Spike-driven Discrete Aggregation for Event-based Object Detection

Supplementary Material

1. Pseudocode

The pseudocode of SDA-MTF is shown in Algorithm 1.

2. More Implementation Details

We train DASNN(M) with 24 batchsize and DASNN-MTF(M) with 20 batchsize for 40 epochs on 4 NVIDIA GeForce RTX 3090. The training of DASNN(M) takes approximately 39 hours, while DASNN-MTF(M) requires about 52 hours under the same experimental settings.

3. Inference Time

We evaluate the inference time of DASNN and DASNN-MTF on the Gen1 dataset, which are 25.3 ms and 37.7 ms per sample, respectively. Due to the finer-grained operations along the temporal dimension, SDA and SDA-MTF trade part of the inference time for substantial performance gains (DASNN is 6.9 ms slower than EAS-SNN yet improves $\text{mAP}_{50:95}$ by 5.3%).

4. Additional Ablations

4.1. Compatibility with Lightweight Architectures

We also evaluate the compatibility of SDA and SDA-MTF with a smaller model, RVT-T (only 4.4M parameters). As shown in Tab. 1, SDA and SDA-MTF consistently improve RVT-T's performance on Gen1 by +1.3 and +1.8 $\text{mAP}_{50:95}$, respectively, confirming their strong compatibility with lightweight architectures.

4.2. Other Attempts

Since a spiking neuron is assigned to each pixel-polarity location (x, y, p), it is natural to consider using individual membrane potential thresholds for different neurons. To explore this idea, we made each neuron's threshold a learnable parameter and trained the model using DASNN-MTF (M). However, we observed a 2.0% performance drop and unstable training dynamics caused by the adaptive thresholds. Therefore, we discarded this design in our final implementation.

5. Additional Experiments

5.1. Heatmap visualization of gates R and F

The visualization from the SDA branch is in Fig. 1. Due to reverse-time processing, the person is actually moving rightward. F maintains strong responses (yellow) inside the target box while suppressing displaced historical motion (black). R progressively filters noise event accumulated in the target region (darker black).

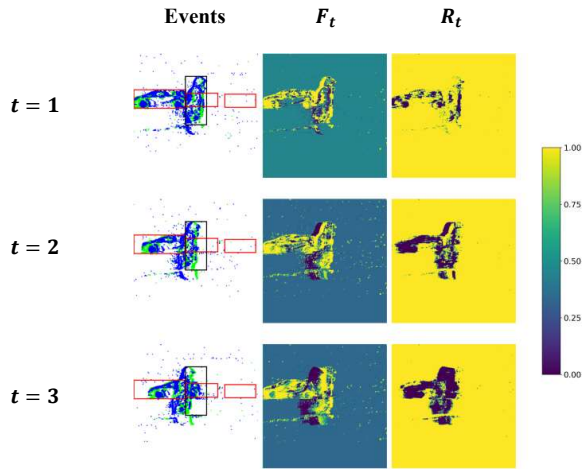

Figure 1. Visualization of gates R and F in a representative event stream.

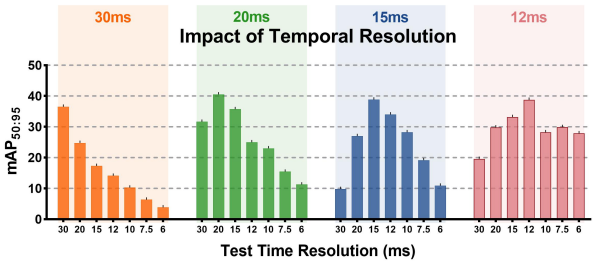

Figure 2. Impact of train temporal resolution on model performance and results across different test temporal resolutions.

5.2. Impact of Temporal Resolution

To investigate the impact of temporal resolution on model performance, we evaluate DASNN-MTF(M) on the Gen1 dataset under different slice intervals Δt_a. We train separate models with four temporal resolutions: 30 ms, 20 ms, 15 ms, and 12 ms, and test them across a wider range, including finer-grained settings of 10 ms, 7.5 ms, and 6 ms.

As shown in Fig. 2, all models achieve the best performance when the test Δt_a matches the train Δt_a. Among them, the model trained at 20 ms yields the highest $\text{mAP}_{50:95}$. Notably, the model trained at 12 ms exhibits strong robustness under finer test temporal resolutions. Unlike other models whose performance degrades with increasing temporal resolution, the 12 ms-trained model

Algorithm 1 Spiking Discrete Aggregation with Multi-Timescale Fusion

Input: Event segments for SAS $< E_1^C, E_2^C, ..., E_{T_m}^C >$, event slices for SDA $< E_1^D, E_2^D, ..., E_{T_a}^D >$, time step T_s for spiking detector

Output: Representation R

Initialize the spike firing state of SAS $s_0^C(x, y, p) = 0$
Initialize the SAS aggregation $f(x, y, p) = 0$
Initialize the spike firing state of SDA $s_0^D(x, y, p) = 0$
Initialize the SDA aggregation $g(x, y, p) = 0$
Initialize the aggregation index $k(x, y, p) = 1$.
$Y = T_a//T_m$
for $i = 1$ to T_m **do**
  if $min(k) > T_s$ **then**
    break
  end if
  $u_i^C(x, y, p), s_i^C(x, y, p) = GRSNN_{SAS}(E_i^C, s_{i-1}^C(x, y, p))$ // Eq.9, 10, 11
  $f_k(x, y, p) = f_k(x, y, p) + u_i^C(x, y, p)$ // Eq.13
  Fetch the aggregation index with spike firing $k'(x', y', p')$
  $M = Sigmoid(f_k(x, y, p))$ // Eq.14
  for $j = 1$ to Y **do**
    $u_j^D(x, y, p), s_j^D(x, y, p) = GRSNN_{SDA}(E_{(i-1)Y+j}^D, s_{j-1}^D(x, y, p), M)$ // Eq.9, 11, 15
    Fetch the potential index with spike firing (x'', y'', p'')
    $g_k(x'', y'', p'') = g_k(x'', y'', p'') + u_j^D(x'', y'', p'')$ // Eq.8
  end for
  $k'(x', y', p') = k'(x', y', p') + 1$
end for
$R = f(x, y, p) + g(x, y, p)$
return R

Table 1. Compatibility of SDA and SDA-MTF when integrated into RVT-T on the Gen1 dataset

Method	Representation	$mAP_{50:95}$
RVT-T	Event Count	44.1
	+ SDA	45.4(**+ 1.3**)
	+ SDA-MTF	45.9(**+ 1.8**)

maintains stable and relatively strong performance even at 6 ms, demonstrating its adaptability to higher temporal resolutions and its generalization ability under challenging, sparse input conditions.

6. Detailed Explanation

6.1. Noise Test Dataset

We construct a noise test dataset by uniformly sampling 10% of the Gen1 test set. 60% of the sampled data are corrupted with random noise following [1], while 40% are injected with natural background noise.

In terms of natural background noise, we consider two scenarios. The first corresponds to events from the Gen1 dataset captured under conditions where there is minimal relative motion between the camera and the background, which we use to simulate sensor-induced artifacts. The second scenario involves motion-related blur, where the background exhibits relative motion with respect to the camera. To simulate this, we retain only the background events by removing events within the ground-truth bounding boxes of target objects.

We integrate SCA and SDA into the YOLOX-M backbone and train both models from scratch on the Gen1 dataset. Without any fine-tuning, the trained models are directly evaluated on the noise test dataset to compare their robustness under different levels of noise corruption.

6.2. Energy Efficiency Analysis

We adopt the energy consumption calculation method from previous SNN research [3, 4]:

$$\begin{aligned} EC_{ANN} &= F \times C_{in} \times C_{out} \times k^2 \times EC_{MAC} \\ EC_{SNN} &= T \times fr \times F \times C_{in} \times C_{out} \times k^2 \times EC_{AC} \end{aligned} \tag{1}$$

where F is the feature output size, C_{in}, C_{out} and k denote the input channel, output channel and kernel size of the convolutional layer. T is the timestep of SNNs and fr

represents the average spike firing rate along T timesteps.

During inference, Sigmoid in SDA-MTF can be replaced by the coarse-branch spike output, with only 0.1 $\text{mAP}_{50:95}$ drop on Gen1. Therefore, we exclude the energy consumption of the Sigmoid operation from the final energy calculation.

The recurrent connections inside spiking neurons and the soft mask in MTF introduce extra computation and we estimate their energy using the same cost as a MAC operation:

$$\begin{aligned} EC_{recurrent} &= T \times 2 \times 2 \times F \times EC_{MAC} \\ EC_{mask} &= T \times 2 \times F \times EC_{MAC} \end{aligned} \tag{2}$$

According to [2], we measure 32-bit floating-point AC and MAC operations, where $EC_{AC} = 0.9pJ$ and $EC_{MAC} = 4.6pJ$.

References

[1] Huachen Fang, Jinjian Wu, Leida Li, Junhui Hou, Weisheng Dong, and Guangming Shi. Aednet: Asynchronous event denoising with spatial-temporal correlation among irregular data. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1427–1435, 2022. 2

[2] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015. 3

[3] 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*, pages 253–272. Springer, 2024. 2

[4] 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*, pages 310–328. Springer, 2024. 2