SPIKERED: FULLY SPIKE-DRIVEN RGB-EVENT OBJECT DETECTION WITH ASYMMETRIC SPIKING FUSION AND BOX-GUIDED QUERY REFINEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs) offer substantial potential for energy-efficient visual detection. However, existing spiking object detectors primarily rely on unimodal inputs, while existing multimodal spiking approaches typically retain ANN components, limiting their energy-efficiency advantages. To address this issue, we propose SpikeRED, a fully spike-driven RGB-event object detection framework. Its Asymmetric Spiking Fusion (ASF) module enables more effective Event-to-RGB interaction with lower energy overhead. The Box-Guided Spiking Decoder further enhances the representation capability of spiking object queries through reference-box guidance. Experiments on DSEC-Det and PKU-DAVIS-SOD demonstrate that SpikeRED achieves performance comparable to leading multimodal ANN detectors while substantially reducing theoretical inference energy. To our knowledge, SpikeRED is the first fully spike-driven query-based multimodal object detection framework.

1 INTRODUCTION

Spiking neural networks (SNNs) are a promising energy-efficient alternative to conventional artificial neural networks (ANNs). Unlike ANNs, which propagate information mainly through dense matrix multiplications, SNNs emulate neural dynamics and communicate through binary spikes. This spike-based inter-layer communication replaces dense matrix multiplications with sparse accumulations, improving energy efficiency (Yao et al., 2023).

Recent SNN-based vision models address image classification (Zhou et al., 2022), object tracking (Zhang et al., 2022b), object segmentation (Parameshwara et al., 2021), and object detection. SpikeYOLO (Luo et al., 2024) and EMS-YOLO (Su et al., 2023), for example, demonstrate the potential of SNNs for detection. Yet existing spiking detectors primarily use unimodal inputs, making them vulnerable to information loss from a single sensor under extreme illumination or motion blur. EOLO (Cao et al., 2024b) explores low-power multimodal detection by jointly modeling RGB frames and events, using an SNN for event features. However, only its event branch is spiking: its RGB branch, fusion modules, and detection head still rely on ANN computation. This prevents end-to-end spike-driven processing and limits the benefits of sparse computation. Moreover, its experiments mainly use exposure-variation benchmarks derived from VOC and COCO, with limited validation on real-world RGB-event driving datasets and complex traffic scenes. Its energy analysis primarily covers the SNN event backbone, lacking an end-to-end comparison of complete multimodal detectors. Its symmetric fusion also performs both Event-to-RGB and RGB-to-Event interactions, adding computational overhead that further reduces the energy-efficiency benefits of multimodal SNNs.

To address these limitations, we propose SpikeRED, a fully spike-driven detection framework with a query-based spiking decoder for complex scenarios such as autonomous driving. To reduce the energy overhead of bidirectional multimodal fusion, we design Asymmetric Spiking Fusion (ASF), which performs directional Event-to-RGB fusion at low spatial resolutions.

We also propose a Box-Guided Spiking Decoder to enhance the limited representations of low-timestep spiking object queries. It leverages reference boxes to guide sparse multiscale feature

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

sampling and injects box-aligned local features into the queries through residual enhancement, providing stronger visual evidence while preserving iterative query refinement.

Our main contributions are as follows:

1. We design ASF, which improves detection accuracy over EOLO's bidirectional fusion while reducing whole-network theoretical inference energy by 22.7% on DSEC-Det and 26.5% on PKU-DAVIS-SOD.
2. We design a Box-Guided Spiking Decoder that injects reference-box-aligned, spiking-gated multiscale features as a residual correction before each decoder layer. It improves mAP over Plain Spiking Attention by 6.8% and 6.9% on the two datasets, respectively.
3. These designs form SpikeRED, to our knowledge the first fully spike-driven query-based multimodal object detector. SpikeRED-L achieves 30.7% and 32.2% mAP on DSEC-Det and PKU-DAVIS-SOD, respectively, outperforming the multimodal ANN detector SFNet while reducing theoretical inference energy by over 85%. SpikeRED-S/M also outperform the hybrid model EOLO at lower energy cost. With RGB-only inputs, SpikeRED-L* improves on SpikeYOLO by up to 2.2% mAP.

2 Related Work

2.1 SNNs for Visual Tasks

Recent advances in spiking neural networks have substantially improved their capability for visual representation learning. Early visual SNNs largely relied on ANN-to-SNN conversion, whereas surrogate-gradient learning (Neftci et al., 2019) has enabled direct training with only a few time steps and facilitated the development of increasingly sophisticated architectures, from convolutional networks to Transformers. Spikformer (Zhou et al., 2022) introduces softmax-free self-attention with spiking queries, keys, and values. Spike-Driven Transformer (Yao et al., 2023) further replaces expensive floating-point operations with spike-driven sparse accumulations, demonstrating the representational capacity of spiking Transformers. These studies provide the architectural foundation for our spiking backbones and attention modules, but mainly focus on classification tasks.

2.2 Query-Based Object Detection Methods

DETR (Carion et al., 2020) formulates object detection as a set prediction problem based on learnable object queries, while RT-DETR (Zhao et al., 2024) initializes queries with high-quality proposals selected from encoder outputs for real-time object detection. Subsequent methods further exploit reference or predicted boxes to guide iterative query refinement. DAB-DETR (Liu et al., 2022) represents positional queries as dynamically updated anchor boxes and uses their widths and heights to modulate spatial priors in cross-attention. REGO (Chen et al., 2022) extracts RoI-aligned local features around predicted boxes and progressively refines detection results through additional recurrent decoding stages. These methods demonstrate that predicted boxes can provide useful spatial guidance for query refinement. In low-timestep SNNs, sparse spike activations limit the visual content encoded in object queries, making target-specific local details harder to capture during decoding. We therefore use reference boxes to guide local feature enhancement within the existing decoder layers, without introducing additional refinement decoders as in REGO.

2.3 SNNs for RGB-Event Object Detection

Spiking-YOLO (Kim et al., 2020), EMS-YOLO (Su et al., 2023), and SpikeYOLO (Luo et al., 2024) have advanced spiking object detection, but mainly focus on unimodal inputs. Recent works have begun to incorporate event information into spiking detection frameworks. EOLO (Cao et al., 2024b) introduces an SNN event branch but retains ANN-based RGB, fusion, and detection modules. EGSTNet (Wang et al., 2025a) combines a spiking backbone with ANN DINO-DETR modules, while LOD-Drive (Han et al., 2026) adopts hybrid spiking motion encoding. In contrast, SpikeRED extends spike-driven computation to both modality backbones, cross-modal fusion, and the query-based detector.

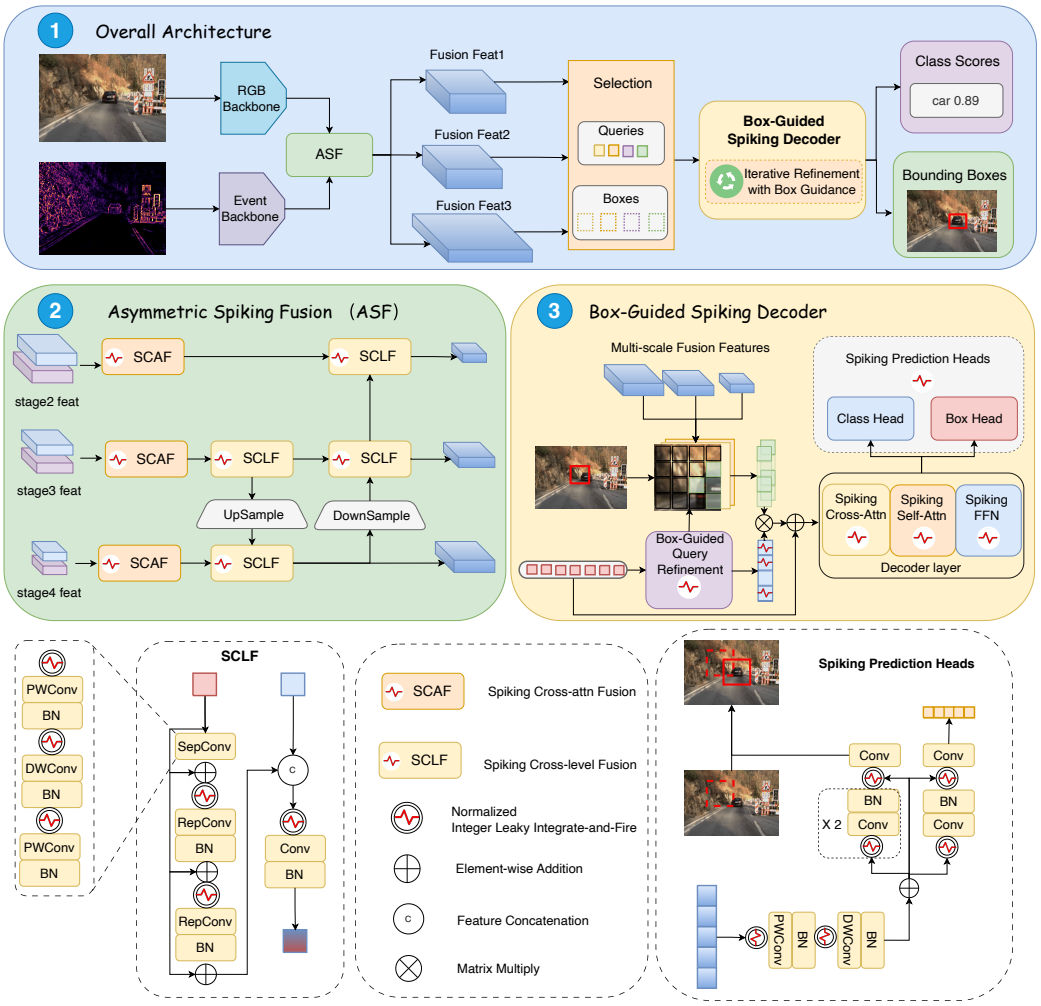

Figure 1: Overview of SpikeRED. ASF performs directional Event-to-RGB fusion, while box-guided query refinement injects reference-box-aligned regional features before each spike-driven decoder layer for iterative detection refinement.

3 METHOD

3.1 OVERALL ARCHITECTURE

As shown in Figure 1, SpikeRED consists of three stages: **(A) Dual-branch spiking feature extraction**, which uses separate backbones for RGB and events; **(B) Asymmetric Spiking Fusion (ASF)**, which transfers event information to the RGB branch; and **(C) Box-guided query decoding and prediction**, which combines the Box-Guided Spiking Decoder with a dual-branch spiking detection head. Each RGB frame is paired with a spatially aligned three-channel event representation. Independent spiking backbones extract four feature levels from each modality. ASF fuses event features directionally into RGB semantic features at the last three levels to produce candidate tokens. We score these tokens and select the top N_q to initialize object queries and reference boxes (Zhao et al., 2024). In each decoder layer, reference boxes guide queries to sample and aggregate local visual features for residual enhancement. The enhanced queries then pass through spiking cross-attention, self-attention, and feed-forward layers. The spiking detection head updates the reference boxes, providing more accurate spatial priors for the next layer and enabling iterative refinement of queries, regional features, and reference boxes.

3.2 SPIKING NEURON MODEL

We use the Normalized Integer Leaky Integrate-and-Fire (NI-LIF) neuron proposed in Spike2Former (Lei et al., 2025) as the basic spiking unit. NI-LIF replaces a single binary firing event with multilevel discrete activations, increasing information capacity with few simulation steps. At inference, integer activations can be equivalently expanded into binary spikes, preserving sparse spike-based communication.

For a continuous input X, the multilevel spiking activation rounds and clips it to $[0, D]$, then normalizes it by the maximum firing level D:

$$I = \mathrm{clip}(\mathrm{round}(X), 0, D), \qquad S = \frac{I}{D}. \tag{1}$$

Thus, $I \in \{0, 1, \ldots, D\}$ and $S \in \{0, 1/D, \ldots, 1\}$. At inference, the integer activation I can be equivalently expanded into D binary spikes:

$$I = \sum_{d=1}^{D} s^d, \qquad S = \frac{1}{D}\sum_{d=1}^{D} s^d, \qquad s^d \in \{0, 1\}. \tag{2}$$

In the following equations, we denote the NI-LIF activation defined above by $\mathrm{Spike}(\cdot)$.

3.3 ASYMMETRIC SPIKING FUSION

ASF exploits event motion and edge cues while avoiding the overhead of bidirectional interaction. Before fusion, two Spike-Driven Transformer v3 (SDT v3) backbones (Yao et al., 2025), with identical topologies but independent parameters, encode RGB images and event voxels. We fuse the last three stages using factorized cross-attention, adopting the matrix-multiplication rearrangement in linear attention (Katharopoulos et al., 2020; Shen et al., 2021) to reduce complexity:

$$\mathbf{Z}^{(l)} = \alpha_l \mathbf{Q}^{(l)} \left[\left(\mathbf{K}^{(l)} \right)^{\mathsf{T}} \mathbf{V}^{(l)} \right], \tag{3}$$

where α_l is a scaling factor, which we set to $2/\sqrt{d_l}$ in this work. This formulation first aggregates event context as $\mathbf{K}^{\mathsf{T}}\mathbf{V}$ and then retrieves it with RGB queries. It avoids explicitly constructing an $N_l \times N_l$ attention matrix, reducing complexity from $\mathcal{O}(N_l^2 d_l)$ to $\mathcal{O}(N_l d_l^2)$.

We observe that spike-quantizing the value branch can excessively attenuate the aggregated responses in the $\mathbf{K}^{\mathsf{T}}\mathbf{V}$ operation, as both operands become sparse and discretized. Moreover, event-feature amplitudes retain fine-grained cues related to event density, temporal distributions, motion, and object boundaries. Additional spike quantization may discard these complementary cues and weaken the guidance of event features for RGB representations. Therefore, we spike-quantize the query and key branches while avoiding additional spike quantization of the projected value features, thereby preserving fine-grained event information:

$$\begin{aligned}
\mathbf{Q}^{(l)} &= \mathrm{Spike}_q^{(l)}\left(\mathrm{BN}_q^{(l)}\left[\mathrm{Conv}_q\left(\mathrm{Spike}(\mathbf{X}_r^{(l)})\right)\right]\right), \\
\mathbf{K}^{(l)} &= \mathrm{Spike}_k^{(l)}\left(\mathrm{BN}_k^{(l)}\left[\mathrm{Conv}_k\left(\mathrm{Spike}(\mathbf{X}_e^{(l)})\right)\right]\right), \\
\mathbf{V}^{(l)} &= \mathrm{BN}_v^{(l)}\left[\mathrm{Conv}_v\left(\mathrm{Spike}(\mathbf{X}_e^{(l)})\right)\right].
\end{aligned} \tag{4}$$

Here, l indexes the fusion level; $\mathbf{X}_r^{(l)}$ and $\mathbf{X}_e^{(l)}$ are the corresponding RGB and event features. $\mathrm{Spike}(\cdot)$ denotes the NI-LIF multilevel spiking activation, while Conv and $\mathrm{BN}(\cdot)$ denote pointwise channel projection and batch normalization.

The fused features pass through a spiking activation and output projection, are added residually to the RGB branch, and undergo multiscale feature extraction to produce candidate tokens.

3.4 BOX-GUIDED SPIKING DECODER

Standard query-based decoders mainly obtain visual content through query–memory attention. At low time steps, discrete and sparse spiking queries have limited capacity to carry precise local visual content. Our Box-Guided Spiking Decoder uses current reference boxes to constrain sparse

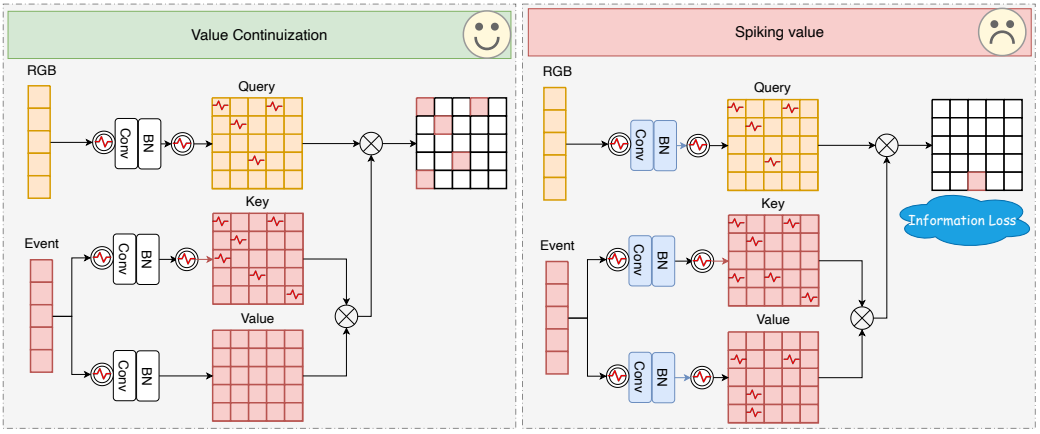

Figure 2: Motivation for value continuization in ASF. Keeping the projected Value continuous preserves dense event information, whereas additional spike quantization introduces information loss during attention aggregation.

sampling across multiple feature levels before each decoder layer. Query-dependent spiking gates aggregate the sampled features, and a lightweight residual correction enriches the current content query while retaining its previous state. The enhanced query then enters the existing spiking cross-attention (SCA), spiking self-attention (SSA), and spiking feed-forward network (SFFN), in that order.

Let $\mathbf{E}^{(l)} \in \mathbb{R}^{B\times N_q\times C}$ and $\mathbf{R}^{(l)} \in [0,1]^{B\times N_q\times 4}$ denote the object-query embeddings and reference boxes entering layer l. Box-guided query enhancement uses two decoupled prediction branches. The offset branch predicts box-relative offsets $\mathbf{O}^{(l)}$ from the current object query, while the gating branch predicts spiking weights $\mathbf{A}^{(l)}$ for the sampling points:

$$\begin{aligned}\mathbf{O}^{(l)} &= \tanh\Big\{\mathrm{BN}\Big[\mathrm{Conv}\Big(\mathrm{Spike}\Big(\mathbf{E}^{(l)}\Big)\Big)\Big]\Big\},\\ \mathbf{A}^{(l)} &= \mathrm{Spike}\Big\{D\,\sigma\Big(\mathrm{BN}\Big[\mathrm{Conv}\Big(\mathrm{Spike}\Big(\mathbf{E}^{(l)}\Big)\Big)\Big]\Big)\Big\}.\end{aligned} \tag{5}$$

Here, $\sigma(\cdot)$ denotes the sigmoid function, and D is the maximum firing level of NI-LIF; thus, $\mathbf{A}^{(l)}$ represents multilevel spiking weights in $\{0, 1/D, \ldots, 1\}$. Let $\mathbf{c}_i^{(l)}$ and $\mathbf{s}_i^{(l)}$ denote the normalized center and size of reference box $\mathbf{R}_i^{(l)}$. The j-th offset $\mathbf{O}_{ij}^{(l)} \in [-1,1]^2$ determines the sampling location:

$$\mathbf{p}_{ij}^{(l)} = \mathrm{clip}\Big(\mathbf{c}_i^{(l)} + \tfrac{1}{2}\mathbf{s}_i^{(l)} \odot \mathbf{O}_{ij}^{(l)}, 0, 1\Big). \tag{6}$$

Features sampled at these locations by bilinear interpolation are weighted by $\mathbf{A}^{(l)}$ and summed across sampling points and feature levels to obtain box-aligned features $\mathbf{G}^{(l)}$. A spiking MLP transforms these features for residual injection into the query:

$$\widetilde{\mathbf{E}}^{(l)} = \mathbf{E}^{(l)} + \boldsymbol{\gamma}^{(l)} \odot \mathrm{SpikeMLP}\Big(\mathbf{G}^{(l)}\Big) \tag{7}$$

Here, $\boldsymbol{\gamma}^{(l)}$ is a channel-wise scaling parameter. Small initial values keep the augmented query mapping close to the identity early in training, reducing interference from inaccurate initial reference boxes.

The enhanced queries then pass through SCA to interact with feature memory, SSA to model inter-query relationships, and SFFN to transform channel features, with a residual connection around each module.

3.5 Spiking Prediction Head and Training Objective

The detection head uses a shared feature embedding followed by separate classification and regression branches. Each decoder layer's object queries pass through two sequential SpikeResNet blocks before entering these branches. The lightweight classification branch outputs class logits, while the deeper spiking regression branch predicts box offsets. Both branches use continuous output readouts to preserve the precision of class confidences and coordinates.

Training follows standard set prediction: bipartite matching assigns predictions to ground-truth objects, with auxiliary supervision at each decoder layer (Carion et al., 2020). The objective combines Varifocal Loss for classification (Zhang et al., 2021), L1 and GIoU losses for box regression (Rezatofighi et al., 2019), and an auxiliary denoising-query loss (Zhang et al., 2022a).

4 Experiments

4.1 Experimental Settings

Datasets. We use the manually annotated DSEC-Det dataset (Gehrig et al., 2021; Liu et al., 2024). It contains spatially aligned RGB images and events. The dataset provides 63,931 annotated timestamps across eight traffic-object categories. PKU-DAVIS-SOD (Li et al., 2023) contains 220 driving sequences and 276,000 annotated timestamps. Its synchronized frames and events are captured by a DAVIS346 camera. The recordings cover diverse conditions, including low illumination and high-speed motion. For PKU-DAVIS-SOD, we follow the official training, validation, and test splits.

Implementation details. Input resolutions are 640×640 for DSEC-Det and 416×416 for PKU-DAVIS-SOD, with $T = 1$, $D = 4$, 300 queries, and six decoder layers. We train for 50 epochs on four V100 GPUs using AdamW (Loshchilov & Hutter, 2017), weight decay 10^{-4}, and initial learning rates of 2×10^{-5}, 4×10^{-5}, and 2×10^{-4} for the RGB backbone, event backbone, and remaining modules, respectively. Detailed configurations of the SpikeRED-S/M/L model scales are provided in Appendix A.1.

Evaluation metrics. We report COCO-style mAP_{50} and mAP (IoU $0.50{:}0.05{:}0.95$) (Lin et al., 2014), and theoretical average inference energy per image (mJ). Dense ANN and spike-driven SNN operation energies are estimated as:

$$E_{\mathrm{ANN}} = H_{\mathrm{out}} W_{\mathrm{out}} C_{\mathrm{in}} C_{\mathrm{out}} k^2 E_{\mathrm{MAC}}, \tag{8}$$

$$E_{\mathrm{SNN}} = T D R_{\mathrm{LIF}} H_{\mathrm{out}} W_{\mathrm{out}} C_{\mathrm{in}} C_{\mathrm{out}} k^2 E_{\mathrm{AC}}. \tag{9}$$

Here, $H_{\mathrm{out}} W_{\mathrm{out}}$ is the output spatial area, $C_{\mathrm{in}}/C_{\mathrm{out}}$ the input/output channel counts, and k the kernel size. T counts time steps and D is the maximum integer firing level. For each MultiSpike module, R_{LIF} averages integer spikes divided by D over all valid samples and tensor elements. We use $E_{\mathrm{MAC}} = 4.6\,\mathrm{pJ}$ and $E_{\mathrm{AC}} = 0.9\,\mathrm{pJ}$ for 45 nm, 32-bit floating-point hardware (Horowitz, 2014; Luo et al., 2024).

4.2 Comparison with State-of-the-Art Methods

Table 1 compares SpikeRED with RGB-only, event-only, and RGB+Event detectors using ANN, SNN, and hybrid architectures.

SpikeRED-L achieves the highest mAP in Table 1: 30.7 on DSEC-Det and 32.2 on PKU-DAVIS-SOD. It exceeds SFNet by 0.3% on both datasets while reducing theoretical energy by 85.2% and 86.6%, respectively. SpikeRED-S outperforms EOLO by 3.5% and 3.2% in mAP with 52.9% and 56.5% lower energy.

With RGB-only inputs, SpikeRED-S* achieves accuracy comparable to SpikeYOLO with fewer parameters (17.4M vs. 23.1M) and 28.9% and 37.6% lower energy on the two datasets. SpikeRED-L* improves mAP by 2.2% and 1.6% over SpikeYOLO.

4.3 Ablation Studies

Using SpikeRED-L as the reference model, we conduct two ablation studies on DSEC-Det and PKU-DAVIS-SOD to assess Box-Guided Query Refinement and Asymmetric Spiking Fusion.

Table 1: Comparison of RGB, Event, and RGB+Event detectors on DSEC-Det and PKU-DAVIS-SOD. Bold values indicate the highest detection accuracy across all methods. * denotes an RGB-only variant with the fusion modules removed.

Input	Type	Method	$T \times D$	DSEC-Det			PKU-DAVIS-SOD			Params
				$\mathbf{mAP_{50}} \uparrow$	$\mathbf{mAP} \uparrow$	Energy $(\mathbf{mJ}) \downarrow$	$\mathbf{mAP_{50}} \uparrow$	$\mathbf{mAP} \uparrow$	Energy $(\mathbf{mJ}) \downarrow$	
RGB	ANN	MambaYOLO (Wang et al., 2025b)	1	44.1	28.3	155.9	57.1	29.8	65.9	21.8M
		YOLOv11 (Khanam & Hussain, 2024)	1	46.7	30.3	156.7	58.0	30.9	66.1	20.1M
		YOLOX (Ge et al., 2021)	1	43.5	26.5	169.2	57.5	30.6	71.5	25.3M
	SNN	SpikeYOLO (Luo et al., 2024)	1×4	42.9	26.6	38.1	56.7	29.5	16.5	23.1M
		SpikeRED-S*	1×4	42.9	26.7	27.1	56.5	29.5	10.3	17.4M
		SpikeRED-M*	1×4	46.5	28.2	49.8	58.4	30.8	18.7	24.7M
		SpikeRED-L*	1×4	47.4	28.8	74.9	58.6	31.1	28.9	37.0M
Event	ANN	YOLOX-Voxel (Ge et al., 2021)	1	23.9	11.7	169.2	–	–	–	25.3M
		SAST (Peng et al., 2024)	1	24.3	12.1	131.9	48.7	24.5	51.3	18.5M
		S5-ViT (Zubic et al., 2024)	1	22.3	11.4	133.8	46.6	23.3	50.9	18.2M
		SMamba (Yang et al., 2025)	1	29.0	14.8	98.0	53.0	27.1	41.4	17.0M
RGB + Event	ANN	FPN-Fusion (Tomy et al., 2022)	1	34.4	18.6	545.4	36.6	19.5	231.0	65.6M
		SODFormer (Li et al., 2023)	1	–	–	–	50.4	20.7	303.1	82.0M
		DAGr (Gehrig & Scaramuzza, 2024)	1	–	–	–	49.2	–	–	34.6M
		CAFR (Cao et al., 2024a)	1	23.4	12.5	1732.3	52.0	26.0	624.8	82.4M
		SFNet (Liu et al., 2024)	1	**51.4**	30.4	907.0	59.6	31.9	373.3	57.5M
		ACGR (Li et al., 2025)	1	–	–	–	51.9	–	–	41.4M
	ANN+SNN	EOLO (Cao et al., 2024b)	1	40.1	23.8	99.1	52.8	27.0	42.5	46.3M
	SNN	**SpikeRED-S**	1×4	45.7	27.3	46.7	57.1	30.2	18.5	22.5M
		SpikeRED-M	1×4	50.2	30.2	82.7	59.3	31.5	31.7	34.9M
		SpikeRED-L	1×4	50.8	**30.7**	133.8	**62.3**	**32.2**	49.9	56.5M

4.3.1 Effect of Box-Guided Query Refinement

We evaluate the following three decoder configurations while keeping all other settings unchanged:

- **Box-Guided Query Refinement:** Our complete decoder aggregates multiscale regional features using current reference boxes and refines queries residually before each layer of the same spiking-attention decoder.
- **Spiking Deformable Attention:** Spiking self-attention and spike-based deformable attention, with the sampling operator based on Deformable DETR (Zhu et al., 2020).
- **Plain Spiking Attention:** Plain spiking self- and cross-attention with the same implementation and execution order as our method, but without box-guided regional sampling or residual query enhancement before decoder layers.

Table 2: Effect of Box-Guided Query Refinement with RGB+Event input. All variants share the same backbone, fusion module and prediction head. Energy is the theoretical inference energy per image in mJ.

Dataset	Decoder	$\mathrm{mAP}_{50} \uparrow$	$\mathrm{mAP} \uparrow$	Energy (mJ)$\downarrow$
DSEC-Det	Box-Guided Query Refinement (Ours)	50.8	30.7	133.8
	Spiking Deformable Attention	39.9	22.5	129.1
	Plain Spiking Attention	40.8	23.9	133.5
PKU-DAVIS-SOD	Box-Guided Query Refinement (Ours)	62.3	32.2	49.9
	Spiking Deformable Attention	50.6	24.2	48.8
	Plain Spiking Attention	52.1	25.3	49.7

In Table 2, box-guided refinement improves mAP by 6.8% and 6.9% over Plain Spiking Attention on DSEC-Det and PKU-DAVIS-SOD, with only 0.3 mJ (0.2%) and 0.2 mJ (0.4%) more theoretical inference energy, respectively. Gains over Spiking Deformable Attention are 8.2% and 8.0%, with 3.6% and 2.3% more theoretical inference energy. Figure 3 shows that box guidance concentrates sampling around objects. These comparisons support the effectiveness of box-aligned residual query enhancement in our spiking decoder.

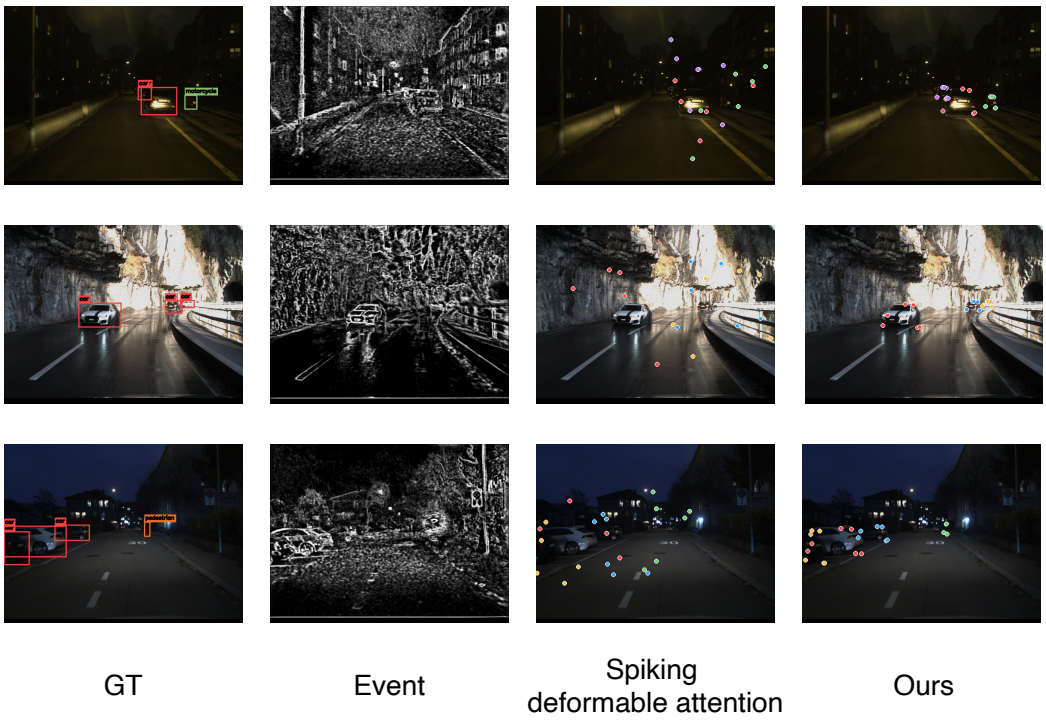

Figure 3: Qualitative comparison of spatial sampling patterns in the decoder ablation. From left to right: RGB images with ground-truth boxes (GT), event representations, Spiking Deformable Attention, and our box-guided decoder. Colored points indicate sampling locations, with different colors corresponding to different objects. In these examples, our method concentrates sampling more closely around object regions.

4.3.2 Effect of Asymmetric Spiking Fusion

We fix the Box-Guided Spiking Decoder and training settings. Baseline ASF uses spiking RGB queries, spiking event keys, and continuous event values. ASF w/ Spiking Value adds NI-LIF activation to the values in Eq. 4. Reversed ASF uses event queries and RGB keys and values; Bidirectional ASF sums both directions' attention outputs before output spiking and projection. Both retain spiking queries and keys and continuous values. EOLO Fusion replaces ASF with the bidirectional fusion module from EOLO (Cao et al., 2024b). These fusion variants preserve all other detection modules; RGB Only and Event Only retain the corresponding backbone without cross-modal fusion.

Baseline ASF achieves the highest mAP among all seven configurations in Table 3. Quantizing the value branch into spikes reduces mAP by 1.6% on DSEC-Det and 1.2% on PKU-DAVIS-SOD, while saving only 0.2 mJ on DSEC-Det and yielding no measurable energy reduction on PKU-DAVIS-SOD at the reported precision. Retaining continuous values therefore improves accuracy with negligible additional whole-network theoretical energy.

Fusion direction also matters. Reversed ASF reduces mAP by 7.0% and 4.6% on the two datasets at similar energy, supporting the use of RGB queries to retrieve event context. Bidirectional ASF lowers mAP by 2.4% and 2.1% while increasing energy by 1.1% and 1.2%. Under the evaluated settings, adding the reverse branch thus provides no accuracy benefit and incurs additional computation.

ASF outperforms EOLO Fusion by 1.8% and 0.7% in mAP with 22.7% and 26.5% lower energy. It also improves mAP over RGB Only by 1.9% and 1.1%, and over Event Only by 17.9% and 7.1%, supporting the benefit of cross-modal fusion. Figure 4 shows vehicles detected by ASF but missed by EOLO Fusion in shadows, low light, and bright tunnel-exit scenes.

Table 3: Effect of Asymmetric Spiking Fusion: value quantization, fusion direction, EOLO fusion, and single-modality inputs. Baseline is the complete ASF model.

Dataset	Method	mAP_{50} ↑	mAP ↑	Energy (mJ)↓
DSEC-Det	Baseline (ASF, Ours)	50.8	30.7	133.8
	ASF w/ Spiking Value	49.9	29.1	133.6
	Bidirectional ASF	49.0	28.3	135.3
	Reversed ASF	40.9	23.7	133.5
	EOLO Fusion	47.6	28.9	173.1
	RGB Only	47.4	28.8	74.9
	Event Only	24.8	12.8	73.0
PKU-DAVIS-SOD	Baseline (ASF, Ours)	62.3	32.2	49.9
	ASF w/ Spiking Value	58.7	31.0	49.9
	Bidirectional ASF	57.3	30.1	50.5
	Reversed ASF	53.1	27.6	49.6
	EOLO Fusion	59.3	31.5	67.9
	RGB Only	58.6	31.1	28.9
	Event Only	48.9	25.1	27.8

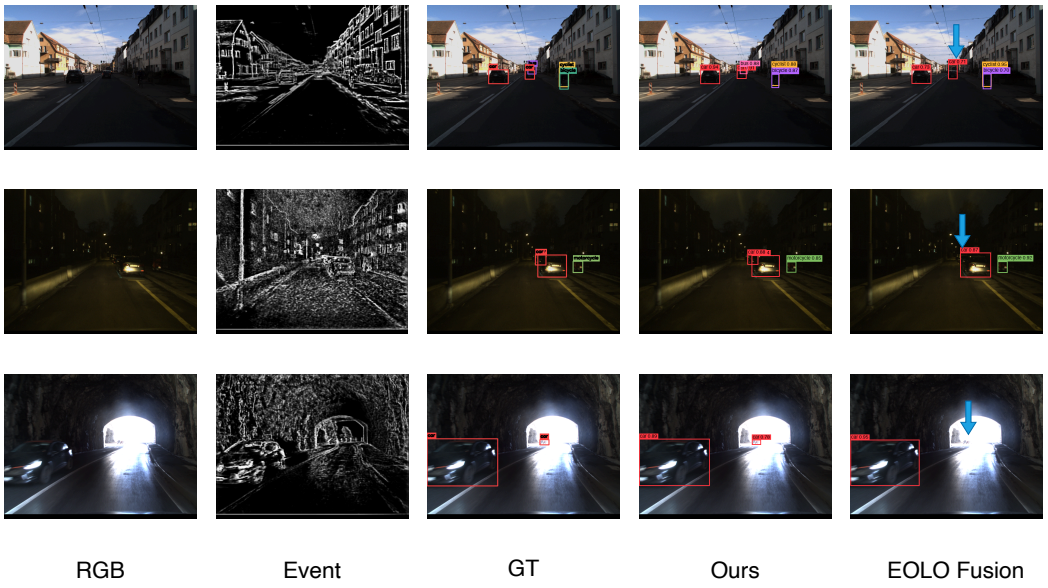

Figure 4: Qualitative detection results for the fusion ablation. From left to right: RGB inputs, event representations, ground-truth boxes (GT), our ASF model, and the EOLO Fusion variant. Blue arrows highlight examples of vehicles missed by EOLO Fusion but detected by our method under challenging illumination.

5 Conclusion

We propose SpikeRED, a fully spike-driven, query-based detector for aligned RGB images and event streams. It consists of dual-branch spiking backbones, Asymmetric Spiking Fusion (ASF), a Box-Guided Spiking Decoder, and a spiking prediction head. ASF transfers event information to the RGB branch through one-way fusion. The decoder uses reference boxes to sample local multiscale features and enhance object queries. SpikeRED-L achieves 30.7% and 32.2% mAP on DSEC-Det and PKU-DAVIS-SOD. It exceeds SFNet by 0.3% mAP on both datasets and reduces theoretical inference energy by over 85%. The proposed decoder also improves mAP by 6.8% and 6.9% over Plain Spiking Attention. These results show that SpikeRED provides accurate and energy-efficient RGB-event detection.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation and to improve the readability and wording of a small number of individual sentences. We did not use generative AI tools to develop the research methodology, design experiments, formulate hypotheses, interpret experimental results, or generate scientific claims. All AI-assisted text was carefully reviewed and revised by the authors. We take full responsibility for the final content of this work, including any text produced with the aid of generative AI.

REFERENCES

Hu Cao, Zehua Zhang, Yan Xia, Xinyi Li, Jiahao Xia, Guang Chen, and Alois Knoll. Embracing events and frames with hierarchical feature refinement network for object detection. In *European Conference on Computer Vision*, pp. 161–177. Springer, 2024a.

Jiahang Cao, Xu Zheng, Yuanhuiyi Lyu, Jiaxu Wang, Renjing Xu, and Lin Wang. Chasing day and night: Towards robust and efficient all-day object detection guided by an event camera. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9026–9032. IEEE, 2024b.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Zhe Chen, Jing Zhang, and Dacheng Tao. Recurrent glimpse-based decoder for detection with transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5250–5259. IEEE, 2022.

Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

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

Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954, 2021.

Zhonghui Han, Xinhao Luo, Jianwu Fang, Yuning Wang, Guoqi Li, and Jianru Xue. Low-latency object detection in driving scenes: A hybrid spiking neural network for motion-compensated fusion. *IEEE Transactions on Intelligent Transportation Systems*, 2026.

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

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

Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

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

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

Dianze Li, Yonghong Tian, and Jianing Li. Sodformer: Streaming object detection with transformer using events and frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (11):14020–14037, 2023.

Dianze Li, Jianing Li, Xu Liu, Xiaopeng Fan, and Yonghong Tian. Asynchronous collaborative graph representation for frames and events. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1655–1666. IEEE, 2025.

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

Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. arxiv 2022. *arXiv preprint arXiv:2201.12329*, 2201, 2022.

Zhanwen Liu, Nan Yang, Yang Wang, Yuke Li, Xiangmo Zhao, and Fei-Yue Wang. Enhancing traffic object detection in variable illumination with rgb-event fusion. *IEEE Transactions on Intelligent Transportation Systems*, 25(12):20335–20350, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

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

Chethan M Parameshwara, Simin Li, Cornelia Fermüller, Nitin J Sanket, Matthew S Evanusa, and Yiannis Aloimonos. Spikems: Deep spiking neural network for motion segmentation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3414–3420. IEEE, 2021.

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

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. *arXiv preprint arXiv:1902.09630*, 2019.

Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3531–3539, 2021.

Qiaoyi Su, Yuhong Chou, Yifan Hu, Jianing Li, Shijie Mei, Ziyang Zhang, and Guoqi Li. Deep directly-trained spiking neural networks for object detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6532–6542. IEEE, 2023.

Abhishek Tomy, Anshul Paigwar, Khushdeep S Mann, Alessandro Renzaglia, and Christian Laugier. Fusing event-based and rgb camera for robust object detection in adverse conditions. In *2022 International conference on robotics and automation (ICRA)*, pp. 933–939. IEEE, 2022.

Hechong Wang, Shilong Jing, Chengshan Han, Yuchen Zhao, Hailong Liu, and Hengyi Lv. Event-guided object detection under spiking transmission. *Knowledge-Based Systems*, pp. 114776, 2025a.

Zeyu Wang, Chen Li, Huiying Xu, Xinzhong Zhu, and Hongbo Li. Mamba yolo: A simple baseline for object detection with state space model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 8205–8213, 2025b.

Nan Yang, Yang Wang, Zhanwen Liu, Meng Li, Yisheng An, and Xiangmo Zhao. Smamba: Sparse mamba for event-based object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 9229–9237, 2025.

Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *Advances in neural information processing systems*, 36:64043–64058, 2023.

Man Yao, Xuerui Qiu, Tianxiang Hu, Jiakui Hu, Yuhong Chou, Keyu Tian, Jianxing Liao, Luziwei Leng, Bo Xu, and Guoqi Li. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4): 2973–2990, 2025.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022a.

Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8514–8523, 2021.

Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking transformers for event-based single object tracking. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8791–8800. IEEE, 2022b.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16965–16974. IEEE, 2024.

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

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

Nikola Zubic, Mathias Gehrig, and Davide Scaramuzza. State space models for event cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5819–5828, 2024.

A SUPPLEMENTARY MATERIAL

A.1 SPIKERED MODEL-SCALE CONFIGURATIONS

SpikeRED-S, SpikeRED-M, and SpikeRED-L mainly differ in the width of their SDT v3 backbones. The mapping to the corresponding backbone configurations is:

SpikeRED-S	`Efficient_Spiking_Transformer_s`
SpikeRED-M	`Efficient_Spiking_Transformer_m`
SpikeRED-L	`Efficient_Spiking_Transformer_l`

For RGB+Event detection, each model contains two topology-identical but parameter-independent backbones, one for RGB and one for events. The backbone stage widths and the corresponding feature-interface dimensions are summarized in Table 4.

Table 4: Configurations of the SpikeRED-S/M/L model scales. C_1–C_4 denote the four SDT v3 backbone output widths. The encoder uses the last three backbone outputs, and the decoder input widths are the output widths of the subsequent multiscale encoder. Parameter counts are for the complete detection models reported in Table 1.

Model	Backbone widths $[C_1, C_2, C_3, C_4]$	Encoder input widths	Decoder input widths	Total params
SpikeRED-S	$[32, 64, 128, 192]$	$[64, 128, 192]$	$[64, 128, 256]$	22.5M
SpikeRED-M	$[48, 96, 192, 240]$	$[96, 192, 240]$	$[96, 192, 384]$	34.9M
SpikeRED-L	$[64, 128, 256, 360]$	$[128, 256, 360]$	$[128, 256, 512]$	56.5M

All three scales use the same backbone block allocation: two convolutional blocks in Stage 1, two convolutional blocks in Stage 2, six spike-driven self-attention blocks in Stage 3, and two in Stage 4. They also share eight attention heads, an MLP expansion ratio of 4, an attention spatial-reduction ratio of 1, and no QKV bias. ASF is applied to the last three backbone stages (stage indices $[1, 2, 3]$) using RGB queries and event keys/values. The multiscale encoder and decoder architecture otherwise remain fixed across scales: the decoder hidden dimension is 256, with three feature levels, 300 object queries, six decoder layers, and 100 denoising queries. All variants use $T = 1$ and $D = 4$. Thus, the S/M/L comparison primarily measures backbone-width scaling while retaining the same fusion and detection design.

The RGB-only variants remove the event backbone and all cross-modal fusion modules while retaining the corresponding S/M/L RGB backbone and the same encoder–decoder design. Their complete parameter counts are 17.4M, 24.7M, and 37.0M, respectively.

A.2 THEORETICAL INFERENCE-ENERGY CALCULATION

We follow the theoretical energy-evaluation protocol commonly used in SNN studies (Horowitz, 2014; Yao et al., 2023; Luo et al., 2024). We first measure the firing rate of every spiking layer. For NI-LIF, the normalized firing rate is

$$R_l = \frac{1}{n_l} \sum_{i=1}^{n_l} \frac{a_i}{D}, \tag{10}$$

where $a_i \in \{0, \ldots, D\}$ is the integer spike and n_l is the number of elements in layer l. Since an integer spike can be expanded into D binary spikes, DR_l gives the average number of spike-driven accumulations per element. We average R_l over all valid test images and calculate it separately for each layer.

Let $\mathcal{F}_l$ denote the number of operations in layer l. A dense ANN layer consumes $E_{\mathrm{MAC}}\mathcal{F}_l$, whereas a spike-driven layer consumes

$$E_l = E_{\mathrm{AC}}\, TDR_l\, \mathcal{F}_l. \tag{11}$$

For a convolution, $\mathcal{F}_{\mathrm{Conv}} = H_{\mathrm{out}}W_{\mathrm{out}}C_{\mathrm{in}}C_{\mathrm{out}}k^2/g$, where g is the number of groups. For a Linear layer with N tokens, $\mathcal{F}_{\mathrm{Linear}} = NC_{\mathrm{in}}C_{\mathrm{out}}$. For an attention multiplication between $M \times K$ and $K \times N$ matrices, $\mathcal{F}_{\mathrm{Attn}} = hMNK$, where h is the number of heads. Each attention multiplication is counted separately and uses the firing rate of its spike operand. The first convolution directly

processes dense RGB or event inputs and is therefore evaluated with MACs. All subsequent spike-driven convolutions, projections, and MLP layers are evaluated with ACs.

Table 5: Theoretical energy calculation used for dense ANN and SpikeRED operations. $\mathcal{F}$ denotes the operation count, and R denotes the firing rate of the spike tensor that drives the operation.

Module	Operation	Dense ANN	SpikeRED
Conv	First convolution	$E_{\mathrm{MAC}}\mathcal{F}_{\mathrm{Conv}}$	$E_{\mathrm{MAC}}\mathcal{F}_{\mathrm{Conv}}$
Conv/Linear	Other layers	$E_{\mathrm{MAC}}\mathcal{F}_l$	$E_{\mathrm{AC}}TDR_l\mathcal{F}_l$
Attention	Q/K/V and output projections	$E_{\mathrm{MAC}}\sum_j \mathcal{F}_j$	$E_{\mathrm{AC}}TD\sum_j R_j\mathcal{F}_j$
Attention	Each matrix multiplication	$E_{\mathrm{MAC}}hMNK$	$E_{\mathrm{AC}}TDR_ShMNK$
MLP	Layer 1 and Layer 2	$E_{\mathrm{MAC}}\sum_j \mathcal{F}_j$	$E_{\mathrm{AC}}TD\sum_j R_j\mathcal{F}_j$

For standard attention, we count QK^{T} and the attention-score–value product. For factorized linear attention, we count KV and $Q(KV)$. SpikeRED uses softmax-free attention, so no softmax energy is added. We use $T = 1$, $D = 4$, $E_{\mathrm{MAC}} = 4.6\,\mathrm{pJ}$, and $E_{\mathrm{AC}} = 0.9\,\mathrm{pJ}$ under the 45 nm hardware assumption. The total inference energy is obtained by summing all layer-wise costs and averaging them per test image.

A.3 Complete Energy Breakdown on PKU-DAVIS-SOD

Table 6 presents the complete operator-wise theoretical inference energy of SpikeRED-L on the PKU-DAVIS-SOD test set, following the protocol in Section A.2. The table contains 282 entries and reports a total of $49871.638\,\mu\mathrm{J} \approx 49.8716\,\mathrm{mJ}$. This is consistent with the 49.9 mJ reported for SpikeRED-L on PKU-DAVIS-SOD in the main results and fusion ablation tables.

Table 6: Complete operator-wise theoretical inference-energy breakdown of SpikeRED-L on PKU-DAVIS-SOD.

Phase	Stage/Layer	Module	Operation	R_{LIF}	Energy (μJ)
RGB backbone	Stage 1	DownSampling	Conv	1.000000	936.164
RGB backbone	Stage 1	DownSampling	Conv	0.294566	211.409
RGB backbone	Stage 1	SepSpikeConv	PWConv1	0.367366	117.181
RGB backbone	Stage 1	SepSpikeConv	DWConv	0.334872	163.562
RGB backbone	Stage 1	SepSpikeConv	PWConv2	0.344007	109.730
RGB backbone	Stage 1	Channel Conv	Conv1	0.426034	2446.112
RGB backbone	Stage 1	Channel Conv	Conv2	0.096776	555.647
RGB backbone	Stage 1	SepSpikeConv	PWConv1	0.182596	58.244
RGB backbone	Stage 1	SepSpikeConv	DWConv	0.232527	56.787
RGB backbone	Stage 1	SepSpikeConv	PWConv2	0.170515	54.390
RGB backbone	Stage 1	Channel Conv	Conv1	0.265827	1526.267
RGB backbone	Stage 1	Channel Conv	Conv2	0.042321	242.987
RGB backbone	Stage 2	DownSampling	Conv	0.217224	155.901
RGB backbone	Stage 2	SepSpikeConv	PWConv1	0.191394	61.050
RGB backbone	Stage 2	SepSpikeConv	DWConv	0.229855	28.067
RGB backbone	Stage 2	SepSpikeConv	PWConv2	0.134926	43.038
RGB backbone	Stage 2	Channel Conv	Conv1	0.197122	1131.791
RGB backbone	Stage 2	Channel Conv	Conv2	0.022962	131.840
RGB backbone	Stage 2	SepSpikeConv	PWConv1	0.284522	90.756
RGB backbone	Stage 2	SepSpikeConv	DWConv	0.144712	17.671
RGB backbone	Stage 2	SepSpikeConv	PWConv2	0.122768	39.160
RGB backbone	Stage 2	Channel Conv	Conv1	0.216759	1244.540
RGB backbone	Stage 2	Channel Conv	Conv2	0.018132	104.105
RGB backbone	Stage 3	DownSampling	Conv	0.237804	170.671
RGB backbone	Stage 3	SepSpikeConv	PWConv1	0.291231	557.375
RGB backbone	Stage 3	SepSpikeConv	DWConv	0.169360	11.395
RGB backbone	Stage 3	SepSpikeConv	PWConv2	0.080907	154.845
RGB backbone	Stage 3	Linear Self-Attention	Q/K/V projections	0.283365	1626.965
RGB backbone	Stage 3	Linear Self-Attention	Q×K^{T} MatMul	0.201407	508.933
RGB backbone	Stage 3	Linear Self-Attention	Score×V MatMul	0.114642	1158.757
RGB backbone	Stage 3	Linear Self-Attention	Output projection	0.574336	2198.398
RGB backbone	Stage 3	Channel MLP	Linear1	0.308254	1179.909
RGB backbone	Stage 3	Channel MLP	Linear2	0.013103	50.155
RGB backbone	Stage 4	DownSampling	Conv	0.218114	440.268
RGB backbone	Stage 4	SepSpikeConv	PWConv1	0.192588	242.965
RGB backbone	Stage 4	SepSpikeConv	DWConv	0.061102	1.927

Continued on next page

Table 6 (continued)

Phase	Stage/Layer	Module	Operation	R_{LIF}	Energy (μJ)
RGB backbone	Stage 4	SepSpikeConv	PWConv2	0.057336	72.334
RGB backbone	Stage 4	Linear Self-Attention	Q/K/V projections	0.197022	745.675
RGB backbone	Stage 4	Linear Self-Attention	Q$\times$K^{T} MatMul	0.192099	227.538
RGB backbone	Stage 4	Linear Self-Attention	Score$\times$V MatMul	0.025978	123.080
RGB backbone	Stage 4	Linear Self-Attention	Output projection	0.223014	562.699
RGB backbone	Stage 4	Channel MLP	Linear1	0.229625	579.380
RGB backbone	Stage 4	Channel MLP	Linear2	0.003744	9.447
Event backbone	Stage 1	DownSampling	Conv	1.000000	936.164
Event backbone	Stage 1	DownSampling	Conv	0.293227	210.449
Event backbone	Stage 1	SepSpikeConv	PWConv1	0.368900	117.670
Event backbone	Stage 1	SepSpikeConv	DWConv	0.342090	167.088
Event backbone	Stage 1	SepSpikeConv	PWConv2	0.371463	118.488
Event backbone	Stage 1	Channel Conv	Conv1	0.406873	2336.096
Event backbone	Stage 1	Channel Conv	Conv2	0.144421	829.207
Event backbone	Stage 1	SepSpikeConv	PWConv1	0.223024	71.139
Event backbone	Stage 1	SepSpikeConv	DWConv	0.280325	68.460
Event backbone	Stage 1	SepSpikeConv	PWConv2	0.180372	57.534
Event backbone	Stage 1	Channel Conv	Conv1	0.293960	1687.793
Event backbone	Stage 1	Channel Conv	Conv2	0.043176	247.898
Event backbone	Stage 2	DownSampling	Conv	0.225116	161.565
Event backbone	Stage 2	SepSpikeConv	PWConv1	0.192764	61.487
Event backbone	Stage 2	SepSpikeConv	DWConv	0.257010	31.383
Event backbone	Stage 2	SepSpikeConv	PWConv2	0.123485	39.389
Event backbone	Stage 2	Channel Conv	Conv1	0.196705	1129.396
Event backbone	Stage 2	Channel Conv	Conv2	0.027111	155.661
Event backbone	Stage 2	SepSpikeConv	PWConv1	0.266086	84.875
Event backbone	Stage 2	SepSpikeConv	DWConv	0.140874	17.202
Event backbone	Stage 2	SepSpikeConv	PWConv2	0.119815	38.218
Event backbone	Stage 2	Channel Conv	Conv1	0.203996	1171.259
Event backbone	Stage 2	Channel Conv	Conv2	0.016220	93.130
Event backbone	Stage 3	DownSampling	Conv	0.216320	155.252
Event backbone	Stage 3	SepSpikeConv	PWConv1	0.282634	540.923
Event backbone	Stage 3	SepSpikeConv	DWConv	0.173517	11.675
Event backbone	Stage 3	SepSpikeConv	PWConv2	0.078380	150.008
Event backbone	Stage 3	Linear Self-Attention	Q/K/V projections	0.274708	1577.261
Event backbone	Stage 3	Linear Self-Attention	Q$\times$K^{T} MatMul	0.210201	531.157
Event backbone	Stage 3	Linear Self-Attention	Score$\times$V MatMul	0.125374	1267.223
Event backbone	Stage 3	Linear Self-Attention	Output projection	0.547501	2095.682
Event backbone	Stage 3	Channel MLP	Linear1	0.299911	1147.976
Event backbone	Stage 3	Channel MLP	Linear2	0.014595	55.867
Event backbone	Stage 4	DownSampling	Conv	0.241660	487.797
Event backbone	Stage 4	SepSpikeConv	PWConv1	0.197185	248.765
Event backbone	Stage 4	SepSpikeConv	DWConv	0.059793	1.886
Event backbone	Stage 4	SepSpikeConv	PWConv2	0.059162	74.638
Event backbone	Stage 4	Linear Self-Attention	Q/K/V projections	0.200887	760.303
Event backbone	Stage 4	Linear Self-Attention	Q$\times$K^{T} MatMul	0.145228	172.020
Event backbone	Stage 4	Linear Self-Attention	Score$\times$V MatMul	0.027026	128.047
Event backbone	Stage 4	Linear Self-Attention	Output projection	0.134533	339.449
Event backbone	Stage 4	Channel MLP	Linear1	0.249302	629.028
Event backbone	Stage 4	Channel MLP	Linear2	0.002933	7.401
Fusion	Stage 2	Linear Cross-Attention	Q projection	0.237804	37.927
Fusion	Stage 2	Linear Cross-Attention	K projection	0.216320	34.501
Fusion	Stage 2	Linear Cross-Attention	V projection	0.216320	34.501
Fusion	Stage 2	Linear Cross-Attention	K$\times$V MatMul	0.052560	1.048
Fusion	Stage 2	Linear Cross-Attention	Q$\times$(K$\times$V) MatMul	0.090117	1.797
Fusion	Stage 2	Linear Cross-Attention	Output projection	0.456590	72.821
Fusion	Stage 3	Linear Cross-Attention	Q projection	0.218114	34.787
Fusion	Stage 3	Linear Cross-Attention	K projection	0.241660	38.542
Fusion	Stage 3	Linear Cross-Attention	V projection	0.241660	38.542
Fusion	Stage 3	Linear Cross-Attention	K$\times$V MatMul	0.079692	1.589
Fusion	Stage 3	Linear Cross-Attention	Q$\times$(K$\times$V) MatMul	0.089447	1.783
Fusion	Stage 3	Linear Cross-Attention	Output projection	0.471228	75.155
Fusion	Stage 4	Linear Cross-Attention	Q projection	0.201694	63.613
Fusion	Stage 4	Linear Cross-Attention	K projection	0.235871	74.392
Fusion	Stage 4	Linear Cross-Attention	V projection	0.235871	74.392
Fusion	Stage 4	Linear Cross-Attention	K$\times$V MatMul	0.080608	3.178
Fusion	Stage 4	Linear Cross-Attention	Q$\times$(K$\times$V) MatMul	0.087733	3.459
Fusion	Stage 4	Linear Cross-Attention	Output projection	0.462905	145.998
FPN	Top-down FPN	MS_StandardConv	Conv	0.158529	35.555
FPN	Top-down FPN	SepSpikeConv	PWConv1	0.061547	19.632
FPN	Top-down FPN	SepSpikeConv	DWConv	0.098698	6.026
FPN	Top-down FPN	SepSpikeConv	PWConv2	0.092445	29.488
FPN	Top-down FPN	Channel Conv	Conv1	0.102779	442.587
FPN	Top-down FPN	Channel Conv	Conv2	0.089236	384.266
FPN	Top-down FPN	MS_StandardConv	Conv	0.179216	28.583

Continued on next page

Table 6 (continued)

Phase	Stage/Layer	Module	Operation	R_{LIF}	Energy (μJ)
FPN	Top-down FPN	SepSpikeConv	PWConv1	0.084847	27.064
FPN	Top-down FPN	SepSpikeConv	DWConv	0.117792	14.383
FPN	Top-down FPN	SepSpikeConv	PWConv2	0.107354	34.243
FPN	Top-down FPN	Channel Conv	Conv1	0.079875	458.610
FPN	Top-down FPN	Channel Conv	Conv2	0.048037	275.806
FPN	Top-down FPN	MS_StandardConv	Conv	0.168810	53.847
FPN	Top-down FPN	SepSpikeConv	PWConv1	0.084316	26.895
FPN	Top-down FPN	SepSpikeConv	DWConv	0.107848	13.169
FPN	Top-down FPN	SepSpikeConv	PWConv2	0.117914	37.612
FPN	Top-down FPN	Channel Conv	Conv1	0.111638	640.978
FPN	Top-down FPN	Channel Conv	Conv2	0.058807	337.643
FPN	Bottom-up PAN	MS_StandardConv	Conv	0.156659	56.217
FPN	Bottom-up PAN	SepSpikeConv	PWConv1	0.084013	26.798
FPN	Bottom-up PAN	SepSpikeConv	DWConv	0.109543	6.688
FPN	Bottom-up PAN	SepSpikeConv	PWConv2	0.109255	34.850
FPN	Bottom-up PAN	Channel Conv	Conv1	0.128301	552.489
FPN	Bottom-up PAN	Channel Conv	Conv2	0.080086	344.864
FPN	Bottom-up PAN	SepSpikeConv	PWConv1	0.124528	158.886
FPN	Bottom-up PAN	SepSpikeConv	DWConv	0.088723	10.834
FPN	Bottom-up PAN	SepSpikeConv	PWConv2	0.080601	102.839
FPN	Bottom-up PAN	Channel Conv	Conv1	0.152023	872.854
FPN	Bottom-up PAN	Channel Conv	Conv2	0.061382	352.430
Decoder	Input Projection	Input projection	Conv	0.121987	38.911
Decoder	Input Projection	Input projection	Conv	0.159944	25.509
Decoder	Input Projection	Input projection	Conv	0.144853	46.205
Decoder	Encoder Query Init	Query projection	Conv1d	0.050888	48.696
Decoder	Encoder Query Init	Detection embedding	Conv1	0.077536	74.196
Decoder	Encoder Query Init	Detection embedding	Conv2	0.124028	118.686
Decoder	Encoder Query Init	Classification	Hidden projection	0.142692	136.547
Decoder	Encoder Query Init	Classification	Class output	0.159842	1.792
Decoder	Encoder Query Init	Box regression	Conv1	0.142692	136.547
Decoder	Encoder Query Init	Box regression	Conv2	0.164260	157.185
Decoder	Encoder Query Init	Box regression	Box output	0.203959	3.050
Decoder	Decoder Layer 1	Self-Attention	Q/K/V projections	0.179406	38.094
Decoder	Decoder Layer 1	Self-Attention	$Q \times K^{T}$ MatMul	0.089284	7.406
Decoder	Decoder Layer 1	Self-Attention	Score $\times$ V MatMul	0.080616	6.687
Decoder	Decoder Layer 1	Self-Attention	Output projection	0.443127	31.364
Decoder	Decoder Layer 1	Linear Cross-Attention	Q projection	0.174441	12.347
Decoder	Decoder Layer 1	Linear Cross-Attention	K projection	0.051968	49.730
Decoder	Decoder Layer 1	Linear Cross-Attention	V projection	0.051968	49.730
Decoder	Decoder Layer 1	Linear Cross-Attention	K $\times$ V MatMul	0.079888	9.556
Decoder	Decoder Layer 1	Linear Cross-Attention	Q $\times$ (K $\times$ V) MatMul	0.101168	0.895
Decoder	Decoder Layer 1	Linear Cross-Attention	Output projection	0.523031	37.020
Decoder	Decoder Layer 1	Channel MLP	Linear1	0.185889	52.628
Decoder	Decoder Layer 1	Channel MLP	Linear2	0.100860	28.555
Decoder	Decoder Layer 1	Weight MLP	Linear1	0.431973	61.149
Decoder	Decoder Layer 1	Weight MLP	Linear2	0.108579	15.370
Decoder	Decoder Layer 1	Sampling offset	Offset projection	0.167300	0.833
Decoder	Decoder Layer 1	Sampling weight	Weight projection	0.167300	1.249
Decoder	Decoder Layer 1	Detection embedding	Conv1	0.171979	12.172
Decoder	Decoder Layer 1	Detection embedding	Conv2	0.165478	11.712
Decoder	Decoder Layer 1	Classification	Hidden projection	0.158577	11.224
Decoder	Decoder Layer 1	Classification	Class output	0.189313	0.157
Decoder	Decoder Layer 1	Box regression	Conv1	0.158577	11.224
Decoder	Decoder Layer 1	Box regression	Conv2	0.130658	9.248
Decoder	Decoder Layer 1	Box regression	Box output	0.139428	0.154
Decoder	Decoder Layer 2	Self-Attention	Q/K/V projections	0.174227	36.995
Decoder	Decoder Layer 2	Self-Attention	$Q \times K^{T}$ MatMul	0.078744	6.531
Decoder	Decoder Layer 2	Self-Attention	Score $\times$ V MatMul	0.066780	5.539
Decoder	Decoder Layer 2	Self-Attention	Output projection	0.238150	16.856
Decoder	Decoder Layer 2	Linear Cross-Attention	Q projection	0.175690	12.435
Decoder	Decoder Layer 2	Linear Cross-Attention	K projection	0.051968	49.730
Decoder	Decoder Layer 2	Linear Cross-Attention	V projection	0.051968	49.730
Decoder	Decoder Layer 2	Linear Cross-Attention	K $\times$ V MatMul	0.085658	10.246
Decoder	Decoder Layer 2	Linear Cross-Attention	Q $\times$ (K $\times$ V) MatMul	0.089828	0.795
Decoder	Decoder Layer 2	Linear Cross-Attention	Output projection	0.507891	35.948
Decoder	Decoder Layer 2	Channel MLP	Linear1	0.185715	52.579
Decoder	Decoder Layer 2	Channel MLP	Linear2	0.096953	27.449
Decoder	Decoder Layer 2	Weight MLP	Linear1	0.411755	58.287
Decoder	Decoder Layer 2	Weight MLP	Linear2	0.108500	15.359
Decoder	Decoder Layer 2	Sampling offset	Offset projection	0.171979	0.856
Decoder	Decoder Layer 2	Sampling weight	Weight projection	0.171979	1.284
Decoder	Decoder Layer 2	Detection embedding	Conv1	0.178439	12.630
Decoder	Decoder Layer 2	Detection embedding	Conv2	0.175566	12.426
Decoder	Decoder Layer 2	Classification	Hidden projection	0.172743	12.227

Continued on next page

Table 6 (continued)

Phase	Stage/Layer	Module	Operation	R_{LIF}	Energy (μJ)
Decoder	Decoder Layer 2	Classification	Class output	0.192230	0.159
Decoder	Decoder Layer 2	Box regression	Conv1	0.172743	12.227
Decoder	Decoder Layer 2	Box regression	Conv2	0.149534	10.584
Decoder	Decoder Layer 2	Box regression	Box output	0.156403	0.173
Decoder	Decoder Layer 3	Self-Attention	Q/K/V projections	0.180975	38.428
Decoder	Decoder Layer 3	Self-Attention	Q$\times$K^{T} MatMul	0.097223	8.064
Decoder	Decoder Layer 3	Self-Attention	Score$\times$V MatMul	0.080327	6.663
Decoder	Decoder Layer 3	Self-Attention	Output projection	0.194011	13.732
Decoder	Decoder Layer 3	Linear Cross-Attention	Q projection	0.181585	12.852
Decoder	Decoder Layer 3	Linear Cross-Attention	K projection	0.051968	49.730
Decoder	Decoder Layer 3	Linear Cross-Attention	V projection	0.051968	49.730
Decoder	Decoder Layer 3	Linear Cross-Attention	K$\times$V MatMul	0.086187	10.309
Decoder	Decoder Layer 3	Linear Cross-Attention	Q$\times$(K$\times$V) MatMul	0.085007	0.752
Decoder	Decoder Layer 3	Linear Cross-Attention	Output projection	0.486703	34.448
Decoder	Decoder Layer 3	Channel MLP	Linear1	0.174608	49.434
Decoder	Decoder Layer 3	Channel MLP	Linear2	0.090811	25.710
Decoder	Decoder Layer 3	Weight MLP	Linear1	0.423280	59.919
Decoder	Decoder Layer 3	Weight MLP	Linear2	0.095040	13.454
Decoder	Decoder Layer 3	Sampling offset	Offset projection	0.178439	0.888
Decoder	Decoder Layer 3	Sampling weight	Weight projection	0.178439	1.332
Decoder	Decoder Layer 3	Detection embedding	Conv1	0.171258	12.121
Decoder	Decoder Layer 3	Detection embedding	Conv2	0.170543	12.071
Decoder	Decoder Layer 3	Classification	Hidden projection	0.157660	11.159
Decoder	Decoder Layer 3	Classification	Class output	0.178812	0.148
Decoder	Decoder Layer 3	Box regression	Conv1	0.157660	11.159
Decoder	Decoder Layer 3	Box regression	Conv2	0.137887	9.759
Decoder	Decoder Layer 3	Box regression	Box output	0.156356	0.173
Decoder	Decoder Layer 4	Self-Attention	Q/K/V projections	0.172558	36.640
Decoder	Decoder Layer 4	Self-Attention	Q$\times$K^{T} MatMul	0.075623	6.273
Decoder	Decoder Layer 4	Self-Attention	Score$\times$V MatMul	0.075991	6.303
Decoder	Decoder Layer 4	Self-Attention	Output projection	0.022765	1.611
Decoder	Decoder Layer 4	Linear Cross-Attention	Q projection	0.172789	12.230
Decoder	Decoder Layer 4	Linear Cross-Attention	K projection	0.051968	49.730
Decoder	Decoder Layer 4	Linear Cross-Attention	V projection	0.051968	49.730
Decoder	Decoder Layer 4	Linear Cross-Attention	K$\times$V MatMul	0.092995	11.124
Decoder	Decoder Layer 4	Linear Cross-Attention	Q$\times$(K$\times$V) MatMul	0.092630	0.820
Decoder	Decoder Layer 4	Linear Cross-Attention	Output projection	0.438949	31.068
Decoder	Decoder Layer 4	Channel MLP	Linear1	0.169083	47.870
Decoder	Decoder Layer 4	Channel MLP	Linear2	0.092008	26.049
Decoder	Decoder Layer 4	Weight MLP	Linear1	0.415570	58.827
Decoder	Decoder Layer 4	Weight MLP	Linear2	0.088339	12.505
Decoder	Decoder Layer 4	Sampling offset	Offset projection	0.171258	0.852
Decoder	Decoder Layer 4	Sampling weight	Weight projection	0.171258	1.278
Decoder	Decoder Layer 4	Detection embedding	Conv1	0.165277	11.698
Decoder	Decoder Layer 4	Detection embedding	Conv2	0.156117	11.050
Decoder	Decoder Layer 4	Classification	Hidden projection	0.145397	10.291
Decoder	Decoder Layer 4	Classification	Class output	0.162365	0.135
Decoder	Decoder Layer 4	Box regression	Conv1	0.145397	10.291
Decoder	Decoder Layer 4	Box regression	Conv2	0.126819	8.976
Decoder	Decoder Layer 4	Box regression	Box output	0.144855	0.160
Decoder	Decoder Layer 5	Self-Attention	Q/K/V projections	0.163843	34.790
Decoder	Decoder Layer 5	Self-Attention	Q$\times$K^{T} MatMul	0.073965	6.135
Decoder	Decoder Layer 5	Self-Attention	Score$\times$V MatMul	0.090090	7.472
Decoder	Decoder Layer 5	Self-Attention	Output projection	0.338988	23.993
Decoder	Decoder Layer 5	Linear Cross-Attention	Q projection	0.166717	11.800
Decoder	Decoder Layer 5	Linear Cross-Attention	K projection	0.051968	49.730
Decoder	Decoder Layer 5	Linear Cross-Attention	V projection	0.051968	49.730
Decoder	Decoder Layer 5	Linear Cross-Attention	K$\times$V MatMul	0.091110	10.898
Decoder	Decoder Layer 5	Linear Cross-Attention	Q$\times$(K$\times$V) MatMul	0.093084	0.824
Decoder	Decoder Layer 5	Linear Cross-Attention	Output projection	0.482485	34.150
Decoder	Decoder Layer 5	Channel MLP	Linear1	0.162390	45.975
Decoder	Decoder Layer 5	Channel MLP	Linear2	0.090954	25.751
Decoder	Decoder Layer 5	Weight MLP	Linear1	0.402774	57.016
Decoder	Decoder Layer 5	Weight MLP	Linear2	0.086214	12.204
Decoder	Decoder Layer 5	Sampling offset	Offset projection	0.165277	0.823
Decoder	Decoder Layer 5	Sampling weight	Weight projection	0.165277	1.234
Decoder	Decoder Layer 5	Detection embedding	Conv1	0.156238	11.058
Decoder	Decoder Layer 5	Detection embedding	Conv2	0.146468	10.367
Decoder	Decoder Layer 5	Classification	Hidden projection	0.145934	10.329
Decoder	Decoder Layer 5	Classification	Class output	0.164922	0.137
Decoder	Decoder Layer 5	Box regression	Conv1	0.145934	10.329
Decoder	Decoder Layer 5	Box regression	Conv2	0.133943	9.480
Decoder	Decoder Layer 5	Box regression	Box output	0.155784	0.172
Decoder	Decoder Layer 6	Self-Attention	Q/K/V projections	0.156352	33.199

Continued on next page

Table 6 (continued)

Phase	Stage/Layer	Module	Operation	R_{LIF}	Energy (μJ)
Decoder	Decoder Layer 6	Self-Attention	Q$\times$K^{T} MatMul	0.079884	6.626
Decoder	Decoder Layer 6	Self-Attention	Score$\times$V MatMul	0.105158	8.722
Decoder	Decoder Layer 6	Self-Attention	Output projection	0.579228	40.997
Decoder	Decoder Layer 6	Linear Cross-Attention	Q projection	0.158767	11.237
Decoder	Decoder Layer 6	Linear Cross-Attention	K projection	0.051968	49.730
Decoder	Decoder Layer 6	Linear Cross-Attention	V projection	0.051968	49.730
Decoder	Decoder Layer 6	Linear Cross-Attention	K$\times$V MatMul	0.092697	11.088
Decoder	Decoder Layer 6	Linear Cross-Attention	Q$\times$(K$\times$V) MatMul	0.081296	0.719
Decoder	Decoder Layer 6	Linear Cross-Attention	Output projection	0.483556	34.226
Decoder	Decoder Layer 6	Channel MLP	Linear1	0.155195	43.938
Decoder	Decoder Layer 6	Channel MLP	Linear2	0.089569	25.358
Decoder	Decoder Layer 6	Weight MLP	Linear1	0.397990	56.339
Decoder	Decoder Layer 6	Weight MLP	Linear2	0.081279	11.506
Decoder	Decoder Layer 6	Sampling offset	Offset projection	0.156238	0.778
Decoder	Decoder Layer 6	Sampling weight	Weight projection	0.156238	1.166
Decoder	Decoder Layer 6	Detection embedding	Conv1	0.147488	10.439
Decoder	Decoder Layer 6	Detection embedding	Conv2	0.144270	10.211
Decoder	Decoder Layer 6	Classification	Hidden projection	0.143846	10.181
Decoder	Decoder Layer 6	Classification	Class output	0.162402	0.135
Decoder	Decoder Layer 6	Box regression	Conv1	0.143846	10.181
Decoder	Decoder Layer 6	Box regression	Conv2	0.126734	8.970
Decoder	Decoder Layer 6	Box regression	Box output	0.145322	0.161
				Total	–	**49871.638**