SPIKEUAV-DETR: ENERGY-EFFICIENT END-TO-END SPIKE-DRIVEN OBJECT DETECTION FOR UNMANNED AERIAL VEHICLE IMAGERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Onboard detection for unmanned aerial vehicles (UAVs) can provide reliable, real-time detection results when network connectivity is limited. However, limited onboard resources require detectors to balance accuracy and energy consumption. Detectors based on spiking neural networks (SNNs) exploit spike-driven computation, resulting in lower energy consumption and making them promising for UAV applications. However, most real-time SNN detectors retain non-maximum suppression (NMS), which requires extensive tuning. Moreover, the discrete spike encoding inherent to SNNs leads to information loss and accuracy degradation. To address these challenges, we propose SpikeUAV-DETR, an end-to-end spike-driven detector that enables NMS-free inference. First, we employ time–channel adaptive firing thresholds that adjust to membrane-potential distributions, thereby improving spike information transmission. Second, we introduce high-resolution cross-level feature fusion to preserve fine-grained details and lightweight gating to inject high-level semantics, thereby enhancing small object detection. Third, we employ hybrid query selection and deformable attention modules to preserve precision with limited dense computation. Experimental results on two UAV imagery datasets show that SpikeUAV-DETR achieves a favorable accuracy–energy trade-off over representative detectors while retaining real-time inference, specifically improving mAP@50 by 1.1 percentage points over the ANN baseline RT-DETR-R18 on VisDrone2019-DET while reducing energy consumption by 52.4%.

1 INTRODUCTION

Unmanned aerial vehicles (UAVs) enable flexible, wide-area observation and have considerable potential for emergency rescue and traffic incident response. These tasks require real-time identification of key targets in UAV imagery to provide timely visual information for on-site operations (Schedl et al., 2021; Liu et al., 2024). Detecting small targets in aerial imagery often requires high-resolution detail (Yang et al., 2022), but transmitting large volumes of high-resolution images in real time can readily overload communication links, increasing latency and packet loss (Damigos et al., 2023). Disaster-related infrastructure damage and limited network coverage in remote areas can further compromise the reliability and availability of these links, making fully cloud-dependent real-time detection difficult to guarantee (Hazra et al., 2020; Schedl et al., 2021). Onboard detection instead processes sensor imagery locally, avoiding bulk raw-image transmission and providing stable, real-time detection results (Cao et al., 2023). However, given limited onboard computing capacity and energy, real-time detectors deployed on UAVs must balance detection accuracy and energy consumption, making efficient onboard object detection a pressing challenge (Liu et al., 2024).

Mainstream UAV object detectors are built on ANNs and achieve strong detection accuracy while supporting real-time inference (Kong et al., 2024; Zhang et al., 2025). However, their dense computation requires numerous multiply–accumulate operations and consequently incurs high energy consumption (Horowitz, 2014; Shen et al., 2025). In response to this challenge, SNNs have attracted increasing attention. By communicating through discrete spikes, an SNN can skip inactive computation on hardware that supports spike-driven execution and replace many multiply–accumulate operations with lower-energy accumulations (Su et al., 2023; Horowitz, 2014; Yao et al., 2023). Developing SNN-based algorithms for UAV object detection therefore offers a promising route toward

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.

reducing energy consumption while maintaining detection accuracy, thereby better accommodating the characteristics of aerial imagery and the deployment constraints of UAV platforms.

Recent studies have developed a range of SNN detectors that combine real-time inference, low energy consumption, and promising detection accuracy (Su et al., 2023; Fan et al., 2026). Most of these detectors adapt YOLO-based one-stage models to spike-driven computation and are directly trained for few-timestep inference. These methods achieve detection accuracy close to their ANN counterparts (Su et al., 2023; Fan et al., 2026), and have been extended to UAV small object detection (Shen et al., 2025; Wang et al., 2026). However, because they inherit the YOLO pipeline, these methods also retain NMS for post-processing, which requires careful tuning for different application scenarios (Zhang et al., 2025; Zhao et al., 2024). Spiking-DETR tries to address this limitation by introducing the DETR architecture into SNNs to perform end-to-end prediction (Ouyang & Jiang, 2024). However, its global self-attention becomes computationally costly at high feature resolutions, making it challenging to balance small object detection accuracy with real-time inference. This motivates efficient end-to-end spike-driven detectors for UAV applications that reduce reliance on manually tuned post-processing while meeting accuracy, latency, and energy constraints.

However, combining spike-driven computation with end-to-end set prediction poses a key challenge: preserving detection-relevant information during spike encoding, transmission, and feature transformation. Object detection requires both accurate classification and precise bounding-box regression; for small, dense, or occluded UAV objects, both predictions depend on preserving and accurately using limited semantic and spatial cues throughout the network. Thresholded sparse transmission can omit weak yet informative feature responses, while discrete firing quantizes membrane-potential amplitudes, thereby impairing both classification and localization (Wang et al., 2026; Fan et al., 2026). Reducing these losses requires better neuronal encoding of feature responses and network architectures designed to preserve and use spike information.

At the neuronal level, conventional IF and LIF neurons produce binary spikes that limit the resolution of membrane-potential responses (Wang et al., 2026; Fan et al., 2026). PLIF improves temporal adaptability by learning the membrane time constant (Fang et al., 2021), while I-LIF and LIMF improve response resolution through integer-valued training or multi-level firing (Luo et al., 2025; Wang et al., 2026). However, multi-level firing with fixed thresholds does not adapt to changes in neuronal response ranges and therefore weakens the retention and transmission of target information.

In network architecture design, SNN detectors incorporate convolutional and Transformer-based modules to improve feature extraction and interaction while retaining spike-driven computation (Su et al., 2023; Fan et al., 2024; Li et al., 2025b; Yao et al., 2024). UAV-oriented SNN detectors further improve aerial small object detection through high-resolution, multi-scale, and frequency-domain feature processing (Shen et al., 2025; Wang et al., 2026; Cai et al., 2026). However, how to use high-level semantics to enhance shallow fine-grained features and exploit them for small object detection at controlled computational cost remains underexplored. Moreover, adapting precision-sensitive detection mechanisms, such as deformable attention (Zhu et al., 2021b), to SNNs calls for designs that explicitly balance detection accuracy and energy consumption.

To overcome these challenges, we develop SpikeUAV-DETR, a spike-driven end-to-end detector for real-time UAV-based object detection. The detector employs bipartite matching during training and directly predicts a one-to-one set of objects, thereby avoiding complex post-processing parameter tuning. We further introduce adaptive multi-level neuronal dynamics, efficient high-resolution feature enhancement, and SNN–ANN hybrid modules. Together, these designs mitigate information loss while retaining the energy advantages of SNNs, enabling SpikeUAV-DETR to achieve both high detection accuracy and energy efficiency. Our main contributions are summarized as follows:

- **A time–channel adaptive multi-level spiking neuron:** We propose a new spiking neuron that adapts its firing-threshold sequence across time and channels, improving information transmission with minimal parameter overhead.
- **High-resolution feature fusion and gated semantic enhancement modules:** First, $\mathbf{S}_2$ fusion–downsampling (S2FD) performs cross-level fusion at high spatial resolution and uses channel-expanded downsampling to aggregate the fused information, preserving detailed features while controlling computation. Second, cross-scale gated fusion (CSGF) is designed to selectively supplement shallow features with high-level semantic information through lightweight gating, thereby improving small object detection.

- **SNN–ANN hybrid modules for precision-sensitive operations:** First, we design time-aware query selection with an SNN block for costly feature projection and an ANN block that generates and aggregates continuous candidate scores across timesteps to select initial object queries. Second, we introduce SNN–ANN hybrid deformable attention, assigning dense projections to spiking blocks and precision-sensitive operations to ANN branches to balance feature-transformation energy efficiency with computational accuracy.

2 Related Work

UAV object detection. Reliable UAV object detection is fundamental to aerial perception, yet small, dense, and occluded targets together with real-time inference requirements make it challenging. Current mainstream UAV object detection methods are predominantly based on artificial neural network (ANN) architectures and largely follow the YOLO and DETR paradigms. YOLO-based methods are widely adopted for their efficient single-stage dense prediction, while UAV-oriented variants further improve detection performance through additional prediction scales, high-resolution prediction layers, attention-enhanced detection heads, and enhanced feature fusion (Zhu et al., 2021a; Zhang et al., 2021; Tang et al., 2024). However, these methods still rely on manually designed post-processing procedures, such as non-maximum suppression (NMS). In contrast, DETR formulates object detection as end-to-end set prediction, eliminating such complex post-processing (Carion et al., 2020). Deformable DETR and RT-DETR further improve multi-scale processing and inference efficiency (Zhu et al., 2021b; Zhao et al., 2024), and subsequent variants adapt this paradigm to UAV imagery (Kong et al., 2024; Zhang et al., 2025; Liao et al., 2026b). Despite their strong accuracy and efficiency, these ANN detectors rely predominantly on dense multiply–accumulate computation, motivating spike-driven alternatives for energy-constrained UAV platforms.

Spike-driven object detection. Spike-driven object detection exploits sparse spiking computation to reduce energy consumption. Early SNN detectors mainly relied on ANN-to-SNN conversion, but typically required hundreds or thousands of timesteps, resulting in high inference latency (Kim et al., 2020; Wang et al., 2026). Direct training with surrogate gradients enables few-timestep inference and has led to increasingly capable detectors, including EMS-YOLO (Su et al., 2023), SFOD (Fan et al., 2024), MSD (Li et al., 2025b), and SpikeDet (Fan et al., 2026). Recent UAV-oriented SNN detectors further improve aerial small object detection through multi-scale feature fusion, multi-level firing, and the use of frequency-domain features (Shen et al., 2025; Wang et al., 2026; Cai et al., 2026). However, most real-time SNN detectors still follow YOLO pipelines and retain NMS, while Spiking-DETR enables end-to-end spike-driven detection but inherits the computational burden of the original DETR pipeline (Ouyang & Jiang, 2024; Zhao et al., 2024). Thus, real-time end-to-end spike-driven detection for UAV imagery remains insufficiently explored.

Spiking neurons. Spiking neurons determine how SNNs integrate, encode, and transmit information, thereby affecting object-detection performance. IF and LIF neurons are widely used in SNN detectors, but their simple designs limit information transmission. PLIF improves temporal adaptability through learnable membrane time constants (Fang et al., 2021; Fan et al., 2024; Ahmed et al., 2025), while EMM-Det introduces dynamic leakage constants to prevent spike decay and improve long-term memory capacity (Cai et al., 2026). To reduce quantization error, I-LIF improves response resolution through integer-valued responses while preserving spike-driven inference (Luo et al., 2025). LIMF further reduces this error through weighted multi-level firing (Wang et al., 2026). However, existing multi-level firing neurons generally use fixed threshold ladders that cannot accommodate varying membrane-potential ranges, motivating adaptive threshold adjustment.

3 Methods

3.1 Overall Architecture

As shown in Fig. 1, SpikeUAV-DETR is built on RT-DETR (Zhao et al., 2024) and comprises three components: a spiking backbone, a detail-enhanced encoder, and an SNN–ANN hybrid decoder.

Input and backbone. The input is $\mathbf{X} \in \mathbb{R}^{3\times H\times W}$. The EMS-ResNet34 backbone (Su et al., 2023) applies the initial convolution once, expands the resulting features across T timesteps, and extracts multi-scale spiking features $\mathbf{S}_2$–$\mathbf{S}_5$, which are fed into the encoder.

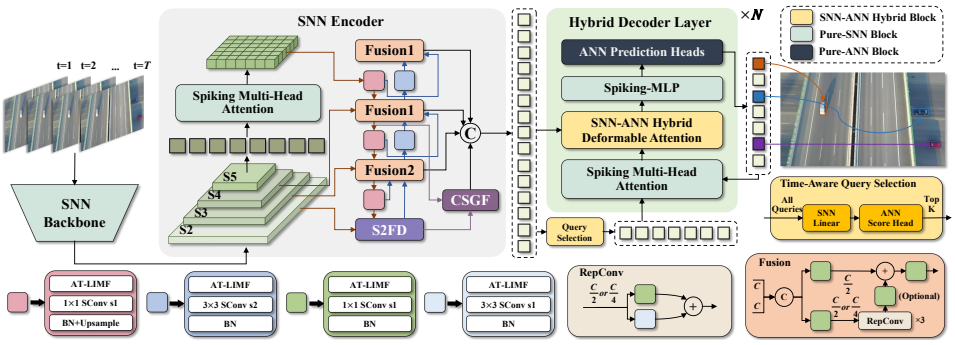

Figure 1: Overall architecture of the SpikeUAV-DETR.

Encoder. The encoder applies Spike-Driven Self-Attention (SDSA-3) to $\mathbf{S}_5$ for global context modeling (Yao et al., 2024), followed by top-down and bottom-up multi-scale fusion. To reduce the cost of high-resolution fusion, Fusion2 adopts a branch width of $C/4$. At the end of the top-down pathway, S2FD combines $\mathbf{S}_2$ with the upsampled feature from the $\mathbf{S}_3$ level and downsamples the fused result. CSGF further enhances the S2FD-derived feature with high-level semantics. Finally, the four encoder outputs are flattened and concatenated for subsequent query selection and decoding.

Query selection and decoder. Time-aware query selection aggregates candidate scores across timesteps to initialize decoder queries. Each decoder layer retains SDSA-3 and the spiking MLP (Yao et al., 2024), while incorporating SNN–ANN hybrid deformable attention. In the hybrid deformable attention, dense feature projections are spike-driven, whereas precision-sensitive operations remain ANN-based. The decoder iteratively refines queries and reference boxes, and aggregates the final features across timesteps before ANN prediction heads produce the predictions.

3.2 Adaptive-Threshold Leaky Integrate-and-Multi-Fire Neuron

Motivation. Multi-level spiking neurons have been widely used in object detection to reduce spike quantization error (Luo et al., 2025; Wang et al., 2026). However, these multi-level spiking neurons generally use fixed threshold sequences, so their discretization granularity and representable range cannot be optimized jointly with the synaptic weights. We therefore develop adaptive-threshold leaky integrate-and-multi-fire (AT-LIMF) neurons based on LIMF, in which each layer learns time- and channel-dependent threshold scales, allowing its threshold sequence to adapt to membrane-potential distributions. Detailed background and design motivation are provided in Appendix C.1.

Time–channel adaptive thresholds. In LIMF, the fixed threshold sequence $\{2^k V_{\mathrm{th}} \mid k = -m, \ldots, n\}$ defines $n + m + 1$ binary firing bits. Building on this structure, each AT-LIMF layer learns a time–channel scaling matrix $\mathbf{A} = [a_{t,c}] \in \mathbb{R}^{T\times C}$ and shares it across the remaining dimensions. With adaptive scaling, the base threshold $\theta_{t,c}$ and minimum firing interval $\Delta_{t,c}$ become:

$$\theta_{t,c} = V_{\mathrm{th}} \exp(a_{t,c}) = 2^m \Delta_{t,c}, \qquad t \in \{1, \ldots, T\}, \quad c \in \{1, \ldots, C\}. \tag{1}$$

Here, $a_{t,c}$ is a learnable threshold-scaling parameter and is mapped exponentially to ensure a positive scale; V_{th} is the initial base threshold. The resulting adaptive threshold sequence $\{2^k\theta_{t,c} \mid k = -m, \ldots, n\}$ preserves the original firing bits while jointly adjusting their discretization granularity and representable range across timesteps and channels. Together, these bits encode $Q_{\max} + 1$ response levels, where $Q_{\max} = 2^{n+m+1} - 1$. Smaller scales provide finer amplitude resolution over a narrower range, whereas larger scales extend the range with coarser resolution. This design adds only TC parameters per layer and ensures spatial translation equivariance; further analysis of its advantages is provided in Appendix C.2.

Neuronal dynamics. Let $I_{t,c}$ and $h_{t-1,c}$ denote the current synaptic input and the post-reset membrane potential from the previous timestep, respectively. Omitting layer, batch, and spatial or token indices and setting $h_{0,c} = 0$, AT-LIMF updates the pre-firing membrane potential as:

$$v_{t,c} = (1 - \lambda^{-1})h_{t-1,c} + I_{t,c}. \tag{2}$$

Here, λ is the membrane time constant. AT-LIMF then discretizes the membrane potential using the minimum firing interval $\Delta_{t,c}$ as the quantization step, yielding the normalized output $s_{t,c}$ and

post-reset membrane potential $h_{t,c}$:

$$s_{t,c} = \frac{1}{Q_{\max}} \left\lfloor \mathrm{clip}\left(\frac{v_{t,c}}{\Delta_{t,c}},\, 0,\, Q_{\max} \right) \right\rfloor, \tag{3}$$

$$h_{t,c} = v_{t,c} - \mathrm{sg}(Q_{\max} s_{t,c} \Delta_{t,c})\,. \tag{4}$$

The output $s_{t,c} \in [0, 1]$ selects one of $Q_{\max} + 1$ normalized response levels and is passed to subsequent layers. This normalization standardizes the response range, leaving amplitude scaling to downstream synaptic weights and mitigating magnitude growth during spike-matrix multiplication in attention modules. In the soft reset, $\mathrm{sg}(\cdot)$ acts as the identity in the forward pass and blocks gradients through the reset subtraction during backpropagation.

Training and inference. During training, we initialize all $a_{t,c}$ to zero and jointly optimize the scaling parameters and synaptic weights by backpropagating the detection loss. At inference, the learned time–channel threshold parameters remain fixed, and the response can be decomposed losslessly into weighted binary firing bits, thereby preserving the energy advantage of spike-driven computation. Training and inference design details are provided in Appendix C.3.

3.3 Detail-Preserving S2 Fusion and Gate-Controlled Semantic Enhancement

Shallow features preserve rich fine-grained spatial details that are particularly important for small object detection. However, the original RT-DETR encoder uses only $\mathbf{S}_3$–$\mathbf{S}_5$, leaving the higher-resolution $\mathbf{S}_2$ feature outside cross-scale encoding, which may discard fine-grained details of small objects. To better exploit the information in $\mathbf{S}_2$ while controlling additional computation, we introduce S2FD and CSGF, as shown in Fig. 2.

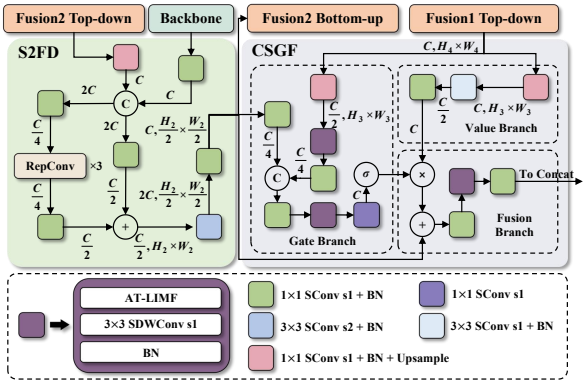

Figure 2: Structure of the proposed S2FD and CSGF.

$\mathbf{S}_2$ fusion–downsampling (S2FD). S2FD follows the basic structure of the RT-DETR fusion module and adapts it to the high spatial resolution of $\mathbf{S}_2$. The projected $\mathbf{S}_2$ and the twofold-upsampled top-down feature from $\mathbf{S}_3$ are concatenated and processed by the dual-branch fusion structure to obtain the high-resolution fused feature $\mathbf{Z}_2$. A stride-2 3×3 SNN convolution expands $\mathbf{Z}_2$ from $C/2$ to $2C$ channels, followed by a 1×1 SNN convolution that compresses it to C channels and produces $\mathbf{F}_{\mathrm{S2FD}}$. The expanded intermediate feature supports information aggregation before compression, helping preserve informative details from high-resolution features. Convolutional downsampling reduces the spatial resolution to that of $\mathbf{S}_3$ while preserving the regular spatial structure, facilitating subsequent deformable attention and limiting downstream computational cost. The resulting $\mathbf{F}_{\mathrm{S2FD}}$ is shared by the bottom-up pathway and CSGF.

Cross-scale gated fusion (CSGF). CSGF enhances $\mathbf{F}_{\mathrm{S2FD}}$ using the high-level feature from the first top-down fusion of $\mathbf{S}_5$ and $\mathbf{S}_4$. This feature incorporates information from two higher backbone stages and requires only one twofold upsampling operation for spatial alignment. Since the contribution of high-level semantics varies across timesteps, channels, and spatial locations, CSGF uses a learned gate jointly determined by the aligned S2FD and high-level features. Specifically, the two features are transformed, concatenated, and processed by lightweight depthwise and pointwise SNN convolutions to produce the gate feature $\mathbf{U}_{\mathrm{g}}$ with limited additional computation. The final gate projection omits batch normalization, allowing its bias to control the overall activation tendency of the gate. In parallel, the value branch transforms the aligned high-level feature into the semantic value $\mathbf{V}_4$. The gated semantic injection is formulated as:

$$\widetilde{\mathbf{Y}}_2 = \mathbf{F}_{\mathrm{S2FD}} + \sigma(\mathbf{U}_{\mathrm{g}}) \odot \mathbf{V}_4, \tag{5}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\odot$ denotes element-wise multiplication. The residual path preserves the fine-grained information inherited from $\mathbf{S}_2$, while the learned gate selectively introduces complementary high-level semantics. The fused feature is finally refined by a lightweight

SNN structure to produce $\mathbf{Y}_2$, which preserves spatial cues from $\mathbf{S}_2$ while incorporating high-level semantic context, providing complementary information for small object detection.

3.4 SNN–ANN Hybrid Design for Query Selection and Deformable Attention

Query selection ranks candidates using fine-grained class scores, while deformable attention uses continuous sampling offsets and attention weights to sample and aggregate continuous-valued features. These processes rely on precision-sensitive continuous-valued computations. Applying spike-driven computation to these operations would cause two problems. First, spike quantization would impair continuous predictions such as class scores and sampling offsets. Second, approximating multiplication between continuous values with synaptic operations would also introduce additional error. Both would impair detection accuracy. Conversely, applying a fully ANN-based pipeline at every timestep increases energy consumption and fails to exploit temporal information effectively. We therefore use SNN–ANN hybrid designs: spike-driven blocks perform dense feature transformations and temporal integration, while retaining ANN layers for precision-sensitive operations.

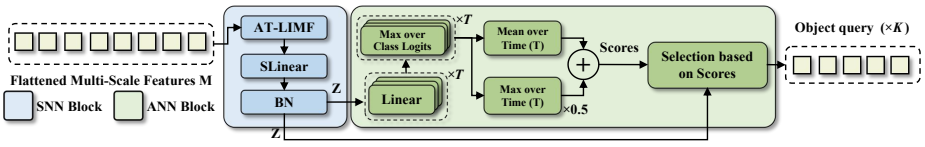

Figure 3: Structure of the proposed time-aware query selection.

Time-aware query selection. Let $\mathbf{M} \in \mathbb{R}^{T\times B\times N\times C}$ denote the flattened and concatenated encoder features. As shown in Fig. 3, an SNN linear block projects $\mathbf{M}$ into candidate features $\mathbf{Z}$, which are scored at each timestep by a shared ANN classification head. The scores are then aggregated across timesteps to obtain a common selection score s_i for each candidate:

$$s_i = T^{-1}\sum_{t=1}^{T} \max_{\kappa} \ell_{t,i,\kappa} + 0.5 \max_{t,\kappa} \ell_{t,i,\kappa}, \qquad i = 1, \ldots, N, \tag{6}$$

The score in equation 6 combines sustained and transient responses across timesteps, allowing query selection to exploit temporal evidence more comprehensively. The top-K candidates ($K = 300$) are then selected as the initial object queries for the decoder. The costly all-token projection is performed by the spike-driven block, while continuous-valued computation is restricted to class-score prediction and temporal score aggregation, limiting MAC and energy overhead.

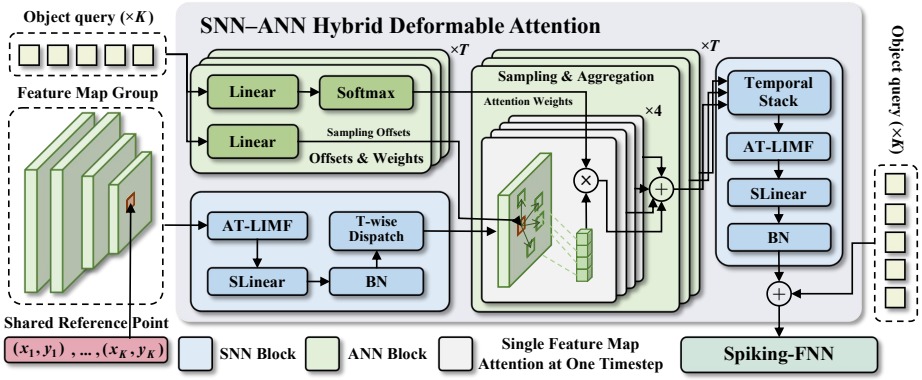

Figure 4: Structure of the proposed SNN–ANN hybrid deformable attention.

SNN–ANN hybrid deformable attention. As shown in Fig. 4, the hybrid deformable attention takes the K selected object queries and the multi-scale feature sequence $\mathbf{M}$ as inputs. An SNN linear block first performs value projection over all tokens in $\mathbf{M}$, which accounts for the main dense feature transformation. For each timestep independently, shared ANN branches first predict continuous sampling offsets and normalized attention weights, and then perform deformable sampling and weighted aggregation. Finally, the SNN output projection transforms the aggregated features into a residual term, which is used to update the K selected object queries. Since the retained ANN branches and continuous-valued computation operate only on the K selected object queries and a fixed number of sparse sampling locations, their additional computation and energy overhead

are effectively constrained. This allocation preserves the continuous-valued precision required for offset prediction, attention-weight estimation, and deformable sampling, while retaining the energy-efficiency and temporal-processing advantages of spike-driven feature processing.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. SpikeUAV-DETR is primarily evaluated on VisDrone2019-DET (Zhu et al., 2022), a large-scale UAV benchmark covering ten object categories, using 6,471 training images and 548 validation images for training and evaluation, respectively. We further evaluate cross-dataset generalization on UAVVaste (Kraft et al., 2021), an aerial rubbish detection dataset, using 540 training images for training and 116 test images for final evaluation (see Appendix A.1 for details).

Implementation details. All experiments are conducted with PyTorch on NVIDIA H800 GPUs. Input images are resized to 640×640 and augmented with Mosaic (Bochkovskiy et al., 2020). The COCO-pretrained EMS-ResNet34 backbone (Su et al., 2023) is trained for 120 epochs using AdamW with a batch size of 6. Following SpikeSOD (Wang et al., 2026), we set $n = 2$, $m = 3$, and $V_{\text{th}} = 1$ for multi-level firing. Inference speed is measured on a single H800 GPU with a batch size of 1. Further implementation details are provided in Appendix A.2.

Detection and efficiency metrics. Detection accuracy is evaluated using AP@50 and AP@50:95 for individual categories and mAP@50 and mAP@50:95 across all categories, with object scales defined following the COCO standard (Lin et al., 2014). Inference efficiency is measured in frames per second (FPS). Computational cost is reported as floating-point operations (FLOPs) required for multiply–accumulate operations (MACs) and effective synaptic operations (SOPs) for spike-driven computation. Detailed metric definitions are provided in Appendix A.3.

Energy estimation. Following common SNN energy evaluation practice (Su et al., 2023; Luo et al., 2025), the model energy is estimated as:

$$E_{\text{model}} = 0.5\, e_{\text{MAC}} N_{\text{FLOPs}} + e_{\text{AC}} N_{\text{SOPs}}. \tag{7}$$

Here, N_{FLOPs} denotes the number of floating-point operations required for the MACs. N_{SOPs} denotes the total number of effective synaptic operations performed through accumulate operations (ACs) in spike-driven computation. SOPs are estimated from each SNN module's structure, average firing rate, and T timesteps, then summed to obtain N_{SOPs}. We assume a 32-bit floating-point implementation in 45-nm CMOS technology for all models, with $e_{\text{MAC}} = 4.6\,\text{pJ}$ and $e_{\text{AC}} = 0.9\,\text{pJ}$ (Horowitz, 2014). The detailed estimation method is provided in Appendix B.

4.2 COMPARATIVE EXPERIMENTS

Comparison with baselines. Table 1 compares the overall and scale-wise accuracy of SpikeUAV-DETR($T = 3$) with its ANN baseline RT-DETR-R18 and a spike-driven baseline. The spike-driven baseline disables all proposed components. It uses LIMF, three-level feature fusion, and ANN query selection and deformable attention on temporally averaged features. Our detector achieves the best overall performance. The gain is most pronounced for small objects, whose mAP@50 exceeds that of the spike-driven baseline and RT-DETR-R18 by 7.8 and 2.4 percentage points, respectively. These results demonstrate that the proposed design strengthens fine-grained feature representation, particularly for small object detection. Visual comparisons are provided in Appendix D.

Table 1: Comparison with baselines on VisDrone2019-DET.

Scale	RT-DETR-R18		Spike-driven Baseline		Ours ($T = 3$)	
	mAP @50/%	mAP @50:95/%	mAP @50/%	mAP @50:95/%	mAP @50/%	mAP @50:95/%
Small	34.8	17.1	29.4	13.6	**37.2**	**18.6**
Medium	58.6	37.4	52.6	31.9	**59.2**	**37.8**
Large	**65.8**	**49.0**	54.5	40.8	58.7	44.9
Overall	44.5	26.1	38.7	21.8	**46.0**	**27.0**

Comparison with representative detectors on VisDrone2019-DET. As shown in Table 2, SpikeUAV-DETR offers a competitive balance between accuracy and energy across ANN detectors with different computational budgets. Compared with lightweight, low-computation detectors such as YOLOv11-N, the $T = 2$ configuration consumes more energy but achieves substantially higher

mAP@50. Against stronger medium-computation detectors such as HIC-YOLOv5 and RT-DETR-R18, the $T = 2$ configuration achieves higher mAP@50 with lower energy consumption. Relative to the high-computation, high-performing RT-DETR-R50, the $T = 3$ configuration trails by only 1.0 percentage point in mAP@50 while consuming 70.0% less energy. Notably, both configurations outperform all DETR-based comparators except RT-DETR-R50 in mAP@50 and use substantially less energy than every DETR-based comparator. This supports the effectiveness of the proposed modifications to the DETR architecture. Among the evaluated SNN detectors, SpikeUAV-DETR achieves the highest mAP@50 and performs end-to-end prediction, avoiding complex post-processing parameter tuning across UAV-based object detection scenarios.

Table 2: Comparison with representative detectors on VisDrone2019-DET.

Type	ID	Model[a]	$T \times D$[b] (T)	Base Detector	Input Size	Params (M)	GFLOPs[c] + GSOPs	Energy (mJ/img)	mAP[d] @50/%
ANN	1	YOLOv8-N (Jocher et al., 2023)	1	YOLO	640 × 640	3.0	8.2	18.9	26.9
	2	YOLOv8-M (Jocher et al., 2023)	1	YOLO	640 × 640	25.9	78.9	181.5	40.7
	3	YOLOv8-L (Jocher et al., 2023)	1	YOLO	640 × 640	43.7	165.2	380.0	42.7
	4	YOLOv11-N (Khanam & Hussain, 2024)	1	YOLO	640 × 640	2.6	6.5	15.0	33.4
	5	YOLOv11-S (Khanam & Hussain, 2024)	1	YOLO	640 × 640	9.4	21.3	49.0	38.7
	6	YOLOv11-M (Khanam & Hussain, 2024)	1	YOLO	640 × 640	20.0	67.7	155.7	43.1
	7	FDB-YOLO (Song et al., 2024)	1	YOLO	640 × 640	8.5	62.4	143.5	39.9
	8	HIC-YOLOv5 (Tang et al., 2024)	1	YOLO	640 × 640	9.3	31.2	71.8	44.3
	9	QueryDet (Yang et al., 2022)	1	RetinaNet	2400 × 2400	33.9	212.0	487.6	**48.1**
	10	DREB-Net (Li et al., 2025a)	1	CenterNet	1024 × 1024	30.3	206.9	475.9	34.4
	11	DETR (Carion et al., 2020)	1	DETR	1333 × 750	60.0	187.0	430.1	40.1
	12	Deformable DETR (Zhu et al., 2021b)	1	DETR	1333 × 800	40.0	173.0	397.9	42.2
	13	RT-DETR-R18 (Zhao et al., 2024)	1	DETR	640 × 640	20.1	60.0	138.0	44.5
	14	RT-DETR-R50 (Zhao et al., 2024)	1	DETR	640 × 640	42.0	136.0	312.8	47.0
SNN	15	EMS-YOLO-R34 (Su et al., 2023)	5	YOLO	640 × 640	25.3	1.9+38.4	39.0	20.9
	16	SpikeYOLO-L (Luo et al., 2025)	1 × 4	YOLO	640 × 640	68.7	0.7+92.1	84.6	43.6
	17	Spiking Trans-YOLO (Huo et al., 2025)	1 × 4	YOLO	640 × 640	26.2	0.5+34.8	32.4	40.1
	18	SNN-Drone-S (Liao et al., 2026a)	1 × 4	YOLO	640 × 640	23.1	–	8.6	40.2
Ours	19	SpikeUAV-DETR	2 × 6	DETR	640 × 640	32.1	3.2+64.7	65.7	45.6
	20	SpikeUAV-DETR	3 × 6	DETR	640 × 640	32.2	3.7+94.9	93.8	<u>46.0</u>

[a] Some entries use reproduction results reported by other studies (Zhang et al., 2025; Wang et al., 2026). [b] $T \times D$ is reported for detectors using multi-level firing neurons, where D denotes the number of binary firing places per timestep. [c] For SNN-containing models, computational cost is reported as GFLOPs + GSOPs. [d] Bold indicates the best overall result; underlining highlights the best result of our detectors.

Energy–accuracy trade-off analysis. Figure 5 clearly presents the relationship between detection accuracy and energy consumption, with point labels corresponding to the model IDs in Table 2. Models in the upper-right region generally achieve high accuracy at high energy cost, whereas those in the lower-left region consume less energy but provide lower accuracy. In contrast, both SpikeUAV-DETR configurations are located in the upper-left region, indicating that the proposed model achieves detection accuracy comparable to that of high-complexity detectors at substantially lower energy consumption and further validating its balance between detection accuracy and energy consumption.

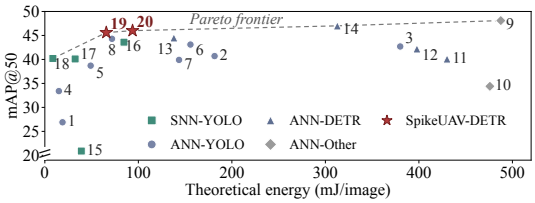

Figure 5: Energy–accuracy comparison.

Inference performance. Table 3 compares the inference speeds of SpikeUAV-DETR and representative real-time ANN-YOLO, DETR, and SNN-YOLO detectors. Against medium-to-large ANN detectors, the $T = 2$ configuration operates within the throughput range of RT-DETR-R50 and YOLOv8-L, with speed differences below 10%. Against small-to-medium ANN detectors, even the faster $T = 2$ configuration is 42.3% slower than the original ANN baseline RT-DETR-R18, showing that temporal spike processing and the proposed modifications introduce additional inference latency. Against the SNN comparator EMS-YOLO, both temporal configurations achieve higher inference speeds, with a maximum improvement of 38.9%. These results show that, despite the additional latency, SpikeUAV-DETR retains real-time inference capability across different temporal configurations.

Table 3: Inference speed comparison.

ID	Model	GFLOPs+GSOPs	FPS
3	YOLOv8-L	165.2	96.9
6	YOLOv11-M	67.7	149.9
13	RT-DETR-R18	60.0	151.7
14	RT-DETR-R50	136.0	80.6
15	EMS-YOLO-R34	1.9 + 38.4	63.0
19	SpikeUAV-DETR ($T = 2$)	3.2 + 64.7	87.5
20	SpikeUAV-DETR ($T = 3$)	3.7 + 94.9	65.0

Note: IDs correspond to the detectors in Table 2.

Comparison with representative detectors on UAVVaste. As shown in Table 4, SpikeUAV-DETR at $T = 3$ outperforms all compared ANN and SNN detectors in mAP@50:95. At $T = 2$, our model achieves higher detection accuracy than most compared models while offering considerable energy savings, achieving a favorable accuracy–energy trade-off. These results show that SpikeUAV-DETR retains low energy consumption and competitive detection accuracy when adapted to different UAV-based object detection tasks, demonstrating its generalization across detection scenarios.

Table 4: Comparison with representative detectors on UAVVaste.

Type	Model	$T \times D$ (T)	Base Detector	Input Size	Params (M)	GFLOPs + GSOPs	Energy (mJ/img)	mAP @50:95/%	mAP @50/%
ANN	YOLOv8-L (Jocher et al., 2023)	1	YOLO	640×640	43.7	165.4	380.4	46.3	75.0
	HIC-YOLOv5 (Tang et al., 2024)	1	YOLO	640×640	9.3	30.5	70.3	45.2	77.2
	RT-DETR-R18 (Zhao et al., 2024)	1	DETR	640×640	20.1	60.4	138.9	44.3	78.4
	RT-DETR-R50 (Zhao et al., 2024)	1	DETR	640×640	42.0	137.4	316.1	47.8	**81.2**
	UAV-DETR-R50 (Zhang et al., 2025)	1	DETR	640×640	42.0	170.0	391.0	37.5	75.9
SNN	EMS-YOLO-R34 (Su et al., 2023)	5	YOLO	640×640	25.3	$1.9 + 36.6$	37.4	42.4	75.8
	SpikeYOLO-L (Luo et al., 2025)	1×4	YOLO	640×640	68.7	$0.7 + 91.2$	83.7	48.5	78.8
Ours	SpikeUAV-DETR	2×6	DETR	640×640	32.1	$2.9 + 64.4$	64.7	47.4	79.3
	SpikeUAV-DETR	3×6	DETR	640×640	32.1	$3.2 + 94.8$	92.6	**49.0**	80.4

4.3 Ablation Studies

NMS sensitivity and end-to-end prediction. To further assess post-processing effects on SNN detectors, we used EMS-YOLO to examine how NMS confidence and IoU thresholds affect Precision and Recall. Both thresholds affect these metrics, with non-monotonic changes observed when varying the IoU threshold. These results highlight the need for extensive tuning of such post-processing. In contrast, SpikeUAV-DETR avoids this complex post-processing tuning through one-to-one set prediction, simplifying adaptation across scenarios (see Appendix E for details).

Component ablation. We conducted ablation experiments on VisDrone2019-DET using SpikeUAV-DETR at $T = 3$ to analyze the impact of each component on model performance. In Table 5, Baseline denotes the spike-driven baseline obtained by disabling all proposed improvements, and Hybrid Q/D denotes time-aware query selection and SNN–ANN hybrid deformable attention.

Table 5: Ablation results on VisDrone2019-DET.

Baseline	AT-LIMF	S2FD+CSGF	Hybrid Q/D	mAP@50	mAP@50:95	Params(M)	Energy(mJ/img)	FPS
√	–	–	–	38.7	21.8	30.378	63.6	97.5
√	√	–	–	39.3	22.0	30.440	62.9	96.3
√	√	√	–	44.6	25.8	32.145	108.1	70.9
√	√	√	√	**46.0**	**27.0**	32.154	93.8	65.0

AT-LIMF improves mAP@50 from 38.7% to 39.3% with negligible parameter and runtime overhead. This result shows that time–channel adaptive thresholds improve multi-level spike representation and information transmission at low complexity cost. S2FD and CSGF yield the largest accuracy improvement, increasing mAP@50 by 5.3 percentage points. This demonstrates the effectiveness of aggregating fine-grained shallow information and supplementing it with high-level semantics. However, high-resolution processing introduces additional computational overhead and model complexity, increasing parameter count and energy consumption by 5.6% and 71.9%, respectively, and reducing FPS by 26.4%. Hybrid Q/D further improves accuracy, with the complete model reaching 46.0% mAP@50. Compared with the preceding configuration, it adds only 0.009M parameters while reducing energy consumption by 13.2%. These results indicate that the hybrid design effectively combines energy-efficient spike-driven blocks with precision-preserving ANN blocks.

5 Conclusion

We introduce SpikeUAV-DETR, a spike-driven end-to-end detector for UAV imagery, combining time–channel adaptive multi-level firing, fine-grained cross-scale feature enhancement, and SNN–ANN hybrid designs for query selection and deformable attention. Experiments demonstrate that SpikeUAV-DETR achieves a favorable accuracy–energy trade-off over representative ANN and SNN detectors while retaining real-time end-to-end inference, supporting its potential for resource-constrained UAV perception. Future work will explore deployment on neuromorphic hardware.

AI USE STATEMENT

In this work, we used generative AI tools for translation, language editing, and assisting with the drafting of initial versions of selected sentences and paragraphs of the manuscript. This included improving readability, refining wording, and assisting with the presentation and organization of the manuscript. We also used generative AI tools for information retrieval and literature discovery, including searching for relevant work and supporting the collection of related information. In addition, generative AI tools were used to assist with method implementation and software development, primarily through code writing and debugging. Generative AI tools were also used to provide feedback on selected design details of the research methodology and experiments, and to assist with mathematical derivations and the writing of proofs.

We did not use generative AI tools to develop theoretical or conceptual frameworks, formulate mathematical claims, propose or refine research hypotheses, or interpret experimental results. Other required-disclosure tasks, including synthetic data generation, dataset cleaning or reformatting, and qualitative or thematic data analysis, were not applicable to this work.

All AI-assisted outputs were carefully reviewed by the authors and, where necessary, revised before being incorporated into the work. AI-assisted drafts were checked for accuracy and consistency with the underlying research and were revised and polished as necessary before inclusion in the final manuscript. Retrieved literature and information were manually checked against the original sources. AI-assisted code was reviewed, tested, and validated by the authors, while feedback on methodological or experimental design details and assistance with mathematical derivations and proofs were independently checked and verified by the authors.

The authors take full responsibility for the final content of this work, including all text, claims, code, experimental results, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work presents a methodological contribution and is evaluated exclusively on publicly available UAV imagery benchmarks. No private, sensitive, or personally identifying data are collected or used. We do not anticipate direct harms arising from our experiments; however, UAV-based object detection systems may raise privacy concerns or be misused in surveillance-related applications.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide detailed descriptions of the proposed SpikeUAV-DETR architecture and its components in the main text. All experimental settings are provided in Appendix A. During the review process, we will provide reviewers and area chairs with access to an anonymous code repository via the discussion forum, and the code will be made publicly available after the review process.

REFERENCES

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

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

Zheming Cai, Hanning Luo, Tiange Liu, Yingjie Xia, Luming Zhang, Pan Li, and Lei Wang. Emmdet: Energy-efficient multidrone tiny object detection by memory-enhanced spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2026. doi: 10.1109/TNNLS.2026.3680142.

Zhen Cao, Lammert Kooistra, Wensheng Wang, Leifeng Guo, and João Valente. Real-time object detection based on uav remote sensing: A systematic literature review. *Drones*, 7(10), 2023. ISSN 2504-446X. doi: 10.3390/drones7100620. URL https://www.mdpi.com/2504-446X/7/10/620.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 213–229, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58452-8.

Gerasimos Damigos, Tore Lindgren, Sara Sandberg, and George Nikolakopoulos. Performance of sensor data process offloading on 5g-enabled uavs. *Sensors*, 23(2), 2023. ISSN 1424-8220. doi: 10.3390/s23020864. URL https://www.mdpi.com/1424-8220/23/2/864.

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

Yimeng Fan, Changsong Liu, Mingyang Li, Dongze Liu, Yuting Su, Yanyan Liu, and Wei Zhang. Spikedet: Better firing patterns for accurate and energy-efficient object detection with spiking neural networks, 2026. URL https://arxiv.org/abs/2501.15151.

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

Krishnandu Hazra, Vijay K. Shah, Simone Silvestri, Vaneet Aggarwal, Sajal K. Das, Subrata Nandi, and Sujoy Saha. Designing efficient communication infrastructure in post-disaster situations with limited availability of network resources. *Computer Communications*, 164:54–68, 2020. ISSN 0140-3664. doi: https://doi.org/10.1016/j.comcom.2020.09.019. URL https://www.sciencedirect.com/science/article/pii/S0140366420319393.

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, 2014. doi: 10.1109/ISSCC.2014.6757323.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Advances in neural information processing systems*, 29, 2016.

Yushi Huo, Hongwei Ge, Guozhi Tang, Shengxuan Gao, and Jiale Xu. Spiking trans-yolo: A range-adaptive energy-efficient bridge between yolo and transformer. *Neurocomputing*, 645:130407, 2025. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2025.130407. URL https://www.sciencedirect.com/science/article/pii/S0925231225010793.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8. https://github.com/ultralytics/ultralytics, 2023.

Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements, 2024. URL https://arxiv.org/abs/2410.17725.

Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: Spiking neural network for energy-efficient object detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11270–11277, Apr. 2020. doi: 10.1609/aaai.v34i07.6787. URL https://ojs.aaai.org/index.php/AAAI/article/view/6787.

Yaning Kong, Xiangfeng Shang, and Shijie Jia. Drone-detr: Efficient small object detection for remote sensing image using enhanced rt-detr model. *Sensors*, 24(17), 2024. ISSN 1424-8220. doi: 10.3390/s24175496. URL https://www.mdpi.com/1424-8220/24/17/5496.

Marek Kraft, Mateusz Piechocki, Bartosz Ptak, and Krzysztof Walas. Autonomous, onboard vision-based trash and litter detection in low altitude aerial images collected by an unmanned aerial vehicle. *Remote Sensing*, 13(5), 2021. ISSN 2072-4292. doi: 10.3390/rs13050965. URL https://www.mdpi.com/2072-4292/13/5/965.

Qingpeng Li, Yuxin Zhang, Leyuan Fang, Yuhan Kang, Shutao Li, and Xiao Xiang Zhu. Drebnet: Dual-stream restoration embedding blur-feature fusion network for high-mobility uav object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–18, 2025a. doi: 10.1109/TGRS.2025.3543270.

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

Hai Liao, Song Chen, Yan Liang, and Fan Min. Snn-drone: Low-power drone-view object detection with integer-valued spiking neural networks. *IEEE Internet of Things Journal*, 13(5):9375–9395, 2026a. doi: 10.1109/JIOT.2025.3644346.

Ningsheng Liao, Yuning Zhang, Zhongliang Yu, Jiangshuai Huang, Mi Zhu, and Bo Peng. Uav-detr: Few-parameter detr for small object detection in high-altitude uav images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 19:2575–2587, 2026b. doi: 10.1109/JSTARS.2025.3645731.

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.

Ziye Liu, Chen Chen, Ziqin Huang, Yoong Choon Chang, Lei Liu, and Qingqi Pei. A low-cost and lightweight real-time object-detection method based on uav remote sensing in transportation systems. *Remote Sensing*, 16(19), 2024. ISSN 2072-4292. doi: 10.3390/rs16193712. URL https://www.mdpi.com/2072-4292/16/19/3712.

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 Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 253–272, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73411-3.

Hanyu Ouyang and Jie Jiang. Spiking-detr: A spike-driven end-to-end object detection framework on spike-form data streams using spiking-transformer and spiking residual learning. *Available at SSRN 4706194*, 2024.

D. C. Schedl, I. Kurmi, and O. Bimber. An autonomous drone for search and rescue in forests using airborne optical sectioning. *Science Robotics*, 6(55):eabg1188, 2021. doi: 10.1126/scirobotics.abg1188. URL https://www.science.org/doi/abs/10.1126/scirobotics.abg1188.

Yiyu Shen, Hu Liu, Keke Zha, Xu Liu, and Yanan Ding. Sa-yolo: Spike-driven attention for energy-efficient uav-based small object detection. *IEEE Internet of Things Journal*, 12(21):44851–44866, 2025. doi: 10.1109/JIOT.2025.3596434.

Yaolian Song, Can Wang, Dayan Li, and Xinyi Liu. UAV small target detection algorithm based on improved YOLOv5s. *Journal of Zhejiang University (Engineering Science)*, 58(12):2417–2426, 2024. doi: 10.3785/j.issn.1008-973X.2024.12.001.

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

Shiyi Tang, Shu Zhang, and Yini Fang. Hic-yolov5: Improved yolov5 for small object detection. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6614–6619, 2024. doi: 10.1109/ICRA57147.2024.10610273.

Siqi Wang, Jiang Liu, Sutthiphong Srigrarom, Cheng Xiang, Boo Cheong Khoo, and Baigen Cai. Small object detection for uav using bio-inspired spiking neural network. *Journal of Traffic and Transportation Engineering*, 26(3):261–275, 2026. doi: 10.19818/j.cnki.1671-1637.2026.156.

Chenhongyi Yang, Zehao Huang, and Naiyan Wang. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13668–13677, June 2022.

Man Yao, JiaKui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 64043–64058. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ca0f5358dbadda74b3049711887e9ead-Paper-Conference.pdf.

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

Huaxiang Zhang, Hao Zhang, Kai Liu, Zhongxue Gan, and Guo-Niu Zhu. Uav-detr: Efficient end-to-end object detection for unmanned aerial vehicle imagery. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 15143–15149, 2025. doi: 10.1109/IROS60139.2025.11246176.

Zixiao Zhang, Xiaoqiang Lu, Guojin Cao, Yuting Yang, Licheng Jiao, and Fang Liu. Vit-yolo:transformer-based yolo for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 2799–2808, October 2021.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16965–16974, 2024.

Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7380–7399, 2022. doi: 10.1109/TPAMI.2021.3119563.

Xingkui Zhu, Shuchang Lyu, Xu Wang, and Qi Zhao. Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 2778–2788, October 2021a.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=gZ9hCDWe6ke.

A Additional Experimental Details

A.1 Dataset

A.1.1 VisDrone2019-DET

VisDrone2019-DET (Zhu et al., 2022) is adopted as the primary benchmark for evaluating the proposed model in low-altitude UAV object detection. Collected by the AISKYEYE team, it is the image-based object detection subset of VisDrone2019. The dataset contains 8,629 annotated images, officially divided into 6,471 training, 548 validation, and 1,610 test images, with resolutions ranging from 480×360 to 2000×1500 pixels. The models are trained on the training set and evaluated on the validation set.

The dataset provides approximately 457,000 instances across ten categories: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor. Small objects with areas below 32^2 pixels account for approximately 62.34% of all annotated instances, making small object detection a major challenge.

The images were captured on urban and rural roads across 14 Chinese cities under diverse weather conditions, illumination levels, flight altitudes, viewing angles, and object densities. Frequent occlusion, cluttered backgrounds, dense object distributions, and substantial viewpoint variation further increase the difficulty of detection, making VisDrone2019-DET a challenging benchmark for evaluating small object detection, dense-object detection, and robustness in UAV imagery.

A.1.2 UAVVASTE

UAVVaste (Kraft et al., 2021) is used as an additional benchmark for UAV-based object detection. The released version used in this study contains 772 images with 3,718 manually annotated waste instances in urban and natural environments, including streets, parks, and lawns, for UAV-based rubbish detection. The dataset is officially divided into 540 training images, 116 validation images, and 116 test images. The dataset is publicly available in COCO format with bounding-box annotations and segmentation masks; this study uses the bounding-box annotations for object detection. Following this split, the models are trained on the training set and validated on the validation set, with final results reported on the test set. Complementing VisDrone2019-DET, UAVVaste is used to evaluate the applicability of the proposed model across different UAV-based object detection tasks.

A.2 IMPLEMENTATION DETAILS

All models proposed in this study are trained and evaluated under the same hardware and software configurations. Experiments were conducted on a computing platform equipped with NVIDIA H800 Tensor Core GPUs and running Ubuntu 20.04.6. The software environment comprised Python 3.10.13, PyTorch 2.1.2, and CUDA 11.8.

Training configuration. Training is performed on a single NVIDIA H800 GPU. The ResNet-34 backbone adopted from EMS-YOLO (Su et al., 2023) is initialized with weights pretrained on COCO. The model is trained for 120 epochs with a batch size of 6 using the AdamW optimizer. All parameter groups first undergo a 500-iteration linear warm-up. The backbone learning rate is then fixed at 1×10^{-5}, whereas the main learning rate is cosine-annealed from 1×10^{-4} to 1×10^{-5} over the remaining training period. On VisDrone2019-DET, Mosaic augmentation (Bochkovskiy et al., 2020) is applied with a probability of 0.9 for the first 90 epochs and disabled for the final 30 epochs. On UAVVaste, Mosaic is applied with a probability of 0.9 for the first 110 epochs and disabled for the final 10 epochs. Following SpikeSOD (Wang et al., 2026), the multi-level firing parameters are set to $n = 2$ and $m = 3$. The base firing threshold is initialized as $V_{\mathrm{th}} = 1$ and subsequently adjusted through the learned time–channel scaling factors during training. The main training hyperparameters of SpikeUAV-DETR are summarized in Table 6.

Table 6: Training hyperparameters of SpikeUAV-DETR.

Hyperparameter	Setting
Optimizer	AdamW
Training epochs	120
Batch size	6
Warm-up	500 iterations, linear
Backbone learning rate	1×10^{-5}
Main learning rate	$1 \times 10^{-4} \rightarrow 1 \times 10^{-5}$, cosine annealing
Mosaic probability (VisDrone2019-DET)	0.9 (epochs 1–90), 0 (epochs 91–120)
Mosaic probability (UAVVaste)	0.9 (epochs 1–110), 0 (epochs 111–120)
Multi-level firing	$n = 2,\ m = 3$
Base firing threshold	$V_{\mathrm{th}} = 1$

Inference configuration. For inference speed evaluation, all models are tested at an input resolution of 640×640 with FP32 precision on a single H800 GPU under identical hardware conditions. The software environment follows the officially recommended configurations and maintains compatibility with the H800 GPU to ensure fair and reliable comparisons among different models.

A.3 EVALUATION METRICS

Detection accuracy. We evaluate detection accuracy using average precision (AP) and mean average precision (mAP), reporting AP@50, AP@50:95, mAP@50, and mAP@50:95. AP@50 denotes the AP of a single category at an intersection-over-union (IoU) threshold of 0.50, whereas AP@50:95 averages the class-specific AP over ten IoU thresholds from 0.50 to 0.95 in increments of 0.05. AP@50 and AP@50:95 are first calculated separately for each category in the dataset and then averaged across all categories to obtain mAP@50 and mAP@50:95, respectively.

Scale definitions. We report detection performance separately for small, medium, and large objects, categorized by ground-truth (GT) bounding-box area following the COCO scale definition (Lin et al., 2014). With area A measured in square pixels, the three categories satisfy $A < 32^2$, $32^2 \leq A < 96^2$, and $A \geq 96^2$, respectively. For each scale, we calculate AP@50 and AP@50:95 separately for each object category using the procedure described above. These values are then averaged across all categories to obtain scale-specific mAP@50 and mAP@50:95.

Inference efficiency and computational cost. Inference efficiency is measured in frames per second (FPS). Model size is reported as the number of trainable parameters in millions (M), while computational cost is measured using floating-point operations (FLOPs) for MAC computation and effective synaptic operations (SOPs) for spike-driven computation.

B Theoretical Energy Evaluation

We follow the energy evaluation method commonly used in the SNN field to evaluate the energy consumption of different ANN and SNN models (Su et al., 2023; Luo et al., 2025; Yao et al., 2024).

An ANN consists of modules that process continuous-valued activations using floating-point multiply–accumulate operations (MACs), and its computational cost is measured in FLOPs. Since one MAC contains one multiplication and one addition, it is counted as two FLOPs. For ANN module i, let $OP_{\mathrm{MAC},i}$ denote its MAC count during one inference and $N_{\mathrm{FLOPs},i} = 2OP_{\mathrm{MAC},i}$ denote the corresponding number of FLOPs. The energy consumption of this module is estimated as:

$$E_i^{\mathrm{ANN}} = e_{\mathrm{MAC}} OP_{\mathrm{MAC},i} = 0.5\, e_{\mathrm{MAC}} N_{\mathrm{FLOPs},i}, \tag{8}$$

where e_{MAC} is the energy consumed by one MAC.

An SNN consists of modules that transmit binary spikes and perform spike-triggered accumulate operations (ACs) on synaptic weights on neuromorphic hardware. Inactive synaptic operations can be skipped; therefore, we use the number of accumulations actually triggered by spikes, termed effective SOPs, to measure SNN computational cost. For SNN module j, its effective SOP count and energy over T timesteps are estimated as:

$$N_{\mathrm{SOPs},j} = T f_j OP_{\mathrm{AC},j}, \tag{9}$$

$$E_j^{\mathrm{SNN}} = e_{\mathrm{AC}} N_{\mathrm{SOPs},j}. \tag{10}$$

Here, f_j is the average spike firing rate of module j, $OP_{\mathrm{AC},j}$ is its AC count within one timestep when $f_j = 1$, and e_{AC} is the energy consumed by one AC. For multi-level spiking neurons such as I-LIF and AT-LIMF, each neuron can be regarded as comprising multiple binary firing branches, and f_j is the total firing rate summed over all branches; it can therefore exceed 1. Multiple branches activated by one neuron within the same timestep therefore contribute multiple spike-triggered accumulations. When the SNN and ANN counterparts share the same topology, $OP_{\mathrm{AC},j}$ generally equals the MAC count required by the ANN module for the tensor corresponding to one SNN timestep.

Some SNNs, including the proposed network, employ hybrid modules that contain both MAC computation and spike-driven computation. For hybrid module k, let $OP_{\mathrm{MAC},k}$ denote the number of MACs during one inference, and let $OP_{\mathrm{AC},k}$ denote the number of ACs in its spike-driven computation within one timestep when $f_k = 1$. Its energy is estimated from these two types of computation as:

$$E_k^{\mathrm{hybrid}} = e_{\mathrm{MAC}} OP_{\mathrm{MAC},k} + T e_{\mathrm{AC}} f_k OP_{\mathrm{AC},k}. \tag{11}$$

Equivalently:

$$E_k^{\mathrm{hybrid}} = 0.5\, e_{\mathrm{MAC}} N_{\mathrm{FLOPs},k} + e_{\mathrm{AC}} N_{\mathrm{SOPs},k}. \tag{12}$$

Let L_{ANN}, L_{SNN}, and L_{H} denote the numbers of ANN, SNN, and hybrid modules in a network, respectively. The total network energy is first expressed as the sum of the three module-wise energy terms:

$$E_{\mathrm{model}} = \sum\nolimits_{i=1}^{L_{\mathrm{ANN}}} E_i^{\mathrm{ANN}} + \sum\nolimits_{j=1}^{L_{\mathrm{SNN}}} E_j^{\mathrm{SNN}} + \sum\nolimits_{k=1}^{L_{\mathrm{H}}} E_k^{\mathrm{hybrid}}. \tag{13}$$

Grouping the operation counts by type gives:

$$N_{\mathrm{FLOPs}} = \sum_{i=1}^{L_{\mathrm{ANN}}} N_{\mathrm{FLOPs},i} + \sum_{k=1}^{L_{\mathrm{H}}} N_{\mathrm{FLOPs},k}, \tag{14}$$

$$N_{\mathrm{SOPs}} = \sum_{j=1}^{L_{\mathrm{SNN}}} N_{\mathrm{SOPs},j} + \sum_{k=1}^{L_{\mathrm{H}}} N_{\mathrm{SOPs},k}. \tag{15}$$

where N_{FLOPs} and N_{SOPs} aggregate the FLOPs and effective SOPs, respectively, across the corresponding modules and operation types. Using these two totals, the unified network energy estimate is:

$$E_{\mathrm{model}} = 0.5\, e_{\mathrm{MAC}} N_{\mathrm{FLOPs}} + e_{\mathrm{AC}} N_{\mathrm{SOPs}}. \tag{16}$$

For a pure ANN or SNN, the absent operation type is set to zero; both terms are retained for a hybrid model.

For consistent comparison, all models are assumed to use 45-nm CMOS technology. The energy costs of 32-bit floating-point MACs and ACs are set to $e_{\mathrm{MAC}} = 4.6\,\mathrm{pJ}$ and $e_{\mathrm{AC}} = 0.9\,\mathrm{pJ}$ (Horowitz, 2014). The resulting energy is reported in $\mathrm{mJ/image}$.

C Theoretical Details of the AT-LIMF Neuron

C.1 Neuronal Background and Design Motivation

Spiking neurons convert accumulated membrane potentials into discrete spikes according to their firing dynamics. This conversion governs the preservation and propagation of feature information throughout the network.

Leaky integrate-and-fire neuron. The leaky integrate-and-fire (LIF) neuron is widely used in SNNs because it provides a practical balance between neuronal dynamics and computational complexity (Luo et al., 2025). For neuron i in layer l, LIF first accumulates the weighted outputs of its presynaptic neurons:

$$I_{l,i}(t) = \sum_j w_{l,ij} S_{l-1,j}(t) + b_{l,i}, \tag{17}$$

where $w_{l,ij}$ and $b_{l,i}$ are the synaptic weight and bias, respectively, and $S_{l-1,j}(t)$ is the presynaptic output at timestep t. The input is integrated with the membrane state retained from the preceding timestep, after which the neuron fires and resets according to

$$v_{l,i}(t) = u_{l,i}(t-1) + \lambda^{-1}(I_{l,i}(t) - [u_{l,i}(t-1) - u_{\mathrm{rest}}])\,, \tag{18}$$

$$S_{l,i}(t) = H(v_{l,i}(t) - V_{\mathrm{th}})\,, \tag{19}$$

$$u_{l,i}(t) = v_{l,i}(t)[1 - S_{l,i}(t)] + u_{\mathrm{rest}} S_{l,i}(t). \tag{20}$$

Here, $v_{l,i}(t)$ and $u_{l,i}(t)$ denote the pre-firing and post-reset membrane potentials, respectively; λ controls leakage toward the resting potential u_{rest}; and $H(\cdot)$ is the Heaviside step function. In LIF, the firing process maps every suprathreshold membrane potential to the same output, $S_{l,i}(t) = 1$. Consequently, amplitude differences among these membrane potentials are not preserved, introducing substantial quantization error that degrades object detection performance.

Integer leaky integrate-and-fire neuron. To reduce the quantization error caused by binary LIF firing, the integer leaky integrate-and-fire (I-LIF) neuron replaces the binary training response with a bounded integer-valued response (Luo et al., 2025):

$$z(t) = \mathrm{clip}(\mathrm{round}(v(t)), 0, D)\,, \qquad z(t) \in \{0, 1, \ldots, D\}, \tag{21}$$

I-LIF uses D as the maximum integer output; thus, $z(t)$ contains one zero-response level and D nonzero firing levels, preserving more membrane-potential amplitude information than binary LIF. During inference, I-LIF converts each integer response into binary spikes to retain spike-driven accumulation. However, adjacent integer responses remain separated by a single threshold interval, leaving a relatively coarse quantization granularity that discards membrane-potential differences within each interval (Wang et al., 2026).

Leaky integrate-and-multi-fire neuron. To further reduce the quantization error remaining in I-LIF, SpikeSOD introduces the leaky integrate-and-multi-fire (LIMF) neuron, which combines binary firing places with different weights to provide finer amplitude granularity (Wang et al., 2026). Let $\mathcal{K} = \{-m, -m+1, \ldots, n\}$ denote the firing-place index set. LIMF retains V_{th} as the base threshold and constructs $n + m + 1$ firing places weighted by $\{2^k V_{\mathrm{th}} \mid k \in \mathcal{K}\}$. Its firing process is given by

$$\text{for } \rho_{l,i,n+1}(t) \leftarrow v_{l,i}(t),\ k \leftarrow n;\ k \geq -m;\ k \leftarrow k-1 \text{ do}$$

$$S_{l,i,k}(t) = H\big(\rho_{l,i,k+1}(t) - 2^k V_{\mathrm{th}}\big), \tag{22}$$

$$\rho_{l,i,k}(t) = \rho_{l,i,k+1}(t) - 2^k V_{\mathrm{th}} S_{l,i,k}(t), \tag{23}$$

$$\text{end for}$$

$$r_{l,i}^{\mathrm{LIMF}}(t) = V_{\mathrm{th}} \sum\nolimits_{k \in \mathcal{K}} 2^k S_{l,i,k}(t). \tag{24}$$

Starting from $\rho_{l,i,n+1}(t) = v_{l,i}(t)$, the loop determines each firing bit and updates the residual membrane potential from high to low. The resulting bits form $r_{l,i}^{\mathrm{LIMF}}(t)$ with a resolution of $2^{-m} V_{\mathrm{th}}$, providing a finer representation of membrane-potential amplitudes than I-LIF.

Problem analysis and design motivation. Multi-level firing is designed to represent membrane-potential amplitudes with multiple discrete response levels, thereby reducing the quantization error introduced by binary firing. To achieve this goal, the firing thresholds must allocate the available response levels over the membrane-potential range encountered by the neuron. The lowest firing threshold determines the quantization interval, whereas the complete threshold ladder determines the representable range. The effectiveness of multi-level firing therefore depends on whether the threshold ladder covers the membrane-potential distribution while providing a suitable balance between resolution and range.

However, the original LIMF uses a uniform, non-learnable firing-threshold sequence $\{2^k V_{\mathrm{th}} \mid k \in \mathcal{K}\}$ throughout the network, requiring the same thresholds to accommodate different membrane-potential distributions. If its lowest interval $2^{-m} V_{\mathrm{th}}$ is too large for a weak-response distribution, small but informative responses are mapped to zero or to only a few low codes; if its upper bound $V_{\mathrm{th}}(2^{n+1} - 2^{-m})$ is too small for a strong-response distribution, larger responses are clipped to the same maximum code and their amplitude differences are lost.

To address this limitation, the fixed thresholds can be replaced with learnable thresholds, allowing the firing ladder to be optimized jointly with the network during training. However, parameterizing learnable thresholds requires balancing the degrees of freedom against parameter sharing. Too few degrees of freedom, such as learning only one shared threshold scale per layer, still cannot accommodate different membrane-potential distributions within that layer; too many, such as learning a complete firing ladder independently for every neuron, can readily cause the number of threshold parameters to grow sharply and increase the risk of spatial overfitting. An efficient threshold-ladder adjustment mechanism is therefore needed to balance adaptability, parameter count, and spatial generalization.

Based on the above analysis, we propose AT-LIMF, which adapts the firing ladder across timesteps and feature channels while controlling parameter overhead and preserving spatial generalization. Its design advantages are analyzed in Appendix C.2, while its training and inference details are provided in Appendix C.3.

C.2 Design Advantages of AT-LIMF

Adaptive allocation of response levels. AT-LIMF introduces a set of time–channel scaling parameters into each neuronal layer to adaptively adjust the complete LIMF firing ladder. The timestep-dependent scale follows the temporal evolution of the membrane potential, whereas the channel-dependent scale accommodates persistent magnitude differences among feature channels. Changing this scale jointly adjusts the minimum firing interval and the upper representable bound. A smaller scale improves the resolution of weak responses, whereas a larger scale extends the coding range and reduces saturation. During training, the scale parameters are optimized jointly with the network, allowing the firing ladder to adapt to the network's evolving response characteristics and thereby improving the effective utilization of its response levels.

Parameter efficiency and spatial generalization. Within each neuronal layer, AT-LIMF shares its scaling parameters across all spatial or token positions, thereby controlling the parameter count. For N_{AT} layers with channel widths $\{C_q\}$, it introduces only $T\sum_{q=1}^{N_{\mathrm{AT}}} C_q$ parameters. This sharing also preserves translation equivariance for convolutional features, thereby reducing overfitting to specific spatial locations and safeguarding the network's spatial generalization.

In summary, AT-LIMF preserves the multi-level coding structure of LIMF while using a compact time–channel parameterization to better align the firing ladder with membrane-potential distributions, thereby enhancing the neuron's capacity to retain and transmit information with limited parameter overhead.

C.3 AT-LIMF Training and Inference Details

C.3.1 Training

Before training, the hyperparameters n, m, and λ are specified, and all $a_{t,c}$ are initialized to zero. During the training forward pass, the time–channel scaling parameters first determine the corresponding base threshold and minimum firing interval. AT-LIMF then follows the neuronal dynamics defined in the main text to perform integration, firing, and soft reset, passing the normalized discrete output $s_{t,c}$ to the subsequent network layer. During backpropagation, gradients of the detection loss propagate across network layers and timesteps. Let $q_{t,c} = Q_{\max} s_{t,c}$ denote the integer code before normalization, and let $\xi_{t,c} = v_{t,c}/\Delta_{t,c}$ denote the ratio of the membrane potential to the adaptive minimum firing interval. For the combined clipping and floor operation, a straight-through estimator (STE) (Hubara et al., 2016) is used to pass gradients:

$$\frac{\partial q_{t,c}}{\partial \xi_{t,c}} \approx \begin{cases} 1, & 0 \le \xi_{t,c} \le Q_{\max}, \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

Within the valid coding range, the STE approximates the local derivative of the discrete encoding with respect to $\xi_{t,c}$ as one, allowing the detection-loss gradient to pass through the neuron layer and continue propagating to preceding network layers. Gradients through the reset subtraction are blocked by the stop-gradient operator.

In particular, the time–channel threshold-scaling parameter affects $\xi_{t,c}$ through:

$$\xi_{t,c} = \frac{v_{t,c}}{\Delta_{t,c}} = \frac{v_{t,c}}{2^{-m} V_{\mathrm{th}} \exp(a_{t,c})}. \tag{26}$$

Treating the current membrane potential $v_{t,c}$ as locally constant gives:

$$\frac{\partial \xi_{t,c}}{\partial a_{t,c}} = -\frac{v_{t,c}}{2^{-m} V_{\mathrm{th}} \exp(a_{t,c})} = -\xi_{t,c}. \tag{27}$$

Restoring the batch index b and spatial or token index p, the same $a_{t,c}$ is shared across these dimensions. The resulting approximate gradient of the detection loss with respect to the threshold-scaling parameter is:

$$\frac{\partial \mathcal{L}_{\mathrm{det}}}{\partial a_{t,c}} \approx -\sum_{b,p} \frac{\partial \mathcal{L}_{\mathrm{det}}}{\partial s_{b,t,c,p}} \frac{\xi_{b,t,c,p}}{Q_{\max}} \left(\frac{\partial q_{b,t,c,p}}{\partial \xi_{b,t,c,p}} \right)_{\mathrm{STE}}. \tag{28}$$

Here, $(\partial q_{b,t,c,p}/\partial \xi_{b,t,c,p})_{\mathrm{STE}}$ denotes the rectangular-window STE derivative defined in equation 25. Thus, the threshold-scaling parameter is updated along the path $\mathcal{L}_{\mathrm{det}} \to s_{t,c} \to q_{t,c} \to \xi_{t,c} \to a_{t,c}$, with gradients from all shared positions accumulated into the same parameter. The optimizer then uses the accumulated gradient to update the corresponding threshold-scaling parameter.

C.3.2 Inference

During inference, the learned threshold parameters and synaptic weights remain fixed. The pre-normalization integer code $q_{l-1,j}$ of neuron j in layer $l-1$ can be represented by the following lossless binary decomposition:

$$q_{l-1,j} = \sum_{k \in \mathcal{K}} 2^{k+m} S_{l-1,j,k}, \qquad S_{l-1,j,k} \in \{0,1\}. \tag{29}$$

Dividing equation 29 by $Q_{\max}$ reconstructs the normalized response $s_{l-1,j}$. For neuron i in layer l, let $\mathcal{N}_i$ denote its presynaptic neurons in layer $l-1$, and let $w_{l,ij}$ denote the synaptic weight from neuron j to neuron i. The equivalent weight associated with firing bit k is $w_{l,ij,k} = 2^{k+m} Q_{\max}^{-1} w_{l,ij}$. This construction yields the same scalar synaptic input as the normalized responses used during training:

$$I_{l,i} = \sum_{j \in \mathcal{N}_i} w_{l,ij} s_{l-1,j} + b_{l,i} = \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N}_i} w_{l,ij,k} S_{l-1,j,k} + b_{l,i}. \tag{30}$$

Because each $S_{l-1,j,k}$ is binary, the corresponding MACs can be replaced by spike-triggered sparse ACs, thereby retaining spike-driven inference and the energy advantage of SNNs.

D Visualization Results

Figure 6 compares RT-DETR-R18, the spike-driven baseline, and SpikeUAV-DETR ($T = 3$) across four representative UAV-based object detection scenarios. From left to right, each row shows the ground truth followed by predictions from RT-DETR-R18, the spike-driven baseline, and SpikeUAV-DETR. Blue boxes mark ground truth, green boxes true positives, and red boxes false positives; both confidence and IoU thresholds are set to 0.5.



Figure 6: Visualization results

Panels (1-1)–(1-4) show a daytime aerial scene containing objects at different viewing distances. The purple rectangle marks nearby occluded objects. SpikeUAV-DETR detects all of these objects, whereas the comparison models produce missed or false detections. The yellow rectangle marks distant small objects. SpikeUAV-DETR detects more of these objects than the comparison models.

Panels (2-1)–(2-4) show a medium-to-close-range daytime aerial scene containing numerous overlapping objects, with the overlapping groups enclosed by yellow rectangles. SpikeUAV-DETR correctly detects all objects within these rectangles and better distinguishes individual objects within the overlapping groups, whereas the comparison models produce several missed or false detections.

Panels (3-1)–(3-4) show an aerial scene containing densely clustered small objects. SpikeUAV-DETR produces fewer detection errors and more accurately localized boxes than the comparison models.

Panels (4-1)–(4-4) show a low-light nighttime aerial scene. SpikeUAV-DETR produces fewer false positives and more stable predictions than the comparison models.

Overall, these qualitative comparisons show that SpikeUAV-DETR produces fewer missed and false detections than the two comparison models under scale variation, occlusion, dense object distributions, and low illumination. These results demonstrate the model's detection accuracy and robustness across challenging UAV-based detection scenarios.

E NMS Sensitivity and End-to-End Prediction

Most real-time SNN detectors are developed from YOLO-based architectures and therefore retain non-maximum suppression (NMS). NMS introduces confidence and IoU thresholds whose settings can affect post-processing latency and detection results, requiring complex tuning for different scenes (Zhao et al., 2024). Using EMS-YOLO (Su et al., 2023) on VisDrone2019-DET, we evaluated the effects of NMS thresholds on precision, recall, and NMS latency. Figure 7(a) sweeps the IoU threshold with the confidence threshold fixed at 0.001, whereas Figure 7(b) sweeps the confidence threshold with the NMS IoU threshold fixed at 0.7.

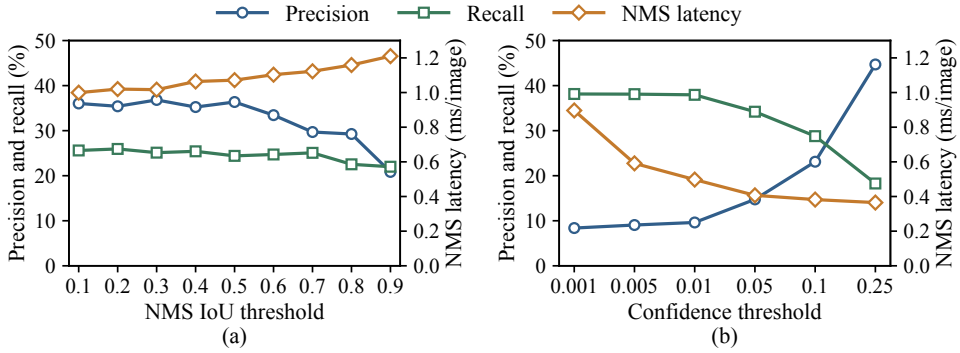

Figure 7: NMS threshold sensitivity of EMS-YOLO on VisDrone2019-DET.

The latency curves show that varying the confidence and IoU thresholds affects NMS latency, causing inference speed to fluctuate across hyperparameter settings.

The accuracy curves further show how the two thresholds affect the Precision–Recall balance. The confidence threshold determines which candidates enter NMS according to their prediction scores; increasing it removes more low-score predictions, producing higher Precision and lower Recall in Figure 7(b) and allowing relatively straightforward adjustment according to the required operating preference. The IoU threshold determines whether overlapping predictions are duplicates. The resulting suppression behavior depends on object density, occlusion, scale, and prediction overlap. Too low a value may suppress valid predictions for adjacent or mutually occluded objects, whereas too high a value may retain duplicates. Consistent with this mechanism, Figure 7(a) shows coupled, non-monotonic changes in Precision and Recall as the IoU threshold varies. Consequently, selecting an appropriate IoU threshold requires more extensive scene-specific tuning. The confidence and IoU thresholds jointly affect Precision and Recall, and their coupled effects make the optimal parameter combination difficult to determine across different scenes, further increasing the complexity of NMS tuning.

By contrast, SpikeUAV-DETR inherits DETR's one-to-one set prediction based on bipartite matching (Carion et al., 2020; Zhao et al., 2024). During training, each ground-truth object is matched to a single prediction, encouraging the detector to produce a prediction set without duplicate detections. At inference, the detector predicts the one-to-one object set, thereby eliminating the need for NMS. This allows SpikeUAV-DETR to adapt to different scenarios and deployment requirements through a simple adjustment of a single threshold, indicating greater scene adaptability than YOLO-based SNN detectors.