(u_{t^k}-\theta)$

represents the adjustment from threshold-based gradient redirection, enhancing the update of firing time $s_t^k(x, y, p)$ especially when near the threshold θ, thereby achieving more precise sampling away from critical points. Moreover, the implementation of a spike-based sampling module introduces the dilemma of handling data after the final spike firing. An intuitive idea is to incorporate the remaining information into the embedding to reduce information loss. However, this leads to a scenario where neurons, even without firing, accumulate substantial information for representation. This inadvertently causes sampling degradation by diminishing the incentive for the sampling module to learn accurate spike timing. Hence, we propose the strategy of Residual Potential Dropout (RPD) to exclude all events following the last spike, enhancing the learning of precise firing moments. To illustrate the efficacy of RPD, we show the potential aggregation distribution with and without RPD. The results in Fig. 3 reveal a more compact feature distribution with RPD. Moreover, employing SAT for gradient guidance (as in the case of ARSNN+SAT), we observe that sampler training without RPD tends to exhibit mode collapse towards the non-firing spectrum with concentrated negative membrane potentials.

4 Experiments

The effectiveness of the proposed approach is first validated in Sec. 4.1. In Sec. 4.2, we compare our framework with state-of-the-art event-based methods across benchmark datasets. Sec. 4.3 explores adaptability, hyper-parameter sensitivity, and sampling diversity, while Sec. 4.4 examines energy efficiency.

Implementation Details. The ADAM optimizer is applied with learning rates of 1e-3 for Gen1, 1e-4 for N-Caltech 101, and 5e-4 for 1Mpx. Consistent with [21,25], data augmentation techniques, including random Zoom-in and horizontal flipping, are employed for all datasets. For Gen-1, event sampling is capped at 200ms prior to the timestamp of labeling. For 1Mpx, we preprocess the event stream following [21], while using a subsequence of length 4 for BPTT training. Unlike the hard reset used in the sampling module, downstream networks implement a soft reset strategy. Instead of employing a learning rate scheduler, we leverage model weight smoothing during training through the use of an exponential moving average with a momentum of 0.9999. Additionally, we utilize SpikingJelly [14] to enhance the training efficiency of the spike-based backbone.

Benchmark Datasets. The Neuromorphic-Caltech 101 (N-Caltech 101) dataset [47], captured by an ATIS event camera [51], comprises 8,709 event streams, each containing 300 ms of saccades with bounding box annotations and object contours. We partition this dataset into an 8:2 split for training and validation. The Prophesee Gen1 dataset [11] offers over 39 hours of automotive recordings, annotated with more than 228k car and 28k pedestrian labels, facilitating event-based detection in diverse driving scenarios. The 1Mpx dataset [50] features high-resolution driving recordings (1280×720) across various scenarios, including over 24 million labeled bounding boxes of cars, pedestrians, and two-wheelers, annotated at 60Hz. Following the original evaluation protocol, we exclude pre-

Table 1: Performance comparison between Event Count and the proposed spike-based approaches, with and without the adaptive sampling mechanism.

Method	RPD	SAT	N-Caltech 101		Gen1	
			mAP_{50}	$mAP_{50:95}$	mAP_{50}	$mAP_{50:95}$
Event Count	-	-	0.587	0.369	0.611	0.321
SNN	-	-	0.567	0.367	0.670	0.366
RSNN	-	-	0.634	0.412	0.719	0.399
ARSNN	×	×	0.600	0.398	0.687	0.373
ARSNN	✓	×	0.586	0.391	0.714	0.393
ARSNN	×	✓	0.592	0.376	0.714	0.398
ARSNN	✓	✓	0.664	0.437	0.731	0.409

Table 2: Performance comparison with popular event representation methods on both Gen1 and N-Caltech 101 datasets.

Method	N-Caltech 101		Gen1	
	mAP_{50}	$mAP_{50:95}$	mAP_{50}	$mAP_{50:95}$
Event Count	0.587	0.369	0.611	0.321
Voxel Grid	0.598	0.391	0.515	0.247
Time Surface	0.497	0.288	0.654	0.354
Voxel Cube	0.589	0.401	0.651	0.340
SNN	0.567	0.367	0.670	0.366
RSNN	0.634	0.412	0.719	0.399
ARSNN	0.664	0.437	0.731	0.409

diction boxes with a side length below 10 (20) pixels or a diagonal shorter than 30 (60) pixels after non-maximum suppression for the Gen1 (1Mpx) dataset.

4.1 Ablation Study

In this section, we conduct a comparative analysis to ascertain the efficacy of individual components within the proposed sampling module, as detailed in Table 1. In addition to comparing with Event Count used in early aggregation, we introduce two novel embedding strategies leveraging SNNs and RSNNs, predicated on potential aggregation. The results show that the integration of recurrent synaptic connections significantly enhances network performance on both datasets, with observed improvements of 6.7% on N-Caltech 101 and 4.9% on Gen1. Notably, employing the SAT and RPD modules in isolation with adaptive sampling sometimes results in a performance decline compared to RSNN (58.6% vs. 63.4% in mAP_{50} on N-Caltech 101). Conversely, when SAT and RPD are synergistically combined within the SNN framework, there is a significant performance improvement, yielding increases of 3.0% and 1.2% in mAP_{50} over RSNN on the N-Caltech 101 and Gen1 datasets, respectively. This demonstrates the critical importance of integrating these components to achieve optimal network performance. Furthermore, Table 2 compares our approach with established event representations such as Voxel Grid [2], Time Surface [57], and Voxel Cube [9]. It shows that the proposed adaptive method outperforms these alternatives on both datasets. Moreover, the framework can incorporate various event representations at the early aggregation stage for better performance.

4.2 Performance Comparison

We evaluate our proposed method against existing models on the Gen1, N-Caltech 101 and 1Mpx datasets. To improve the overall efficacy, we adopt PLIF [16] neurons with learnable decay and tdBN [71] in the downstream detector. The models, with sizes of 8.92M (SYOLOX-S) and 25.3M (SYOLOX-M) parameters, mirror the network architectures of YOLOX [19]. As shown in Table 3, our technique outperforms the state-of-the-art spike-based method [58] (0.354 vs. 0.310

Table 3: Performance comparison with the state-of-the-art methods on the Gen1 dataset. The letters M and S in parentheses denote medium and small-scale network structures, respectively, achieved by scaling the network width.

Method	Representation	Backbone	Head	Temporal	#Params	Timestep	mAP_{50}	$mAP_{50:95}$
RED [50]	Event Volum	CNN+RNN	SSD	Yes	24.1M	-	-	0.40
RVT [21]	HIST.	Transformer+RNN	YOLOX	Yes	18.5M	21	-	0.472
ERGO-12 [75]	ERGO-12	Transformer	YOLOv6	No	-	-	-	0.504
AEGNN [54]	Graph	GNN	YOLO	No	20.0M	-	-	0.163
NVS-S [37]	Graph	GNN	YOLO	No	0.9M	-	-	0.086
TAF [41]	CNN	CNN	YOLOX	No	14.8M	-	-	0.454
ASTMNet [35]	1D TCNN	CNN+RNN	SSD	Yes	>100M	3	-	0.467
MatrixLSTM [5]	LSTM	CNN	YOLOv3	Yes	61.5M	-	-	0.310
VC-DenseNet [9]	Voxel Cube	SNN	SSD	Yes	8.2M	5	-	0.189
VC-MobileNet [9]	Voxel Cube	SNN	SSD	Yes	24.26M	5	-	0.147
LT-SNN [28]	HIST.	SNN	YOLOv2	Yes	-	-	-	0.298
KD-SNN [3]	HIST.	SNN	CenterNet	Yes	12.97M	5	-	0.229
EMS-ResNet10 [58]	HIST.	SNN	YOLOv3	Yes	6.2M	5	0.547	0.267
EMS-ResNet18 [58]	HIST.	SNN	YOLOv3	Yes	9.34M	5	0.565	0.286
EMS-ResNet34 [58]	HIST.	SNN	YOLOv3	Yes	14.40M	5	0.590	0.310
TR-YOLO [67]	HIST.	SNN	YOLOv3	Yes	8.7M	3	0.451	-
EAS-SNN (M)	**ARSNN**	**SNN**	YOLOX$^{\diamond}$	Yes	25.3M	3	0.731	0.409
			YOLOX*	Yes	25.3M	3	0.718	0.393
			YOLOX†	Yes	25.3M	3	0.699	0.375
EAS-SNN (S)	**ARSNN**	**SNN**	YOLOX$^{\diamond}$	Yes	8.92M	3	0.687	0.372
			YOLOX*	Yes	8.92M	3	0.692	0.372
			YOLOX†	Yes	8.92M	3	0.675	0.354

$\dagger$ *denotes the model uses a spiking backbone, a spiking feature pyramid network, and a spiking detection head.*
$*$ *denotes the model uses a spiking backbone, a spiking feature pyramid network, and a non-spiking detection head.*
$\diamond$ *denotes the model uses a spiking backbone, a non-spiking feature pyramid network, and a non-spiking detection head.*

Table 4: Performance comparison with the existing methods on N-Caltech 101.

Method	Representation	Backbone	Head	Temporal	#Params	Timestep	mAP_{50}	$mAP_{50:95}$
YOLE [4]	Leaky Surface	CNN	YOLO	No	-	-	0.398	-
AEGNN [54]	Graph	GNN	YOLO	No	20.0M	-	0.595	-
NVS-S [37]	Graph	GNN	YOLO	No	0.9M	-	0.346	-
ASYNet [44]	HIST.	CNN	YOLO	No	-	-	0.643	-
ASYNet [44]	QUE.	CNN	YOLO	No	-	-	0.615	-
EAS-SNN(M)	**ARSNN**	**SNN**	YOLO-X	Yes	25.3M	3	0.664	0.437
EAS-SNN(S)	**ARSNN**	**SNN**	YOLO-X	Yes	8.92M	3	0.538	0.338

$mAP_{50:95}$) while using fewer parameters (8.92 M vs. 14.40 M) under three time steps. Even compared with ANNs that feature well-designed attention mechanisms [21] and recurrent architectures [35, 50], our method remains competitive and notably achieves over 40% $mAP_{50:95}$ on Gen1 through sparse spike-driven operations, thereby enhancing energy efficiency as detailed in Sec. 4.4. Regarding N-Caltech 101 (Table 4), our approach improves the detection performance with spike-based representation. On the 1Mpx dataset, the largest event-based detection dataset, the proposed method achieves the first spike-based results, demonstrating remarkable performance using SNNs, as depicted in Table 5.

In object detection, each module has a specific role: backbones extract features, Feature Pyramid Networks (FPN) fuse multi-scale features, and detection heads handle coordinate regression and object recognition. We explored the impact of sparse spike representation by integrating a non-spiking detection head and FPN into a hybrid architecture. As shown in Table 3, a spiking FPN within

Table 5: Performance comparison with the state-of-the-art methods on 1Mpx.

Method	Representation	Backbone	Head	Temporal	#Params	Timestep	mAP_{50}	$mAP_{50:95}$
RED [50]	Event Volum	CNN+RNN	SSD	Yes	24.1M	-	-	0.430
RVT [21]	HIST.	Transformer+RNN	YOLOX	Yes	18.5M	21	-	0.474
ERGO-12 [75]	ERGO-12	Transformer	YOLOv6	No	-	-	-	0.406
ERGO-12 [75]	HIST.	Transformer	YOLOv6	No	-	-	-	0.327
E2VID [52]	Reconstruction	CNN	RetinaNet	No	-	-	-	0.250
ASTMNet [35]	1D TCNN	CNN+RNN	SSD	Yes	>100M	3	-	0.483
Events+YOLOv3 [29]	HIST.	CNN	YOLOv3	No	-	-	-	0.346
EAS-SNN (M)	**ARSNN**	**SNN**	YOLOX	Yes	25.3M	3	0.651	0.362

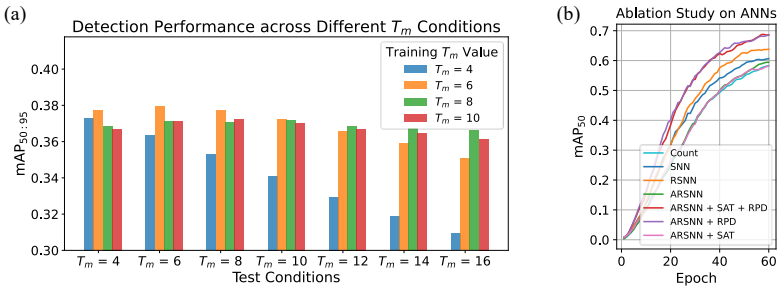


Fig. 4: (a) Assessing the scalability of the early aggregation step T_m in mismatched testing scenarios on Gen1. (b) Evaluating the effectiveness of the proposed adaptive sampling mechanism within conventional dense neural networks on N-Caltech 101.

a compact 8.92M network outperforms its non-spiking counterpart, indicating that spike-based architectures might offer regularization benefits for event-based detection. However, spiking detection heads within constrained time steps appear to reduce network performance.

4.3 Experimental Analysis

Scalability to Dense Neural Network: In this section, our objective is to validate the efficacy of the proposed adaptive sampling technique within conventional dense neural networks. Here, we only transmit the initial aggregated potential to subsequent networks. Fig. 4b displays the mAP_{50} training curves on the N-Caltech 101 dataset, comparing different methods. The combination of ARSNN with SAT and RPD demonstrates promising performance improvements, while ARSNN with SAT or RPD alone leads to declined performance, highlighting the synergistic benefits of our proposed modules. Additionally, recurrent synaptic connections further improve network performance.

Impact of Early Aggregation: We investigate the influence of early aggregation, as defined in Eq. (4), by modulating the aggregation step T_m on Gen1 using spiking YOLOX-S. The first three slices sampled from T_m-step inputs are aggregated and fed sequentially into a spike-based backbone. The parameter T_m indicates the temporal resolution of inputs, where higher T_m corresponds to finer granularity. As depicted in Fig. 4a, optimal performance is generally observed when T_m remains consistent between the training and testing phases. Notably,

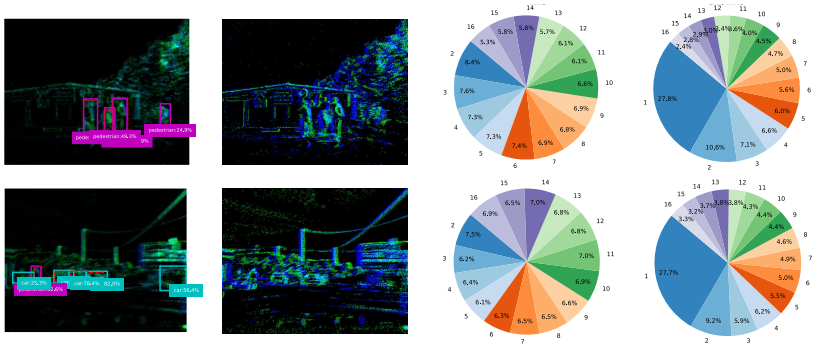


Fig. 5: Exploring the behavior of sampling module. The first column illustrates bounding boxes of prediction, whereas the second column provides pixel-wise firing count of spike neurons. The third and fourth columns, respectively, depict the distribution of neuron firing time and the sampling duration, quantified as the inter-spike interval.

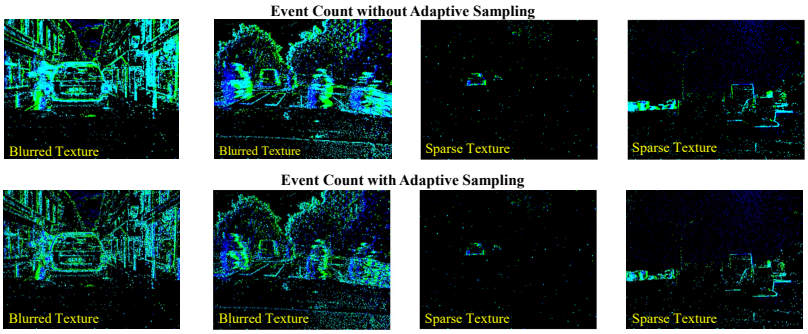


Fig. 6: Comparison of event count images with and without the adaptive sampling module under blurred and sparse contexts.

the model trained with $T_m = 4$, sustains over 30% $\text{mAP}_{50:95}$, despite a fourfold increase in granularity. Overall, models trained with $T_m = 8$ demonstrate considerable robustness, delivering promising performance under all tested conditions.

Distribution of Event Sampling: In this part, we employ the model trained under $T_m = 8$ and examine the behavior of each neuron while using $T_m = 16$ for inference. The sampling intervals of various spiking neurons are shown in the last column of Fig. 5, which demonstrates significant variability. The longest sample covers the full input, while the shortest sample only includes the input of one time step. Here, we disregard the neurons without spike firing, which typically make up 80%+ due to the sparse event input. Additionally, compared to the event count representation in the first column, the image of the firing count in the second column offers a clearer estimation of the scene. The qualitative results of adaptive sampling are further presented in Fig. 6. The top row shows event count image without adaptive sampling, under blurred and sparse textures. The bottom row demonstrates the improvement in information retention and noise reduction with adaptive sampling in the same contexts.

Table 6: Comparison of energy consumption based on number of synaptic operations.

Module	YOLOX-M		Spiking YOLOX-M			YOLOX-S		Spiking YOLOX-S		
	MAC(G)	Energy(mJ)	AC(G)	MAC(G)	Energy(mJ)	MAC(G)	Energy(mJ)	AC(G)	MAC(G)	Energy(mJ)
Embedding	0	0	0.02	1.63	7.52	0	0	0.02	1.63	7.52
Backbone	16.21	74.57	8.49	1.41	14.12	5.24	24.10	2.66	0.71	5.64
FPN	8.19	37.68	3.40	0	3.06	2.63	12.09	1.26	0	1.14
Head	11.33	52.10	3.78	0	3.40	5.04	23.17	1.79	0	1.61
Total	35.73	164.35(1×)	15.68	3.04	28.10(5.85×)	12.90	59.35(1×)	5.74	2.34	15.91(3.73×)

4.4 Energy Efficiency and Inference Speed

While ANNs execute dense multiplication-and-accumulation (MAC) operations determined by their architecture, SNNs on neuromorphic hardware utilize input-conditioned spikes for sparse accumulation (AC) operations, thereby enhancing energy efficiency. To evaluate the energy efficiency of SNNs, we compute the average AC operations across Gen1, as detailed in Table 6. According to [26], we estimate the energy cost for 32-bit floating-point AC and MAC as 0.9 pJ and 4.6 pJ per operation, respectively. The results show a significant reduction in energy consumption with spike-driven computation, achieving a 5.85× reduction for YOLOX-M and 3.73× for YOLOX-S. It is worth noting that, for ANNs, we omit the energy consumption of the embedding phase, offering a comparatively conservative estimate of power savings. Detailed firing rates and additional results for different models are provided in Supplementary Materials.

Furthermore, we compared the frames per second (FPS) performance of our model with other state-of-the-art methods on the Gen1 dataset. Our model achieves competitive detection speeds of 54.35 FPS, compared to 59.88 FPS for RED [50], 28.09 FPS for ASTMNet [35], 106.38 FPS for RVT-T [21], and 59.52 FPS for GET [49]. This demonstrates the significant potential for higher efficiency on neuromorphic chips, which support sparse spike-driven operations.

5 Conclusion

We introduce an adaptive sampling method utilizing recurrent spiking neural networks (ARSNN), enhanced by Residual Potential Dropout (RPD) and Spike-Aware Training (SAT) specifically to mitigate performance degradation in event-based sampling. Integrated with a spike-based architecture in the downstream neural network, our comprehensive framework, EAS-SNN, outperforms existing spike-based detection methods and offers superior energy efficiency compared to ANNs. To the best of our knowledge, it is the first effort toward end-to-end optimization for event-based sampling and representation. Additionally, we demonstrate the versatility of our proposed sampling technique through its application to dense neural networks. Despite exploring robustness across varying granularities of early aggregation, the ultimate goal is to develop a spike-based sampler that can precisely synchronize with the temporal granularity of raw event streams, measured in microseconds, without necessitating early aggregation. Our future work will focus on achieving this granularity, pending advancements in neuromorphic computing infrastructure.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant 62236007, and the Key R&D Program of Zhejiang under Grant 2023C03001 and 2022C01048.

References

1. Akopyan, F., Sawada, J., Cassidy, A., Alvarez-Icaza, R., Arthur, J., Merolla, P., Imam, N., Nakamura, Y., Datta, P., Nam, G.J., et al.: Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. IEEE transactions on computer-aided design of integrated circuits and systems **34**(10), 1537–1557 (2015)
2. et al., Z.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: CVPR (2019)
3. Bodden, L., Schwaiger, F., Ha, D.B., Kreuzberg, L., Behnke, S.: Spiking centernet: A distillation-boosted spiking neural network for object detection. arXiv preprint arXiv:2402.01287 (2024)
4. Cannici, M., Ciccone, M., Romanoni, A., Matteucci, M.: Asynchronous convolutional networks for object detection in neuromorphic cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
5. Cannici, M., Ciccone, M., Romanoni, A., Matteucci, M.: A differentiable recurrent surface for asynchronous event-based data. In: The European Conference on Computer Vision (ECCV) (August 2020)
6. Cao, J., Sun, M., Wang, Z., Cheng, H., Zhang, Q., Xu, R.: BEEF: Building a bridge from event to frame (2024), `https://openreview.net/forum?id=IHedM0Zem9`
7. Chen, X., Wu, J., Tang, H., Ren, Q., Tan, K.C.: Unleashing the potential of spiking neural networks for sequential modeling with contextual embedding. arXiv preprint arXiv:2308.15150 (2023)
8. Chowdhury, S.S., Rathi, N., Roy, K.: Towards ultra low latency spiking neural networks for vision and sequential tasks using temporal pruning. In: European Conference on Computer Vision. pp. 709–726. Springer (2022)
9. Cordone, L., Miramond, B., Thierion, P.: Object detection with spiking neural networks on automotive event data. In: 2022 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2022)
10. Davies, M., Srinivasa, N., Lin, T.H., Chinya, G., Cao, Y., Choday, S.H., Dimou, G., Joshi, P., Imam, N., Jain, S., et al.: Loihi: A neuromorphic manycore processor with on-chip learning. Ieee Micro **38**(1), 82–99 (2018)
11. De Tournemire, P., Nitti, D., Perot, E., Migliore, D., Sironi, A.: A large scale event-based detection dataset for automotive. arXiv preprint arXiv:2001.08499 (2020)
12. Dong, Y., Li, Y., Zhao, D., Shen, G., Zeng, Y.: Bullying10k: a large-scale neuromorphic dataset towards privacy-preserving bullying recognition. Advances in Neural Information Processing Systems **36** (2024)
13. Duan, C., Ding, J., Chen, S., Yu, Z., Huang, T.: Temporal effective batch normalization in spiking neural networks. Advances in Neural Information Processing Systems **35**, 34377–34390 (2022)
14. Fang, W., Chen, Y., Ding, J., Yu, Z., Masquelier, T., Chen, D., Huang, L., Zhou, H., Li, G., Tian, Y.: Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. Science Advances **9**(40), eadi1480 (2023)

15. Fang, W., Yu, Z., Chen, Y., Huang, T., Masquelier, T., Tian, Y.: Deep residual learning in spiking neural networks. Advances in Neural Information Processing Systems **34**, 21056–21069 (2021)
16. Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., Tian, Y.: Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In: ICCV. pp. 2661–2671 (2021)
17. Finateu, T., Niwa, A., Matolin, D., Tsuchimoto, K., Mascheroni, A., Reynaud, E., Mostafalu, P., Brady, F., Chotard, L., LeGoff, F., et al.: 5.10 a 1280× 720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μm pixels, 1.066 geps readout, programmable event-rate controller and compressive data-formatting pipeline. In: 2020 IEEE International Solid-State Circuits Conference-(ISSCC). pp. 112–114. IEEE (2020)
18. Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., et al.: Event-based vision: A survey. IEEE transactions on pattern analysis and machine intelligence **44**(1), 154–180 (2020)
19. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
20. Gehrig, D., Loquercio, A., Derpanis, K.G., Scaramuzza, D.: End-to-end learning of representations for asynchronous event-based data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5633–5643 (2019)
21. Gehrig, M., Scaramuzza, D.: 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)
22. George, A.M., Banerjee, D., Dey, S., Mukherjee, A., Balamurali, P.: A reservoir-based convolutional spiking neural network for gesture recognition from dvs input. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–9. IEEE (2020)
23. Guo, Y., Chen, Y., Zhang, L., Liu, X., Wang, Y., Huang, X., Ma, Z.: Im-loss: information maximization loss for spiking neural networks. Advances in Neural Information Processing Systems **35**, 156–166 (2022)
24. Guo, Y., Chen, Y., Zhang, L., Wang, Y., Liu, X., Tong, X., Ou, Y., Huang, X., Ma, Z.: Reducing information loss for spiking neural networks. In: European Conference on Computer Vision (Cham: Springer). p. 36–52 (2022)
25. Hamaguchi, R., Furukawa, Y., Onishi, M., Sakurada, K.: 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)
26. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. Advances in Neural Information Processing Systems **28** (2015)
27. Hao, Z., Shi, X., Huang, Z., Bu, T., Yu, Z., Huang, T.: A progressive training framework for spiking neural networks with learnable multi-hierarchical model. In: The Twelfth International Conference on Learning Representations (2024), `https://openreview.net/forum?id=g52tgL8jy6`
28. Hasssan, A., Meng, J., sun Seo, J.: LT-SNN: Self-adaptive spiking neural network for event-based classification and object detection (2023), `https://openreview.net/forum?id=oyzMyylgINj`
29. Jiang, Z., Xia, P., Huang, K., Stechele, W., Chen, G., Bing, Z., Knoll, A.: Mixed frame-/event-driven fast pedestrian detection. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 8332–8338. IEEE (2019)

30. Kim, Y., Li, Y., Park, H., Venkatesha, Y., Panda, P.: Neural architecture search for spiking neural networks. In: European Conference on Computer Vision. pp. 36–56. Springer (2022)
31. Kugele, A., Pfeil, T., Pfeiffer, M., Chicca, E.: Hybrid snn-ann: Energy-efficient classification and object detection for event-based vision. In: DAGM German Conference on Pattern Recognition. pp. 297–312. Springer (2021)
32. Kugele, A., Pfeil, T., Pfeiffer, M., Chicca, E.: How many events make an object? improving single-frame object detection on the 1 mpx dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3912–3921 (2023)
33. Lagorce, X., Meyer, C., Ieng, S.H., Filliat, D., Benosman, R.: Asynchronous event-based multikernel algorithm for high-speed visual features tracking. IEEE transactions on neural networks and learning systems **26**(8), 1710–1720 (2014)
34. Lagorce, X., Orchard, G., Galluppi, F., Shi, B.E., Benosman, R.B.: Hots: a hierarchy of event-based time-surfaces for pattern recognition. IEEE transactions on pattern analysis and machine intelligence **39**(7), 1346–1359 (2016)
35. Li, J., Li, J., Zhu, L., Xiang, X., Huang, T., Tian, Y.: Asynchronous spatio-temporal memory network for continuous event-based object detection. IEEE Transactions on Image Processing **31**, 2975–2987 (2022)
36. Li, J., Wang, X., Zhu, L., Li, J., Huang, T., Tian, Y.: Retinomorphic object detection in asynchronous visual streams. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 1332–1340 (2022)
37. Li, Y., Zhou, H., Yang, B., Zhang, Y., Cui, Z., Bao, H., Zhang, G.: Graph-based asynchronous event processing for rapid object recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 934–943 (2021)
38. Li, Y., Guo, Y., Zhang, S., Deng, S., Hai, Y., Gu, S.: Differentiable spike: Rethinking gradient-descent for training spiking neural networks. NeurIPS **34** (2021)
39. Lian, S., Shen, J., Liu, Q., Wang, Z., Yan, R., Tang, H.: Learnable surrogate gradient for direct training spiking neural networks. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23. pp. 3002–3010 (2023)
40. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. IEEE Journal of Solid-State Circuits **43**(2), 566–576 (2008). https://doi.org/10.1109/JSSC.2007.914337
41. Liu, B., Xu, C., Yang, W., Yu, H., Yu, L.: Motion robust high-speed light-weighted object detection with event camera. IEEE Transactions on Instrumentation and Measurement (2023)
42. Liu, Q., Xing, D., Tang, H., Ma, D., Pan, G.: Event-based action recognition using motion information and spiking neural networks. In: IJCAI. pp. 1743–1749 (2021)
43. Messikommer, N., Gehrig, D., Gehrig, M., Scaramuzza, D.: Bridging the gap between events and frames through unsupervised domain adaptation. IEEE Robotics and Automation Letters **7**(2), 3515–3522 (2022)
44. Messikommer, N., Gehrig, D., Loquercio, A., Scaramuzza, D.: Event-based asynchronous sparse convolutional networks. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16. pp. 415–431. Springer (2020)
45. Mirsadeghi, M., Shalchian, M., Kheradpisheh, S.R., Masquelier, T.: Stidi-bp: Spike time displacement based error backpropagation in multilayer spiking neural networks. Neurocomputing **427**, 131–140 (2021)

46. Mostafa, H.: Supervised learning based on temporal coding in spiking neural networks. IEEE transactions on neural networks and learning systems **29**(7), 3227–3235 (2017)
47. Orchard, G., Jayawant, A., Cohen, G.K., Thakor, N.: Converting static image datasets to spiking neuromorphic datasets using saccades. Frontiers in neuroscience **9**, 437 (2015)
48. Pei, J., Deng, L., Song, S., Zhao, M., Zhang, Y., Wu, S., Wang, G., Zou, Z., Wu, Z., He, W., et al.: Towards artificial general intelligence with hybrid tianjic chip architecture. Nature **572**(7767), 106–111 (2019)
49. Peng, Y., Zhang, Y., Xiong, Z., Sun, X., Wu, F.: Get: Group event transformer for event-based vision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6038–6048 (2023)
50. Perot, E., De Tournemire, P., Nitti, D., Masci, J., Sironi, A.: Learning to detect objects with a 1 megapixel event camera. Advances in Neural Information Processing Systems **33**, 16639–16652 (2020)
51. Posch, C., Matolin, D., Wohlgenannt, R.: A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. IEEE Journal of Solid-State Circuits **46**(1), 259–275 (2010)
52. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: High speed and high dynamic range video with an event camera. IEEE transactions on pattern analysis and machine intelligence **43**(6), 1964–1980 (2019)
53. Ren, D., Ma, Z., Chen, Y., Peng, W., Liu, X., Zhang, Y., Guo, Y.: Spiking pointnet: Spiking neural networks for point clouds. Advances in Neural Information Processing Systems **36** (2024)
54. Schaefer, S., Gehrig, D., Scaramuzza, D.: Aegnn: Asynchronous event-based graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12371–12381 (2022)
55. Serrano-Gotarredona, T., Linares-Barranco, B.: A 128 $\times$128 1.5% contrast sensitivity 0.9% fpn 3 μs latency 4 mw asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers. IEEE Journal of Solid-State Circuits **48**(3), 827–838 (2013)
56. Shrestha, S.B., Orchard, G.: Slayer: Spike layer error reassignment in time. arXiv preprint arXiv:1810.08646 (2018)
57. Sironi, A., Brambilla, M., Bourdis, N., Lagorce, X., Benosman, R.: Hats: Histograms of averaged time surfaces for robust event-based object classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1731–1740 (2018)
58. Su, Q., Chou, Y., Hu, Y., Li, J., Mei, S., Zhang, Z., Li, G.: Deep directly-trained spiking neural networks for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6555–6565 (2023)
59. Wang, D., Jia, X., Zhang, Y., Zhang, X., Wang, Y., Zhang, Z., Wang, D., Lu, H.: Dual memory aggregation network for event-based object detection with learnable representation. arXiv preprint arXiv:2303.09919 (2023)
60. Wang, Z., Jiang, R., Lian, S.J., Yan, R., Tang, H.: Adaptive smoothing gradient learning for spiking neural networks. In: International Conference on Machine Learning (2023)
61. Wang, Z., Zhang, Y., Lian, S., Cui, X., Yan, R., Tang, H.: Toward high-accuracy and low-latency spiking neural networks with two-stage optimization. IEEE Transactions on Neural Networks and Learning Systems (2023)
62. Werbos, P.J.: Generalization of backpropagation with application to a recurrent gas market model. Neural networks **1**(4), 339–356 (1988)

63. Wu, J., Yılmaz, E., Zhang, M., Li, H., Tan, K.C.: Deep spiking neural networks for large vocabulary automatic speech recognition. Frontiers in neuroscience **14**, 199 (2020)
64. Wu, Y., Deng, L., Li, G., Zhu, J., Shi, L.: Spatio-temporal backpropagation for training high-performance spiking neural networks. Frontiers in neuroscience **12**, 331 (2018)
65. Yao, M., Gao, H., Zhao, G., Wang, D., Lin, Y., Yang, Z., Li, G.: Temporal-wise attention spiking neural networks for event streams classification. In: ICCV. pp. 10221–10230 (2021)
66. Yao, X., Li, F., Mo, Z., Cheng, J.: Glif: A unified gated leaky integrate-and-fire neuron for spiking neural networks. arXiv preprint arXiv:2210.13768 (2022)
67. Yuan, M., Zhang, C., Wang, Z., Liu, H., Pan, G., Tang, H.: Trainable spiking-yolo for low-latency and high-performance object detection. Neural Networks **172**, 106092 (2024)
68. Zenke, F., Vogels, T.P.: The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. Neural Computation **33**(4), 899–925 (2021)
69. Zhang, M., Wang, J., Wu, J., Belatreche, A., Amornpaisannon, B., Zhang, Z., Miriyala, V.P.K., Qu, H., Chua, Y., Carlson, T.E., et al.: Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks. IEEE transactions on neural networks and learning systems **33**(5), 1947–1958 (2021)
70. Zhang, W., Li, P.: Temporal spike sequence learning via backpropagation for deep spiking neural networks. Advances in Neural Information Processing Systems **33**, 12022–12033 (2020)
71. Zheng, H., Wu, Y., Deng, L., Hu, Y., Li, G.: Going deeper with directly-trained larger spiking neural networks. In: AAAI 2021. pp. 11062–11070. AAAI Press (2021)
72. Zhou, Z., Zhu, Y., He, C., Wang, Y., Yan, S., Tian, Y., Yuan, L.: Spikformer: When spiking neural network meets transformer. arXiv preprint arXiv:2209.15425 (2022)
73. Zhu, Y., Fang, W., Xie, X., Huang, T., Yu, Z.: Exploring loss functions for time-based training strategy in spiking neural networks. Advances in Neural Information Processing Systems **36** (2024)
74. Zhu, Y., Yu, Z., Fang, W., Xie, X., Huang, T., Masquelier, T.: Training spiking neural networks with event-driven backpropagation. Advances in Neural Information Processing Systems **35**, 30528–30541 (2022)
75. Zubić, N., Gehrig, D., Gehrig, M., Scaramuzza, D.: From chaos comes order: Ordering event representations for object recognition and detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12846–12856 (2023)