2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)
October 19-25, 2025. Hangzhou, China

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3315-4393-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IROS60139.2025.11246424

Context-aware Sparse Spatiotemporal Learning for Event-based Vision

Shenqi Wang and Guangzhi Tang

***Abstract*— Event-based camera has emerged as a promising paradigm for robot perception, offering advantages with high temporal resolution, high dynamic range, and robustness to motion blur. However, existing deep learning-based event processing methods often fail to fully leverage the sparse nature of event data, complicating their integration into resource-constrained edge applications. While neuromorphic computing provides an energy-efficient alternative, spiking neural networks struggle to match of performance of state-of-the-art models in complex event-based vision tasks, like object detection and optical flow. Moreover, achieving high activation sparsity in neural networks is still difficult and often demands careful manual tuning of sparsity-inducing loss terms. Here, we propose Context-aware Sparse Spatiotemporal Learning (CSSL), a novel framework that introduces context-aware thresholding to dynamically regulate neuron activations based on the input distribution, naturally reducing activation density without explicit sparsity constraints. Applied to event-based object detection and optical flow estimation, CSSL achieves comparable or superior performance to state-of-the-art methods while maintaining extremely high neuronal sparsity. Our experimental results highlight CSSL's crucial role in enabling efficient event-based vision for neuromorphic processing.**

I. Introduction

Event camera has emerged as a promising paradigm for visual perception in robotics, offering advantages such as high temporal resolution, high dynamic range, and robustness to motion blur compared to conventional frame-based cameras [1]. These attributes make event cameras well-suited for real-time applications in dynamic environments, such as autonomous driving [2] and drone-based agile flight [3]. However, despite these advantages, many existing methods still treat event-based vision as a conventional visual processing task, failing to fully exploit the inherent sparsity of event data [4], which can be a limiting factor for real-time robotics applications, especially on resource-constrained platforms.

Neuromorphic computing offers an alternative paradigm for processing event data, leveraging the inherent sparsity of event streams to achieve high energy efficiency [5]. Therefore, to fully take advantage of neuromorphic computing, it's necessary for the neural network model to maintain neuron activation sparsity. While sparse computing on neuromorphic platforms is promising, training sparse networks can be challenging and often requires careful tuning of loss functions to achieve satisfactory performance [6], [7]. To bridge this gap, recent approaches have introduced sparse convolutional recurrent learning, which maintains activation sparsity in recurrent neural networks while preserving the representational capacity [8].

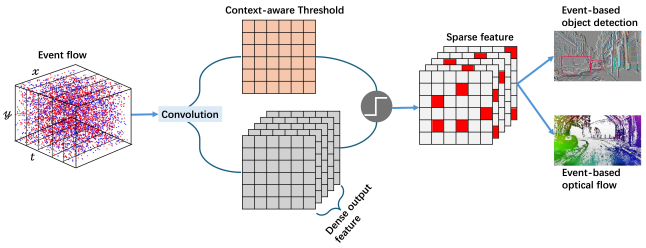


Fig. 1. An overview of the proposed Context-aware Sparse Spatiotemporal Learning (CSSL) framework. CSSL introduces context-aware thresholding to dynamically regulate activations in convolutional modules, selectively filtering out redundant activations while preserving essential information. The framework is applied to event-based object detection and optical flow estimation.

Another promising method to maintain activation sparsity while keeping robust feature learning is introducing a self-adaptive mechanism, that regulates neuron action dynamically based on input characteristics. The context-awareness mechanism in neural networks can dynamically modulate network parameters, activations, or connections based on contextual features extracted from the input [9], which contrasts with traditional networks where the processing remains fixed regardless of the input. These mechanisms allow the network to adapt its processing strategy to focus on the most relevant information in the input, improving performance, efficiency, or robustness. Since the relevant information within an event stream can vary significantly depending on the motion patterns, lighting changes, and object interactions, context awareness mechanism aligns well with event-based vision.

Inspired by recent advances in context-aware neural networks [10], we proposed Context-aware Sparse Spatiotemporal Learning (CSSL), a novel framework that dynamically modulates neuron activations based on contextual information to address these challenges. Unlike standard activation functions such as ReLU, which apply fixed thresholds, CSSL learns adaptive thresholds based on the input distribution, ensuring that only the most relevant neurons are activated. By incorporating context-aware thresholding into both convolutional and recurrent architectures, CSSL maximizes computa-

Shenqi Wang is with the Faculty of Aerospace Engineering, Delft University of Technology, Delft 2628 CD, The Netherlands. s.wang-18@tudelft.nl †Corresponding author

Guangzhi Tang is with the Department of Advanced Computing Sciences, Maastricht University, Maastricht 6211 LK, The Netherlands. guangzhi.tang@maastrichtuniversity.nl

This publication is part of the project Brain-inspired MatMul-free Deep Learning for Sustainable AI on Neuromorphic Processor with file number NGF.1609.243.044 of the research programme AiNed XS Europe which is (partly) financed by the Dutch Research Council (NWO) under the grant https://doi.org/10.61686/MYMVX53467.

Code of the paper is available in https://github.com/ERNIS-LAB/CSSL-Event-Object-Detection

tional efficiency while improving task-specific performance.

Our key contributions are as follows:

- Context-aware sparse spatiotemporal learning framework: We introduced a novel context-aware learning approach that dynamically adjusts activation thresholds based on contextual information, enhancing spatial sparsity while preserving key event-driven features.
- Efficient spatiotemporal learning: We extended context-aware thresholding to both convolutional layers and convolutional recurrent units, enabling efficient processing of event-based data.
- Generalization to multiple tasks: We applied CSSL to event-based object detection and optical flow estimation, demonstrating state-of-the-art computation efficiency with minimal performance loss.
- Computational efficiency: Our experiment results show that CSSL significantly reduces computational overhead while maintaining accuracy, making it well-suited for real-time robotic applications.

II. Methods

A. Overview of CSSL

Event-based convolution is a specialized approach designed for sparse, asynchronous data processing in neuromorphic systems. Unlike standard convolutional operations, which process all pixels in a frame simultaneously, event-based convolution operates on individual spikes (events), significantly reducing computational redundancy and improving efficiency. Traditional frame-based convolutions store and process all pixels uniformly, requiring large memory overheads. In contrast, event-driven depth-first convolution prioritizes processing events as they arrive, integrating information incrementally and releasing memory as soon as it is no longer needed. This technique ensures efficient computation while keeping memory requirements low, making it particularly well-suited for event-based vision applications with dynamic and sparse input patterns.

Building upon event-based convolution [11], we proposed Context-aware Sparse Spatiotemporal Learning (CSSL), a novel framework that introduces context-aware thresholding to dynamically modulate the activation of 2D convolution blocks. To enhance sparsity within convolution computing, we introduce this context-aware thresholding mechanism that adaptively filters activations based on input contextual features. Unlike traditional activation functions (e.g. ReLU) that apply a fixed threshold, our method dynamically determines the threshold value based on the input distribution. The threshold is derived by applying convolution to the input feature, ensuring that its spatial dimensions remain consistent with those of the output feature. The CSSL approach is highly flexible and can be applied to both standard convolutional layers and convolutional recurrent units, making it a generalizable solution for event-based visual processing.

B. Context-aware Threshold for 2D Convolution

Compared to the normal 2D convolution layer generating dense hidden output feature, the context-aware thresholding mechanism generates key events to keep output sparse. Specifically, given an input feature map x, the threshold $v_{th}^{(t)}$ is computed as Equation 1:

$$\begin{aligned} v_{th}^{(t)} &= \sigma\left(W_v x^{(t)} + b_v\right) \\ \tilde{y}^{(t)} &= W_x x^{(t)} + b_x \\ s^{(t)} &= \mathrm{H}\left(\tilde{y}^{(t)} - v_{th}^{(t)}\right) \\ y^{(t)} &= s^{(t)} \odot \tilde{y}^{(t)} \end{aligned} \tag{1}$$

where W_v and b_v denote convolution kernel and bias which computes threshold from input feature $x^{(t)}$. The terms W_x and b_x denote the convolution parameters that process the input feature into an intermediate representation $\tilde{y}^{(t)}$. The symbol $\odot$ denotes element-wise product and $\sigma(\cdot)$ denotes the sigmoid function ensuring that the computed threshold $v_{th}^{(t)}$ are projected into the normalized range of [0,1] and the sparse output feature is positive. The context-aware convolution unit dynamically regulates activations by applying a pixel-wise threshold to the processed feature map. As shown in Figure 2(a), the event activation mask $s^{(t)}$ is generated using a Heaviside step function $H(\cdot)$ every time step which selectively retains informative features while filtering out less relevant activations. This mechanism enhances spatial sparsity by suppressing unnecessary computations while preserving meaningful event-based information. We used a surrogate gradient to estimate the backpropagated gradient of $H(\cdot)$ using the same methods presented in [12].

Given that event-based inputs are inherently spatially sparse, maintaining a dense output across the network is both computationally inefficient and unnecessary for effective feature learning. Instead of applying a uniform threshold across all activations, our proposed pixel-wise context-aware thresholding dynamically adjusts the activation threshold at each spatial location based on the local input context. This ensures that only informative event-driven activations are retained, while irrelevant or noisy activations are suppressed. By leveraging this fine-grained sparsity control, our method enhances computational efficiency and optimizes feature representation, making it particularly well-suited for event-based vision tasks where the relevant information is non-uniformly distributed across the input space.

C. Context-aware Threshold for Residual block

We integrate context-aware thresholding with residual block as shown in Figure 2(b). To accommodate the sparse input feature in the second convolutional layer, the first convolutional layer is replaced with a context-aware convolution. The output from the second convolution is then split channel-wise into two components: a threshold and a dense output feature. The dense output feature is accumulated with input $x^{(t)}$ with the residual connection. Unlike conventional residual blocks, which apply neuron-wise additions between the input and processed feature, our method introduces an additional thresholding step after accumulation. This ensures that the final output remains sparse, effectively reducing redundant activations.

D. Context-aware Threshold for Event-based Convolutional Recurrent Learning

To explore the spatiotemporal sparsity in neural network learning, we extended and generalized our context-aware thresholding to the sparse convolutional recurrent unit [8]. As illustrated in Figure 2(c), we primarily employed the Minimal Gated Unit (MGU) [13] for our experiments within the context-aware convolution recurrent unit, following the event recurrent processing method in [12]. The context-aware thresholding can also be applied to other sparse convolutional recurrent units using the same approach.

As shown in Equation 2, the threshold is obtained by applying convolution to the sparse hidden state from the previous time step $y^{(t-1)}$. This design choice is motivated by the fact that in event-based vision, objects can remain stationary, causing the event camera to cease generating new events. In such scenarios, a threshold derived from the memory cell ensures that the network selectively retains relevant activations, allowing it to produce accurate outputs even in the absence of new input events.

Since the output features resulting from the convolution of the input $x^{(t)}$ do not participate in subsequent convolutions, the dense tensor has a negligible impact on the total number of synaptic operations (SOp). Standard convolution can be used without thresholding mechanism. We added an auxiliary hidden state $c^{(t)}$ to the unit, and $y^{(t)}$ is generated in event-based form as sparse output of the unit.

$$f^{(t)} = \sigma\left(W_{xf}x^{(t)} + W_{yf}y^{(t-1)} + b_f\right) \tag{2a}$$

$$v_{th}^{(t)} = \sigma\left(W_v y^{(t-1)} + b_v\right) \tag{2b}$$

$$\tilde{h}^{(t)} = \tanh\left(W_{hi}\left(f^{(t)} \odot y^{(t-1)}\right) + W_{xh}x^{(t)} + b_h\right) \tag{2c}$$

$$c^{(t)} = \left(1 - f^{(t)}\right) \odot c^{(t-1)} + f^{(t)} \odot \tilde{h}^{(t)} \tag{2d}$$

$$s^{(t)} = \mathrm{H}\left(c^{(t)} - v_{th}^{(t)}\right) \tag{2e}$$

$$y^{(t)} = c^{(t)} \odot s^{(t)} \tag{2f}$$

Inspired by spiking neurons [14] to help to forget, we add a soft-reset mechanism to $c^{(t)}$ after the conv-rec unit generating $y^{(t)}$ as shown in Equation 3:

$$c^{(t)} = c^{(t)} - v_{th}^{(t)} \odot s^{(t)} \tag{3}$$

By dynamically adjusting the hidden state based on the learned context-aware threshold, network maintains sparse yet informative representations over time.

E. Apply to robotics application

To evaluate the generalizability of the proposed CSSL framework, we applied it to two fundamental event-based vision tasks: object detection and optical flow estimation. By incorporating CSSL into different neural architectures, we evaluated its ability to enhance computational efficiency and task performance across diverse robotic perception applications. Specifically, we replaced standard convolution layers with context-aware convolutions, introduced context-aware thresholding in residual and recurrent blocks.

Our experiments are designed to achieve two primary objectives. First, we demonstrated that simply replacing standard convolutional units with our context-aware modules leads to significant performance improvements across different tasks. Second, we showed that our context-aware approach enables the network to naturally achieve high sparsity and robust performance, even without the need for specialized sparsity regularization techniques.

III. Experiments and Results

A. Experimental Setup

1) Event-based Object Detection: For the event-based object detection task, our method was evaluated on the 1 Mpx [15] and Gen1 [16] datasets, adhering to the standard evaluation approach outlined in [15]. Performance is reported by widely used COCO mAP metric [17]. The CSSL framework incorporated MGU within Conv-Rec modules, except in Section III-C, where alternative recurrent unit were explored. The best-performing model on the validation set was subsequently applied to the test set to obtain the final mAP score.

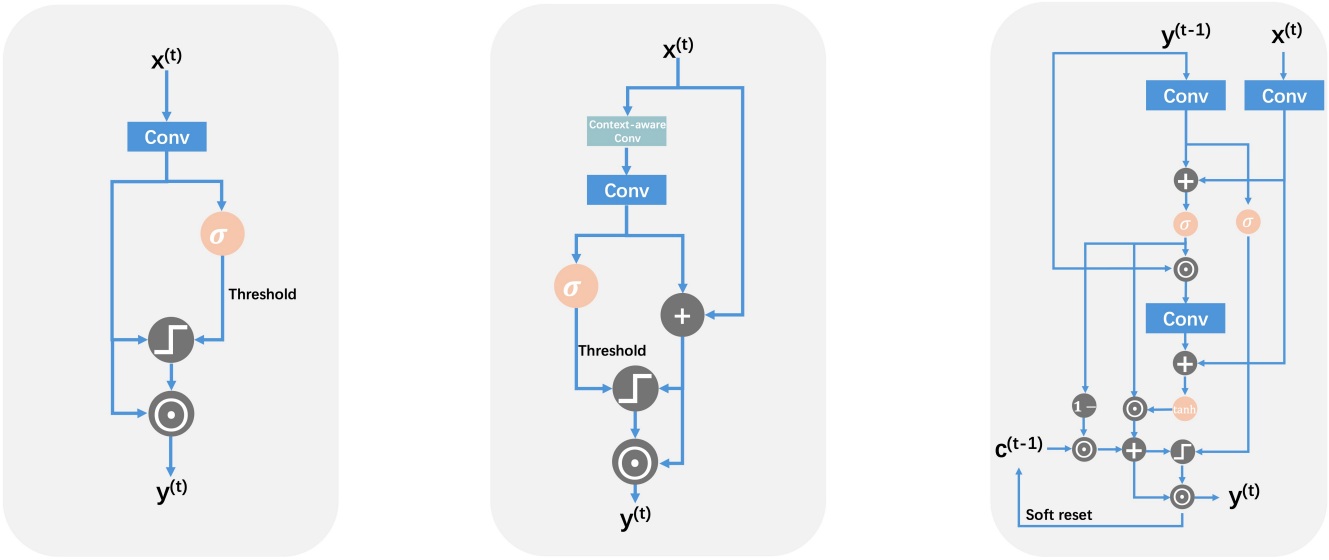
We validated the proposed method using the Sparse Event-based Efficient Detector (SEED) architecture [8], which includes the backbone and the SSD detection head [18]. The architecture of the backbone is shown in Table I. The CSSL-SEED network backbone comprises three key components: the context-aware convolution module, the context-aware residual block, the context-aware conv-rec block.

TABLE I
BACKBONE ARCHITECTURE OF SEED-256

Layer	Channels out	Dimension out(1Mpx)
CSSL Conv	32	[320, 180]
CSSL Residual Connection	64	[160, 90]
CSSL Residual Connection	64	[160, 90]
CSSL Residual Connection	128	[160, 90]
CSSL ConvMGU	256	[80, 45]
CSSL ConvMGU	256	[40, 23]
CSSL ConvMGU	256	[20, 12]

Using ADAM optimizer, all models are trained with full precision with the OneCycle scheduler to adjust the learning rate dynamically. For 1Mpx dataset, models are trained for 50 epochs with the maximum learning rate of 3e-4 and for Gen1 datasets, models are trained for 60 epochs with the maximum learning rate set at 2.5e-4. Using a NVIDIA A100 GPU, we trained our models with a batch size of 8, sequence length of 15 on 1Mpx dataset. The training takes approximately 4 days on a single A100 GPU. We trained our models with a batch size of 5, sequence length of 40 on the Gen1 dataset. The training takes approximately 3 days on a single A100 GPU.

2) Event-based Optical Flow: For the event-based optical flow estimation task, we adopted the same neural network



(a) Context-aware threshold for convolution layer

(b) Context-aware threshold for residual block

(c) Context-aware threshold for Conv-recurrent block

Fig. 2. Illustration of the context-aware thresholding process. The input feature is processed through a context-aware convolution that generates an adaptive threshold. This threshold selectively filters activations, allowing only informative signals to propagate while suppressing irrelevant ones. Compared to traditional fixed-threshold activation functions (e.g., ReLU), CSSL dynamically adjusts the threshold based on input distribution, enhancing spatial sparsity in convolutional layers and spatiotemporal sparsity in recurrent architectures.

architecture as EV-FlowNet in [7]. We integrated our context-aware modules by replacing standard convolution layers with context-aware convolution, conventional residual blocks with context-aware residual blocks, and ConvGRU with context-aware MGU blocks. Performance was assessed using Average Endpoint Error (AEE) and outlier percentage metrics to evaluate both accuracy and robustness.

We followed the experimental setup outlined in [7]. The network architecture, training strategies, and evaluation protocols remain consistent with [7]. The AdamW optimizer was employed, with a OneCycle learning rate scheduler dynamically adjusting the learning rate during training, with a maximum learning rate of 2e-4. The model was trained for 100 epochs on the UZH-FPV dataset [19]. Inference and evaluation were conducted on the MVSEC dataset [20]. The training takes approximately 10 hours on a single RTX4090 GPU.

B. Experiments results

We benchmarked our model against state-of-the-art methods for event-based vision in Table II for object detection and Table III for optical flow estimation.

In the event-based object detection task, CSSL significantly reduces synaptic operations while preserving high detection accuracy. As shown in Table II, the CSSL-based models outperform conventional architectures in terms of computational efficiency, achieving comparable or superior mean Average Precision (mAP) scores with significantly lower GSOp. Notably, the CSSL-MGU model achieves an mAP of 46.4 on the 1Mpx dataset while requiring only 2.8 GSOp, which is only 32.2% of RVT-S [21]. Compared to SNN-based solutions, our model outperforms state-of-the-art methods in mAP while requiring only 7.4% of the synaptic operations. This result indicates that CSSL effectively balances accuracy and efficiency by dynamically filtering activations, thereby reducing redundant computations in dense event-based input streams.

Similarly, in the event-based optical flow estimation task, our CSSL-integrated models demonstrate superior computational efficiency compared to conventional recurrent-based architectures. As summarized in Table III, the fully CSSL-integrated CSSL-EV-FlowNet achieves an AEE of 2.38, outperforming RNN-EV-FlowNet [7] with specialized FA-TReLU sparsification, and only 62.8% neuron density compared to it.

Our experimental results demonstrate the effectiveness of CSSL in improving both computational efficiency and task performance across event-based object detection and optical flow estimation tasks. The integration of context-aware thresholding in convolutional and recurrent modules enables our models to selectively retain informative activations, reducing computational redundancy while maintaining accuracy. The sparsity-driven approach of CSSL allows for a more efficient allocation of computational resources, making it highly suitable for real-time robotic applications where power and memory efficiency are critical.

C. Generalizing to variations of recurrent unit

We extended the CSSL method to various recurrent architectures to validate its generalizability, specifically applying it to MinimalRNN [26] and GRU [27]. These context-aware recurrent units, similar to the approach described in Section II-D, dynamically generate pixel-wise thresholds from the output feature of the convolution applied to the

TABLE II

COMPARING CSSL-SEED WITH STATE-OF-THE-ART OBJECT DETECTION APPROACHES FOR EVENT-BASED VISION

		1 Mpx		Gen1		
Method	Network	mAP	GSOp	mAP	GSOp	Param(M)
ASTMNet [22]	TACN+ConvRec+SSD	48.3	-	46.7	-	>39.6
SNN [23]	Spiking DenseNet+SSD	-	-	18.9	2.33	8.2
SpikeYOLO [24]	Spiking+YOLOv8	-	-	40.4	14.3	23.1
RED [15]	SENet+ConvLSTM+SSD	43.0	26.1	40.0	8.26	24.1
RVT-B [21]	MaxViT+LSTM+YOLOX	47.4	15.6	47.2	5.05	18.5
RVT-S [21]	MaxViT+LSTM+YOLOX	44.1	8.69	46.5	2.78	9.9
RVT-T [21]	MaxViT+LSTM+YOLOX	41.5	3.87	44.1	1.29	4.4
SEED-256 [8]	ECNN+EConvGRU+SSD	44.9	3.83	45.3	1.32	13.9
SEED-128 [8]	ECNN+EConvGRU+SSD	44.1	2.75	44.5	0.99	4.8
CSSL-SEED-256	CSSL-Conv+CSSL-ConvMGU+SSD	46.4	2.80	46.3	1.06	10.7
CSSL-SEED-256	CSSL-Conv+CSSL-ConvGRU+SSD	46.2	3.42	46.4	1.22	13.9
CSSL-SEED-256	CSSL-Conv+CSSL-ConvMinimalRNN+SSD	44.8	2.71	45.5	0.99	7.9

TABLE III

COMPARING CSSL-EV-FLOWNET WITH STATE-OF-THE-ART OPTICAL FLOW ESTIMATION APPROACHES FOR EVENT-BASED VISION

Network	outdoor_day1		indoor_flying_1		indoor_flying2		indoor_flying3		Average		
	AEE	$\%_{outlier}$	AEE	$\%_{outlier}$	AEE	$\%_{outlier}$	AEE	$\%_{outlier}$	AEE	$\%_{outlier}$	Dens.(%)
EV-FlowNet(GRU) [25]	1.69	12.50	2.16	21.51	3.90	40.72	3.00	29.60	2.94	29.35	-
RNN-EV-FlowNet [7]	1.69	12.96	2.02	18.74	3.84	38.17	2.97	27.91	2.88	27.32	16.90
CSSL-EV-FlowNet	1.55	11.64	1.84	14.84	3.43	33.77	2.68	25.01	2.38	21.31	10.61

previous hidden state $y^{(t-1)}$ at the current time step, as formulated in Equation 2b. Additionally, we introduced an auxiliary hidden state $c^{(t)}$ in Equation 2d and applied the thresholding mechanism to regulate its updates, ensuring the generation of events, as described in Equation 2e. However, we did not extend our method to LSTM due to its additional memory cell, which introduces a separate gating mechanism. This additional memory cell complicates the direct integration of our context-aware thresholding approach, making it less compatible with the sparsity-driven processing strategy employed in CSSL.

We evaluated our method on the event-based object detection task on the 1Mpx and Gen1 datasets, with the results presented in Table II. Notably, GRU achieves comparable performance to MGU while requiring slightly higher synaptic operations and parameter count. In contrast, MinimalRNN exhibits lower performance compared to both MGU and GRU but benefits from a reduced model size, highlighting a trade-off between efficiency and accuracy.

D. Effectiveness of context-aware sparse learning

To assess the effectiveness of our proposed context-aware thresholding method, we compared it with a baseline model that employs standard ReLU activation in all straight-forward convolutional layers and incorporates a sparsity loss function, as shown in Equation 4, where β_{sparse} is the weighting coefficient of the sparsity loss, n represents the index of a training sample, N denotes the mini-batch size, t corresponds to the step index within a training sample, T represents the total number of steps per sample, l indicates the index of the layer, L is the overall number of layers, and $\mathbf{x}$ refers to the activation maps.

$$L_{sparse} = \beta_{sparse} \frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{l=1}^{L} ||\mathbf{x}^{(n)(t)(l)}||_1 \quad (4)$$

We employed a two-stage training strategy to evaluate the impact of sparsity constraints on model performance and efficiency. The first stage follows the same setup as described in Section III-A, with training conducted without the sparsity loss constraint. In the second stage, we introduced the sparsity loss and vary its values to assess its effect on performance. Throughout all experiments, we used MGU as the recurrent unit for all convolution recurrent modules. The results are summarized in Table IV.

TABLE IV

RESULTS WITH SPARSITY LOSS

	1Mpx			
	Conv + ReLU + L1		CSSL + L1	
β_{sparse}	mAP	GSOp	mAP	GSOp
1	37.4	1.85	40.0	1.54
0.1	44.8	3.38	46.1	2.17
0.04	44.9	5.18	46.3	2.63
0.01	45.1	5.66	46.2	2.75
0.001	44.3	5.84	46.0	2.82
W/O β_{sparse}	45.8	5.95	46.4	2.8

First, we evaluated a baseline model that replaces our context-aware thresholding with a straightforward convolution that incorporates the ReLU activation function. As observed in Table IV, this modification leads to a decrease in mAP performance in the 1Mpx dataset by 0.6, with synaptic operation increasing by 112.5% compared to our proposed CSSL approach. These findings indicate that our

TABLE V
RESULT COMPARISON OF AVERAGE NEURAL ACTIVATION DENSITY OF EACH SEED LAYER

	Average Neural Activation Density (with mAP)		
	CSSL without sparsity loss (46.4)	SEED without sparsity loss (45.0)	SEED + L1 (0.1) sparsity loss (44.2)
Conv1	0.47	0.66	0.52
Res1_Conv1	0.17	0.61	0.53
Res1_Conv2	0.20	0.72	0.45
Res2_Conv1	0.10	0.56	0.50
Res2_Conv2	0.46	0.84	0.66
Res3_Conv1	0.07	0.43	0.36
Res3_Conv2	0.12	0.50	0.38
Recurrent1	0.08	0.05	0.05
Recurrent2	0.07	0.05	0.04
Recurrent3	0.06	0.04	0.04

context-aware thresholding mechanism not only improves computational efficiency but also enhances the model's ability to extract meaningful features. Additionally, we examined the effect of different β_{sparse} values on training outcomes. The results highlight that the choice of β_{sparse} significantly influences the detection performance. This underscores the advantage of our CSSL method, which achieves optimal performance in a single-stage training process without fine-tuning sensitive hyperparameters.

Furthermore, we investigated whether adding a sparsity loss directly to a pre-trained CSSL model could further reduce the number of synaptic operations. As shown in Table IV, when setting β_{sparse} to 0.04, the model maintains nearly the same performance while achieving a 6% reduction in synaptic operations. However, when increasing β_{sparse} further in an attempt to further decrease synaptic operations, we observed a notable performance degradation. These findings suggest that our proposed CSSL approach inherently achieves a near-optimal trade-off between performance and computational efficiency in the first-stage training itself, without requiring additional sparsity constraints. This demonstrates the effectiveness of our context-aware thresholding mechanism in maximizing performance while minimizing computation cost.

IV. COMPARING PER-LAYER ACTIVATION DENSITY

To further assess how context-aware thresholding enhances activation sparsity, we measured the mean activation density for every layer of the SEED-256 architecture with and without using the proposed CSSL framework (Table V). The activation densities were averaged over the test split of the 1Mpx object-detection dataset. Relative to the baseline SEED model trained with a sparsity loss, our proposed CSSL approach lowers the activation density in convolutional layers, without requiring an additional, compute-intensive fine-tuning phase.

V. DISCUSSION AND CONCLUSION

This paper presents CSSL, a computationally efficient framework for event-based vision. By applying context-aware thresholding mechanism, CSSL effectively reduces neuron activation density in convolutional modules, improving computation efficiency and task performance. Unlike traditional sparsity-driven networks that rely on explicit sparsity regularization, CSSL naturally achieves high activation sparsity by dynamically adapting thresholds based on the input distribution. This eliminates the need for manually tuned sensitive sparsity hyperparameters, making training more stable and efficient. Our approach directly addresses the challenge of high-cost spatiotemporal processing, setting a new benchmark for efficient event-based object detection and optical flow estimation for neuromorphic methods.

As neuromorphic processors increasingly support deep learning models [28], [29], [30], CSSL is positioned to substitute spiking neural networks on neuromorphic processors with superior performance while maintaining low energy consumption.

Beyond object detection and optical flow, the principles of context-aware thresholding can be extended to various event-based tasks, including motion prediction, agile flying, and autonomous navigation. Integrating CSSL with recent advancements in event-based preprocessing [31], [32] could further enhance its adaptability across diverse applications. Additionally, the potential for ultra-fast inference with minimal neuron activation makes CSSL an ideal option for real-time edge deployment in high-speed robotics and automotive systems, where efficiency and responsiveness are important.

REFERENCES

[1] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[2] D. Gehrig and D. Scaramuzza, "Low latency automotive vision with event cameras," 2024.

[3] A. Bhattacharya, M. Cannici, N. Rao, Y. Tao, V. Kumar, N. Matni, and D. Scaramuzza, "Monocular event-based vision for obstacle avoidance with a quadrotor," 2024. [Online]. Available: https://arxiv.org/abs/2411.03303

[4] D. Neil and S.-C. Liu, "Effective sensor fusion with event-based sensors and deep network architectures," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 2282–2285.

[5] J. Yik, K. Van den Berghe, D. den Blanken, Y. Bouhadjar, M. Fabre, P. Hueber, W. Ke, M. A. Khoei, D. Kleyko, N. Pacik-Nelson, *et al.*, "The neurobench framework for benchmarking neuromorphic computing algorithms and systems," *Nature communications*, vol. 16, no. 1, p. 1545, 2025.

[6] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/82f2b308c3b01637c607ce05f52a2fed-Paper.pdf

[7] Y. Xu, G. Tang, A. Yousefzadeh, G. C. de Croon, and M. Sifalakis, "Event-based optical flow on neuromorphic processor: Ann vs. snn comparison based on activation sparsification," *Neural Networks*, vol. 188, p. 107447, 2025.

[8] S. Wang, Y. Xu, A. Yousefzadeh, S. Eissa, H. Corporaal, F. Corradi, and G. Tang, "Sparse convolutional recurrent learning for efficient event-based neuromorphic object detection," *arXiv preprint arXiv:2506.13440*, 2025.

[9] E. Bengio, P.-L. Bacon, J. Pineau, and D. Precup, "Conditional computation in neural networks for faster models," *arXiv preprint arXiv:1511.06297*, 2015.

[10] Z. Liu, J. Wang, T. Dao, T. Zhou, B. Yuan, Z. Song, A. Shrivastava, C. Zhang, Y. Tian, C. Re, and B. Chen, "Deja vu: Contextual sparsity for efficient LLMs at inference time," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 22 137–22 176. [Online]. Available: https://proceedings.mlr.press/v202/liu23am.html

[11] Y. Xu, K. Shidqi, G.-J. van Schaik, R. Bilgic, A. Dobrita, S. Wang, R. Meijer, P. Nembhani, C. Arjmand, P. Martinello, A. Gebregiorgis, S. Hamdioui, P. Detterer, S. Traferro, M. Konijnenburg, K. Vadivel, M. Sifalakis, G. Tang, and A. Yousefzadeh, "Optimizing event-based neural networks on digital neuromorphic architecture: a comprehensive design space exploration," *Frontiers in Neuroscience*, vol. 18, 2024. [Online]. Available: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2024.1335422

[12] A. Subramoney, K. K. Nazeer, M. Schöne, C. Mayr, and D. Kappel, "Efficient recurrent architectures through activity sparsity and sparse back-propagation through time," in *The Eleventh International Conference on Learning Representations*, 2022.

[13] G.-B. Zhou, J. Wu, C.-L. Zhang, and Z.-H. Zhou, "Minimal gated unit for recurrent neural networks," *International Journal of Automation and Computing*, vol. 13, no. 3, pp. 226–234, 2016.

[14] Y. Guo, Y. Chen, L. Zhang, Y. Wang, X. Liu, X. Tong, Y. Ou, X. Huang, and Z. Ma, "Reducing information loss for spiking neural networks," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 36–52.

[15] E. Perot, P. De Tournemire, D. Nitti, J. Masci, and A. Sironi, "Learning to detect objects with a 1 megapixel event camera," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 639–16 652, 2020.

[16] P. De Tournemire, D. Nitti, E. Perot, D. Migliore, and A. Sironi, "A large scale event-based detection dataset for automotive," *arXiv preprint arXiv:2001.08499*, 2020.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[19] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6713–6719.

[20] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3d perception," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018.

[21] M. Gehrig and D. Scaramuzza, "Recurrent vision transformers for object detection with event cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 884–13 893.

[22] J. Li, J. Li, L. Zhu, X. Xiang, T. Huang, and Y. Tian, "Asynchronous spatio-temporal memory network for continuous event-based object detection," *IEEE Transactions on Image Processing*, vol. 31, pp. 2975–2987, 2022.

[23] L. Cordone, B. Miramond, and P. Thierion, "Object detection with spiking neural networks on automotive event data," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.

[24] X. Luo, M. Yao, Y. Chou, B. Xu, and G. Li, "Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection," in *Computer Vision – ECCV 2024*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds. Cham: Springer Nature Switzerland, 2025, pp. 253–272.

[25] J. Hagenaars, F. Paredes-Vallés, and G. De Croon, "Self-supervised learning of event-based optical flow with spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7167–7179, 2021.

[26] M. Chen, "Minimalrnn: Toward more interpretable and trainable recurrent neural networks," *arXiv preprint arXiv:1711.06788*, 2017.

[27] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[28] J. Pei, L. Deng, S. Song, M. Zhao, Y. Zhang, S. Wu, G. Wang, Z. Zou, Z. Wu, W. He, *et al.*, "Towards artificial general intelligence with hybrid tianjic chip architecture," *Nature*, vol. 572, no. 7767, pp. 106–111, 2019.

[29] C. Liu, G. Bellec, B. Vogginger, D. Kappel, J. Partzsch, F. Neumärker, S. Höppner, W. Maass, S. B. Furber, R. Legenstein, *et al.*, "Memory-efficient deep learning on a spinnaker 2 prototype," *Frontiers in neuroscience*, vol. 12, p. 840, 2018.

[30] C. D. Schuman, S. R. Kulkarni, M. Parsa, J. P. Mitchell, B. Kay, *et al.*, "Opportunities for neuromorphic computing algorithms and applications," *Nature Computational Science*, vol. 2, no. 1, pp. 10–19, 2022.

[31] N. Zubić, D. Gehrig, M. Gehrig, and D. Scaramuzza, "From chaos comes order: Ordering event representations for object recognition and detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 12 846–12 856.

[32] Y. Peng, Y. Zhang, P. Xiao, X. Sun, and F. Wu, "Better and faster: Adaptive event conversion for event-based object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, 2023, pp. 2056–2064.