SBMVTRACK: SPIKE-BUDGETED MULTI-VIEW LEARNING FOR POWER-EFFICIENT UAV TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

With sparse and event-driven computation, spiking neural networks show great potential for achieving accurate and power-efficient UAV visual tracking. However, existing SNN-based trackers typically use spike firing rates only for power consumption and lack explicit optimization of actual spike activity. Moreover, regulating spike activity alone does not explicitly encourage stable target representations under partial observations and temporal appearance changes. We propose SBMVTrack, a fully spiking tracking framework that combines spike activity regulation with complementary multi-view representation learning. Specifically, SBMVTrack introduces Energy-Weighted Spike Budgeting (EWSB), which incorporates layer-wise computational costs when regulating spike firing rates and penalizing saturated activations, thereby reducing redundant spike computation. To further improve target representations under the spike budget constraint, we introduce Masked Multi-View Target Modeling (MVTM), which treats the initial template, online template, and search region as temporal views of the same target. By aligning target embeddings between masked and corresponding unmasked views and enforcing cross-view identity consistency, MVTM encourages robustness to missing local cues and temporal appearance changes. Experiments on four UAV benchmarks demonstrate competitive tracking performance with a 24.1% reduction in estimated power consumption relative to the baseline. On VisDrone2018, SBMVTrack achieves a success rate of 70.0%, exceeding SpikeTrack by 9.7 percentage points while reducing estimated power consumption by 45.7%. The source code will be released upon acceptance.

1 INTRODUCTION

Unmanned aerial vehicle (UAV) visual tracking has been widely applied in intelligent surveillance, emergency rescue, autonomous navigation, and low-altitude inspection. However, rapid motion, viewpoint variation, and background interference continue to pose substantial challenges. Meanwhile, limited onboard computational resources and battery capacity require trackers to achieve a favorable balance among accuracy, real-time performance, and power efficiency. In recent years, trackers based on convolutional neural networks and Vision Transformers have significantly improved tracking performance Xu et al. (2020); Ye et al. (2022b); Li et al. (2023) . Nevertheless, their reliance on dense multiply-accumulate operations still limits long-term deployment on resource-constrained UAV platforms.

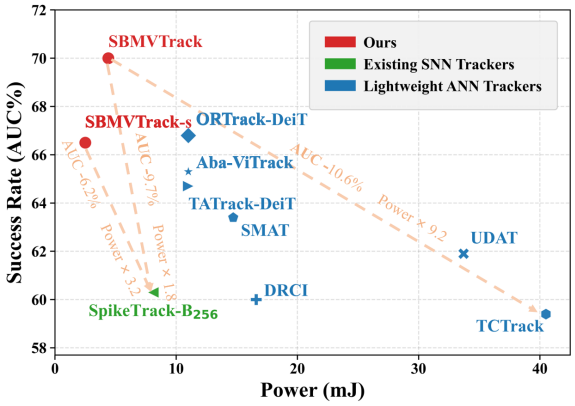

Figure 1: Power–accuracy trade-off on VisDrone2018. SBMVTrack achieves 70.0% AUC at 4.4 mJ estimated power consumption.

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.

Spiking neural networks (SNNs) Maass (1997) transmit information through sparse spikes and perform computation in an event-driven manner, offering a promising alternative for building low-power UAV trackers. Existing SNN-based tracking studies mainly follow two directions: RGB image tracking Zhang et al. (2026) and Event-based tracking Zhang et al. (2025); Yang et al. (2025). Event-based methods leverage the high temporal resolution of event streams to enhance dynamic target modeling, yet their deployment is constrained by the cost of dedicated event cameras. Early RGB-based methods converted Siamese tracking frameworks into spiking form, but did not fully exploit the temporal dynamics of SNNs Xiang et al. (2024). More recently, SpikeTrack Zhang et al. (2026) achieved efficient RGB-based spiking tracking through asymmetric time-step design, unidirectional information propagation, and a memory retrieval mechanism. Although these studies have demonstrated the power-efficiency potential of SNNs for visual tracking, their main focus remains on network architecture and feature interaction design. Spike firing rates are typically used only for post-training power estimation or theoretical analysis, rather than being explicitly incorporated into tracking optimization. This raises a key question: can spike activity be explicitly regulated to reduce power consumption without introducing additional inference overhead, while still maintaining competitive tracking performance?

To address this challenge, we propose SBMVTrack, a fully spiking framework for power-efficient UAV tracking. Our framework comprises two key components: **(i) Energy-Weighted Spike Budgeting (EWSB).** We introduce EWSB to explicitly regulate spike activity during training. Since spike activity in different layers incurs different computational costs, EWSB weights layer-wise activity accordingly and jointly constrains the energy-weighted firing rate and saturation activity. This design discourages excessive spike computation, particularly in computationally expensive layers, while regulating overall activity around a predefined budget. **(ii) Masked Multi-View Target Modeling (MVTM).** Spike budgeting regulates computational activity but does not explicitly encourage representation stability under missing visual evidence. Inspired by masked representation learning in MAE (He et al., 2022) and occlusion-robust tracking in ORTrack (Wu et al., 2025), we introduce MVTM to complement spike regulation with target representation learning. MVTM treats the initial template, online template, and search region as temporal views of the same target and applies masking to construct partial observations. Masked-to-clean target embedding alignment encourages stability when local appearance cues are unavailable, while cross-view identity consistency encourages stable target identity across temporal appearance changes. Both EWSB and MVTM are used only during training, leaving the inference pipeline unchanged and introducing no additional inference overhead.

Extensive experiments on four UAV benchmarks demonstrate a favorable power–accuracy trade-off, with a 24.1% reduction in estimated power consumption relative to the baseline. As shown in Fig. 1, SBMVTrack achieves the highest success rate of 70.0% among the compared trackers on VisDrone2018 at an estimated power consumption of 4.4 mJ. Compared with SpikeTrack, it improves the success rate by 9.7 percentage points while reducing estimated power consumption by 45.7%.

2 Related work

Efficient UAV Visual Tracking. Early UAV visual tracking methods were primarily based on discriminative correlation filters, which enabled efficient target localization through frequency-domain computation Henriques et al. (2014); Danelljan et al. (2016). However, their reliance on handcrafted features limited their representation capability, often leading to performance degradation under complex backgrounds and substantial appearance variations. Subsequently, CNN-based Siamese trackers significantly improved tracking accuracy and robustness by combining deep feature extraction with efficient template matching Xu et al. (2020); Li et al. (2019); Zeng et al. (2023). More recently, Vision Transformers have further advanced UAV tracking by modeling global dependencies between the template and search region through self-attention. To facilitate deployment on resource-constrained platforms, existing studies have introduced lightweight backbones Li et al. (2023), dynamic token computation Li et al. (2024b), knowledge distillation Wu et al. (2025), and asymmetric feature Zhu et al. (2025) interaction to reduce model complexity and inference cost. Although these methods improve the balance between tracking accuracy and computational efficiency, their inference processes still largely rely on dense multiply–accumulate operations. In this work, we aim to explore spike-driven UAV RGB tracking to achieve power-efficient visual processing.

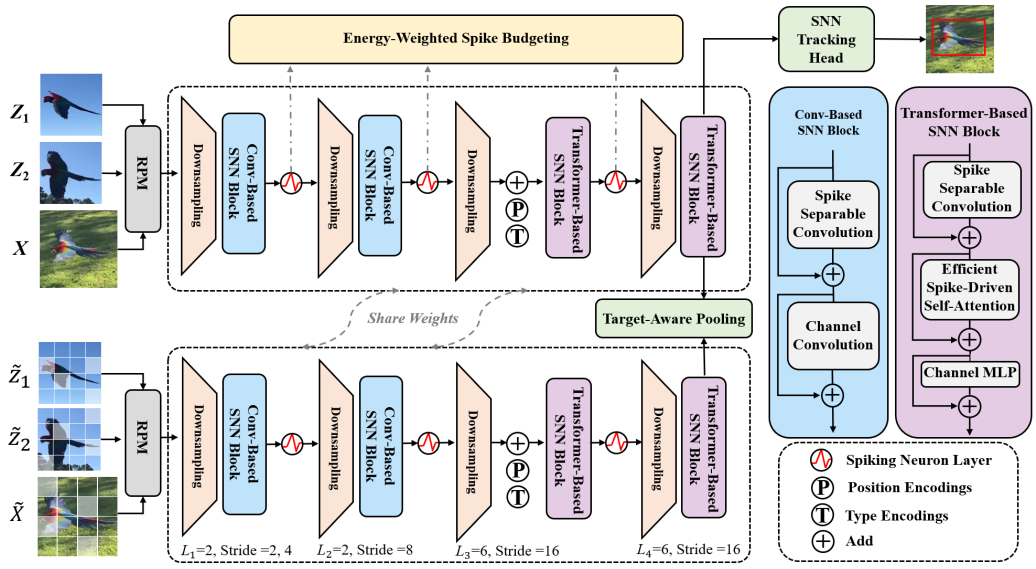

Figure 2: Overview of the SBMVTrack architecture. The network comprises an original branch and a masked branch, both processed by a weight-shared spiking backbone. EWSB regulates energy-weighted spike activity, while MVTM performs masked feature reconstruction and cross-view identity-consistency learning through Target-Aware Pooling. During inference, only the original branch, spiking backbone, and SNN tracking head are retained.

Spiking Neural Networks for Visual Tracking. SNN-based visual tracking methods mainly focus on event-camera inputs Yang et al. (2025); Zhang et al. (2025), leveraging the sparsity and high temporal resolution of event data to achieve promising performance. However, their reliance on dedicated hardware limits practical deployment. For more accessible RGB scenarios, early approaches Luo et al. (2021); Xiang et al. (2024) still relied on specific ANN frameworks for spike-based conversion, limiting the advantages of fully spiking computation. Recently, SpikeTrack Zhang et al. (2026) further improved the accuracy and efficiency of RGB spiking tracking through asymmetric time steps and unidirectional information flow. For UAV tracking scenarios, STATrack Zhong et al. (2026) enhanced target representation learning by introducing mutual information maximization. However, these methods mainly focus on improving representation learning or network architecture, while spike firing rates are still used only for post-hoc power estimation rather than explicitly optimized during training. To address this, we introduce an energy-weighted spike budgeting mechanism from the perspective of spike activity regulation. By constraining the energy-weighted firing rate and saturation activity, it reduces redundant spike computation and lowers power consumption.

Multi-View Target Representation Learning. Multi-view representation learning aims to exploit the consistency and complementarity among different observations of the same target to learn robust feature representations Wang et al. (2025). Recently, masked modeling has been successfully extended to multi-view tasks Weinzaepfel et al. (2022); Nordström et al. (2026). In visual tracking, the initial template, online template, and search region naturally constitute multi-view observations of the target along the temporal dimension. However, existing tracking methods typically treat these observations as independent matching inputs and overlook their potential cross-view consistency. Inspired by multi-view learning, we regard different observations during tracking as temporal views of the target and propose Masked Multi-View Target Modeling to explicitly learn cross-view consistent target representations, improving tracking robustness in challenging scenarios.

3 METHOD

This section details the proposed SBMVTrack. We first describe the spiking neuron model, followed by the network architecture. We then introduce EWSB and MVTM. Finally, we present the training objective and inference procedure.

3.1 SPIKING NEURON MODEL

We adopt the normalized integer leaky integrate-and-fire (NI-LIF) neuron Lei et al. (2025) to construct the fully spiking tracking network. NI-LIF employs normalized integer-valued activations during training and represents them as equivalent spike sequences during inference, thereby facilitating gradient-based optimization while preserving spike-driven computation. Its neuronal dynamics are formulated as:

$$U[t] = H[t-1] + X[t], \tag{1}$$

$$S[t] = \mathrm{Clip}(\mathrm{round}(U[t]),\, 0,\, D)\,/D, \tag{2}$$

$$H[t] = \beta\,(U[t] - S[t] \times D)\,, \tag{3}$$

where $X[t]$ denotes the spatial input current at timestep t, and $U[t]$ is the membrane potential that integrates the temporal information $H[t-1]$ from the previous timestep with the current spatial input $X[t]$. $S[t]$ denotes the normalized integer-valued output within the range $[0, 1]$, and $\mathrm{round}(\cdot)$ represents the rounding operation. $\mathrm{Clip}(x, \min, \max)$ restricts x to the interval $[\min, \max]$, while D is a hyperparameter that determines the maximum integer activation value. The leakage factor β controls the retention of the residual membrane potential across timesteps.

3.2 SBMVTRACK ARCHITECTURE

Fully spiking dual-branch backbone. As illustrated in Fig. 2, SBMVTrack consists of an original tracking branch and a masked auxiliary branch during training, both of which share the same spiking backbone (E-SpikeFormer) Yao et al. (2025) and model parameters. The original tracking branch takes an initial template (Z_1), an online template (Z_2), and a search region (X) as inputs, while the masked branch receives the corresponding masked views ($\widetilde{Z}_1$), ($\widetilde{Z}_2$), and ($\widetilde{X}$). Each input is first processed by the Randomized Patchwork Module (RPM) Yang et al. (2025) to alleviate padding effects introduced during convolutional feature extraction, and is then fed into the shared spiking backbone.

The backbone is composed of Convolution-based SNN blocks and Transformer-based SNN blocks. After RPM processing, the inputs are encoded by the convolutional SNN module, augmented with positional and type embeddings, and subsequently passed to the Transformer-based SNN module. During training, EWSB collects spike activity from different stages of the backbone and imposes an energy-aware budget constraint to regulate redundant spike computation. Finally, the search features from the original tracking branch are fed into a spiking prediction head to estimate the target center, local offset, and bounding-box size. The masked branch is used only during training to construct auxiliary target representations for MVTM.

Conv-based SNN block. Given the input feature $\mathbf{U}$, the convolution-based SNN block performs spatial token mixing and channel-wise interaction through residual spike-driven convolutions:

$$\mathbf{U}' = \mathbf{U} + \mathrm{SSConv}(\mathbf{U}), \tag{4}$$

$$\mathbf{U}'' = \mathbf{U}' + \mathrm{ChConv}(\mathbf{U}'). \tag{5}$$

Here, $\mathrm{SSConv}(\cdot)$ denotes the spike-based separable convolution module for local spatial modeling, while $\mathrm{ChConv}(\cdot)$ performs channel-wise feature transformation. They are formulated as:

$$\mathrm{SSConv}(\mathbf{U}) = \mathrm{Conv}_{\mathrm{pw}}\big(\mathcal{SN}\big(\mathrm{Conv}_{\mathrm{dw}}\big(\mathcal{SN}\big(\mathrm{Conv}_{\mathrm{pw}}(\mathcal{SN}(\mathbf{U}))\big)\big)\big)\big), \tag{6}$$

$$\mathrm{ChConv}(\mathbf{U}') = \mathrm{Conv}\big(\mathcal{SN}\big(\mathrm{Conv}(\mathcal{SN}(\mathbf{U}'))\big)\big), \tag{7}$$

where $\mathcal{SN}(\cdot)$ denotes the spiking neuron layer, and $\mathrm{Conv}_{\mathrm{pw}}(\cdot)$ and $\mathrm{Conv}_{\mathrm{dw}}(\cdot)$ denote pointwise and depthwise convolutions, respectively.

Transformer-based SNN block. Each Transformer-based SNN block consists of a spike-based separable convolution, a spike-driven self-attention module (SDSA), and a channel MLP:

$$\mathbf{U}' = \mathbf{U} + \mathrm{SSConv}(\mathbf{U}), \tag{8}$$

$$\mathbf{U}'' = \mathbf{U}' + \mathrm{SDSA}(\mathbf{U}'), \tag{9}$$

$$\mathbf{U}''' = \mathbf{U}'' + \mathrm{ChMLP}(\mathbf{U}''), \tag{10}$$

$$\mathrm{ChMLP}(\mathbf{U}'') = \mathrm{Linear}\big(\mathcal{SN}\big(\mathrm{Linear}(\mathcal{SN}(\mathbf{U}''))\big)\big). \tag{11}$$

The SDSA module models global interactions among the initial template, online template, and search region in the spiking feature space, while the channel MLP performs channel-wise feature transformation.

SNN tracking head. We employ a lightweight spiking center-based head to estimate the target bounding box (Ye et al., 2022a). After modeling by the spiking backbone, the features corresponding to the search region are selected and fed into three parallel prediction branches. Each branch consists of several Conv-BN-NILIF layers, while the output layer excludes BN and NI-LIF. The three branches predict the target center score map, local offset, and normalized bounding-box width and height, respectively.

3.3 Energy-Weighted Spike Budgeting (EWSB)

Existing SNN-based trackers typically use spike firing rates only for power estimation, without explicitly regulating actual spike activity during training. Moreover, spiking layers incur different computational costs, and their spike activities therefore contribute unequally to the overall power consumption. To address this issue, we propose Energy-Weighted Spike Budgeting (EWSB), which weights layer-wise spike activity according to computational cost and constrains both the energy-weighted firing rate and saturated activity to reduce redundant spike computation. Specifically, let $\mathbf{S}_l \in \mathbb{R}^{B\times T\times C_l\times H_l\times W_l}$ denote the spike tensor of the l-th spiking layer, where B, T, C_l, H_l, and W_l denote the batch size, timesteps, channels, height, and width, respectively. Its average firing rate r_l, normalized energy weight ω_l, and the resulting global energy-weighted firing rate r_{ew} are defined as

$$\begin{aligned} r_l &= \frac{1}{BTC_lH_lW_l}\sum_{b,t,c,h,w}\mathbf{S}_l^{b,t,c,h,w}, \\ \omega_l &= \frac{C_l^{\mathrm{comp}}}{\sum_{j=1}^{L}C_j^{\mathrm{comp}}}, \qquad r_{\mathrm{ew}} = \sum_{l=1}^{L}\omega_l r_l, \end{aligned} \tag{12}$$

where C_l^{comp} denotes the computational cost of the l-th layer, and L is the number of spiking layers involved in EWSB. By assigning larger weights to computationally expensive layers, r_{ew} provides a cost-aware measurement of layer-wise spike activity. We introduce a target spike budget r_{tar} and employ a squared penalty to drive the energy-weighted firing rate toward this predefined budget:

$$\mathcal{L}_{\mathrm{budget}} = \left(r_{\mathrm{ew}} - r_{\mathrm{tar}}\right)^2. \tag{13}$$

Rather than monotonically minimizing spike activity, $\mathcal{L}_{\mathrm{budget}}$ regulates the global energy-weighted firing rate around a predefined budget, thereby suppressing excessive activity while preventing over-sparsification that may impair target representation. Nevertheless, some redundant responses may still exhibit saturated firing. We introduce a saturation activity penalty:

$$\mathcal{L}_{\mathrm{sat}} = \sum_{l=1}^{L}\frac{\omega_l}{N_l}\sum_{i=1}^{N_l}\sigma\left[\alpha\left(S_{l,i} - \tau_{\mathrm{sat}}\right)\right], \tag{14}$$

where N_l is the total number of spike elements in the l-th layer, $\sigma(\cdot)$ denotes the sigmoid function, τ_{sat} is the saturation threshold, and α controls the sharpness of the penalty.

Finally, the EWSB loss is formulated as

$$\mathcal{L}_{\mathrm{EWSB}} = \lambda_{\mathrm{budget}}\mathcal{L}_{\mathrm{budget}} + \lambda_{\mathrm{sat}}\mathcal{L}_{\mathrm{sat}}, \tag{15}$$

where $\lambda_{\mathrm{budget}}$ and λ_{sat} control the strengths of the budget constraint and saturation penalty, respectively.

3.4 Masked Multi-View Target Modeling

The initial template (Z_1), online template (Z_2), and search region (X) describe the target's initial identity, recent state, and current observation, respectively, and can therefore be regarded as three correlated temporal views of the same target. To improve robustness against local information loss and appearance variations, we propose Masked Multi-View Target Modeling (MVTM). Specifically, masks sampled from a Cox process (Wu et al., 2025) are applied to Z_1, Z_2, and X to generate the

masked views $\widetilde{Z}_1$, $\widetilde{Z}_2$, and $\widetilde{X}$. The original and masked views are then processed by a weight-shared spiking backbone. To reduce interference from background regions during multi-view learning, we further introduce Target-Aware Pooling (TAP). TAP generates a Gaussian soft weight map according to the target bounding box and performs weighted aggregation over the spatial features around the target:

$$\mathbf{h}_i = \frac{\sum_{p=1}^{N_i} A_{i,p}\, \mathbf{F}_{i,p}}{\sum_{p=1}^{N_i} A_{i,p}}, \tag{16}$$

where $\mathbf{F}_i$ denotes the feature representation of the i-th view, $A_{i,p}$ is the target-aware weight at spatial position p, and N_i is the number of spatial locations. The pooled features are then passed through a lightweight projection head composed of Linear-ReLU-Linear, obtaining the original-view embedding $\mathbf{q}_i$ and masked-view embedding $\widetilde{\mathbf{q}}_i$, where $i \in \{1, 2, x\}$. MVTM employs a masked feature reconstruction loss to align each masked view with its corresponding complete view:

$$\mathcal{L}_{\text{rec}} = \frac{1}{3} \sum_{i \in \{1,2,x\}} \left\| \widehat{\widetilde{\mathbf{q}}}_i - \widehat{\mathbf{q}}_i \right\|_2^2, \tag{17}$$

where the $\widehat{\cdot}$ denotes L_2 normalization. Meanwhile, the initial template Z_1 is treated as the identity anchor, and the online template and search region are constrained to preserve identity consistency with it:

$$\mathcal{L}_{\text{con}} = \frac{1}{2} \left(\|\widehat{\mathbf{q}}_2 - \widehat{\mathbf{q}}_1\|_2^2 + \|\widehat{\mathbf{q}}_x - \widehat{\mathbf{q}}_1\|_2^2 \right). \tag{18}$$

The overall MVTM objective is defined as

$$\mathcal{L}_{\text{MVTM}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}, \tag{19}$$

where λ_{con} balances masked feature reconstruction and cross-view identity consistency.

3.5 Training Objective

Training. The overall tracking loss is a weighted sum of a focal classification loss Law & Deng (2018), an ℓ_1 regression loss and a generalized IoU loss Rezatofighi et al. (2019):

$$\mathcal{L}_{\text{track}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{GIoU}} \mathcal{L}_{\text{GIoU}} + \lambda_{\text{L1}} \mathcal{L}_{\text{L1}}, \tag{20}$$

where $\lambda_{\text{GIoU}} = 2$ and $\lambda_{\text{L1}} = 5$. The overall training objective is formulated as :

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{track}} + \lambda_{\text{EWSB}} \mathcal{L}_{\text{EWSB}} + \lambda_{\text{MVTM}} \mathcal{L}_{\text{MVTM}}. \tag{21}$$

where $\lambda_{\text{EWSB}} = 1$ and $\lambda_{\text{MVTM}} = 1$ control the contributions of the energy-weighted spike budgeting and masked multi-view target modeling, respectively.

4 Experiments

4.1 Implementation Details

Model. We construct two SBMVTrack variants with different backbone scales to evaluate the trade-off between tracking performance and power efficiency, namely SBMVTrack and SBMVTrack-S, which employ E-SpikeFormer-10M and E-SpikeFormer-5.1M backbones Yao et al. (2025), respectively. The template and search resolutions are set to 128×128 and 256×256, respectively. All NI-LIF neurons operate with a single timestep ($T = 1$), and the spike quantization level is set to $D = 4$. All experiments run on a PC with an i9-10850K processor, 16GB RAM, and an NVIDIA TitanX GPU.

Training strategy. We train SBMVTrack on a combination of LaSOT Fan et al. (2021), GOT-10k Huang et al. (2019a), COCO Lin et al. (2014), and TrackingNet Muller et al. (2018). The batch size is set to 16, and AdamW Loshchilov & Hutter (2017) is adopted with a weight decay of 1×10^{-4}. The initial learning rates of the prediction head and backbone are set to 6×10^{-4} and 6×10^{-5}, respectively. The model is trained for 300 epochs with 60,000 sample pairs per epoch, using a linear warm-up for the first 5 epochs and reducing the learning rate to 0.1× at the 240th epoch.

Table 1: Comparison of SBMVTrack with representative trackers in terms of Precision (Prec.), Success rate (Succ.), Power (mJ), and Parameters (Params.) on UAVTrack112, UAVDT, VisDrone2018, and UAV123. The best and second-best results for each metric are highlighted in bold and underlined, respectively. The compared trackers are categorized into DCF-based, CNN-based, ViT-based, and SNN-based methods.

Tracker	UAVTrack112		UAVDT		VisDrone2018		UAV123		Power	Param.
	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	**(mJ)**	**(M)**
fDSST Danelljan et al. (2016)	56.8	39.1	66.6	38.3	69.8	51.0	58.3	40.5	-	-
MCCT_H Wang et al. (2018)	63.4	43.6	66.8	40.2	80.3	56.7	65.9	45.7	-	-
ARCF Huang et al. (2019b)	67.3	45.6	72.0	45.8	79.7	58.4	67.1	46.8	-	-
AutoTrack Li et al. (2020)	69.4	46.5	71.8	45.0	78.8	57.3	68.9	47.2	-	-
HiFT Cao et al. (2021)	74.2	57.0	65.2	47.5	71.9	52.6	78.7	59.0	33.1	9.9
TCTrack Cao et al. (2022)	76.6	59.4	72.5	53.0	79.9	59.4	80.0	60.5	40.5	9.7
AsymTrack Zhu et al. (2025)	81.3	66.8	84.0	59.0	83.4	60.0	76.7	59.7	8.3	3.4
Aba-ViTrack Li et al. (2023)	82.6	67.6	83.4	59.9	86.1	65.3	86.4	66.4	11.0	8.0
SMAT Gopal & Amer (2024)	82.2	65.3	80.8	58.7	82.5	63.4	81.8	64.6	14.7	8.6
TATrack Li et al. (2024a)	81.5	65.8	83.4	60.6	85.2	64.7	82.7	65.4	11.0	5.6
AVTrack Li et al. (2024b)	80.3	65.4	82.1	58.7	86.0	65.3	84.8	66.8	4.5-8.8	3.5-7.9
ORTrack Wu et al. (2025)	83.2	67.0	83.4	60.1	88.6	66.8	84.3	66.4	11.0	5.8
SGLATrack Xue et al. (2025)	82.8	67.5	81.9	59.9	80.0	61.3	84.9	66.9	7.7	7.9
UETrack Kang et al. (2026)	83.0	69.2	79.6	60.4	80.7	62.2	86.6	67.1	8.5	6.4
SpikeTrack Zhang et al. (2026)	**85.3**	69.2	75.9	56.3	80.2	60.3	**88.7**	67.5	8.1	36.8
SBMVTrack	82.8	**70.1**	**84.3**	**64.8**	**90.0**	**70.0**	87.6	**68.0**	4.4	12.6
SBMVTrack-S	83.8	**70.1**	83.1	63.8	85.3	66.5	85.6	66.6	**2.5**	7.4

Hyperparameter settings. For EWSB, the target energy-weighted firing rate is set to $r_{\text{tar}} = 0.12$, and the spike-budget loss weight is set to $\lambda_{\text{budget}} = 1.0$. The saturation threshold and saturation penalty weight are set to $\tau_{\text{sat}} = 0.9$ and $\lambda_{\text{sat}} = 0.5$, respectively. The sharpness coefficient is set to $\alpha = 10.0$. For MVTM, the weight of the cross-view identity-consistency loss is set to $\lambda_{\text{con}} = 0.5$. Unless otherwise specified, these settings are used in all experiments.

Inference settings. During inference, only the original tracking branch, spiking backbone, and SNN tracking head are retained, while the masked branch, TAP, and projection head used during training are removed. The initial template remains fixed throughout tracking. The online template is updated every 25 frames, but only when the prediction confidence exceeds 0.7 is it replaced by the current target region. Following prior SNN studies Zhang et al. (2026); Yao et al. (2025), we report the average power consumption (mJ) on UAVTrack112 Fu et al. (2021), UAVDT Du et al. (2018), VisDrone2018 Wen et al. (2018), and UAV123 Mueller et al. (2016), calculation details are provided in the supplementary material.

4.2 Comparison with State-of-the-Art Trackers

Tab. 1 compares SBMVTrack with 15 representative trackers on four UAV benchmarks, including traditional DCF-based, CNN-based, and recent ViT-based methods, as well as the SNN-based tracker SpikeTrack. SBMVTrack consistently ranks among the top two in precision (Prec.), success rate (Succ.), and power efficiency (Power). Specifically, SBMVTrack attains a Succ. of 70.1% on UAVTrack112, outperforming all compared methods, and achieves 84.3% Prec. and 64.8% Succ. on UAVDT. On VisDrone2018, it achieves the highest Prec. and Succ. of 90.0% and 70.0%, respectively. On UAV123, it obtains a competitive Prec. of 87.6% and the highest Succ. of 68.0%. Compared with the recent ANN-based lightweight trackers such as UETrack, AVTrack, and ORTrack, SBMVTrack achieves superior or comparable accuracy with significantly lower power consumption. For instance, SBMVTrack consumes only 4.4 mJ, which is 48.2% less than UETrack (8.5 mJ) and 60.0% less than ORTrack (11.0 mJ), while outperforming them on most benchmarks. Compared with SpikeTrack, SBMVTrack reduces estimated power consumption from 8.1 mJ to 4.4 mJ, a reduction of 45.7%, while improving the success rate on VisDrone2018 from 60.3% to

Table 2: EWSB and MVTM ablation on UAV123 and VisDrone2018.

EWSB	MVTM	UAV123		VisDrone2018		Power
		Prec.	Succ.	Prec.	Succ.	(mJ)
		86.1	67.0	82.5	65.0	5.8
✓		87.1	67.9	84.2	65.8	**4.4**
	✓	87.4	**68.0**	84.8	66.7	5.8
✓	✓	**87.6**	**68.0**	**90.0**	**70.0**	**4.4**

Table 3: Reconstruction and consistency ablation on UAV123 and VisDrone2018.

Recon.	Cons.	UAV123		VisDrone2018	
		Prec.	Succ.	Prec.	Succ.
		86.1	67.0	82.5	65.0
✓		84.9	66.4	80.8	63.3
	✓	85.3	66.6	84.4	65.5
✓	✓	**87.4**	**68.0**	**84.8**	**66.7**

Table 4: Ablation study of the key designs in EWSB on UAV123.

Method	Prec.	Succ.	Power (mJ)
Baseline	86.1	67.0	5.8
Unweighted Budget	85.5	66.9	4.8
Energy-Weighted Budget	86.5	67.4	5.1
EWSB	**87.1**	**67.9**	**4.4**

Table 5: Impact of different values of λ_{con}.

λ_{con}	UAV123		VisDrone2018		UAVDT	
	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
0.1	87.3	67.8	83.7	65.9	77.3	60.4
0.5	**87.4**	**68.0**	**84.8**	**66.7**	79.7	61.6
1.0	86.9	67.3	84.6	66.3	**80.9**	**62.8**
1.5	86.9	67.3	82.7	65.1	78.2	60.5

70.0%, a gain of 9.7 percentage points. SBMVTrack-S further reduces the power consumption to 2.5 mJ while maintaining competitive tracking accuracy. These results demonstrate that SBMVTrack achieves a more favorable trade-off between tracking results and power efficiency compared to both ANN-based and existing SNN-based lightweight trackers.

4.3 Ablation Study

Effectiveness of EWSB and MVTM. Table 2 evaluates the contributions of EWSB and MVTM. EWSB reduces the power consumption from 5.8 mJ to 4.4 mJ while improving tracking performance. MVTM further enhances target representation learning without additional inference cost. Their combination achieves the best overall performance on both benchmarks with a 24.1% power reduction, demonstrating their complementary effects.

Analysis of MVTM. Tab. 3 evaluates the two learning objectives in MVTM. Using either reconstruction or consistency alone does not consistently improve tracking performance. In contrast, combining the two objectives achieves clear gains on both benchmarks, indicating that reconstruction and cross-view consistency provide complementary supervision for robust target representation learning.

Analysis of EWSB. Tab. 4 evaluates energy weighting and saturation regulation on UAV123. The unweighted budget reduces power consumption but slightly decreases tracking accuracy. The energy-weighted budget improves precision and success rate over the baseline while reducing power consumption from 5.8 to 5.1 mJ. Adding saturation regulation further improves precision and success rate to 87.1% and 67.9%, respectively, while lowering estimated power consumption to 4.4 mJ.

Impact of λ_{con}. Tab. 5 studies the influence of the cross-view consistency weight. $\lambda_{\text{con}} = 0.5$ achieves the best overall performance, while a smaller value provides insufficient regularization and an excessively large value may over-constrain temporal appearance variations.

Impact of r_{tar}. Tab. 6 analyzes different target firing-rate budgets. A smaller r_{tar} generally reduces the average spike firing rate and power consumption, whereas an overly restrictive budget may degrade tracking performance, especially on UAVDT. We set $r_{\text{tar}} = 0.12$, which reduces Avg. SFR from 0.188 to 0.140 and power consumption from 5.80 mJ to 4.42 mJ, while improving tracking performance on both benchmarks.

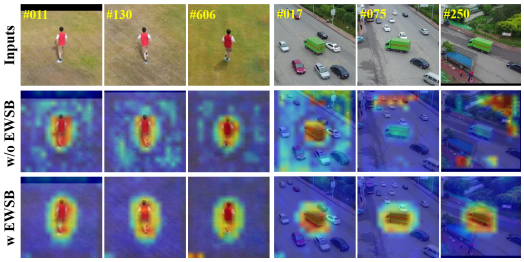

Figure 3: Comparison of search feature activation maps. The second and third rows show the activations without and with EWSB, respectively.

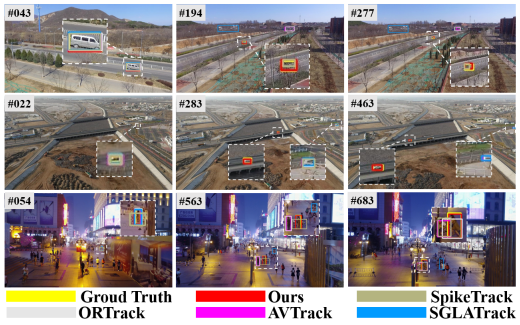

Figure 4: Qualitative evaluation on three video sequences from UAVDT, UAV123, and VisDrone2018, respectively.

Table 6: Impact of r_{tar} on tracking accuracy, average spike firing rate (SFR), and power consumption.

r_{tar}	UAV123		UAVDT		Avg. SFR ↓	Power (mJ) ↓
	Succ.	Prec.	Succ.	Prec.		
Baseline	67.0	86.1	61.2	78.5	0.188	5.80
0.08	**68.8**	**88.7**	61.9	79.9	**0.131**	**4.12**
0.10	67.4	86.4	61.8	80.2	0.133	4.21
0.12	67.9	87.1	**64.1**	**82.9**	0.140	4.42
0.14	66.6	85.3	63.2	80.8	0.146	4.61
0.16	67.0	86.0	62.9	80.8	0.153	4.77

Table 7: Inference speed, tracking accuracy, and theoretical power consumption on VisDrone2018.

Method	Speed (FPS)		Prec.	Succ.	Power (mJ)
	GPU	CPU			
OSTrack	65.4	5.9	84.2	64.8	98.9
ORTrack	**226.4**	**55.4**	88.6	66.8	11.0
SpikeTrack	60.4	12.1	80.2	60.3	8.1
SBMVTrack	118.8	24.4	**90.0**	**70.0**	4.4
SBMVTrack-S	157.0	33.3	85.3	66.5	**2.5**

Speed Evaluation. As shown in Tab. 7, SBMVTrack achieves higher accuracy and lower power consumption than the compared trackers. The lightweight SBMVTrack-S achieves 157.0 FPS on the GPU and 33.3 FPS on the CPU, demonstrating inference throughput above 30 FPS on the evaluated CPU without requiring specialized neuromorphic hardware.

Qualitative Results. As shown in Fig. 3, the baseline produces dispersed responses over background regions, whereas SBMVTrack concentrates its activations more clearly around the target, indicating more discriminative target representations. Fig. 4 compares SBMVTrack with four advanced UAV trackers on representative sequences from UAVDT, UAV123, and VisDrone2018. Under challenging conditions such as background clutter, similar-object, and small targets, SBMVTrack maintains stable localization, whereas several competing methods fail to track the target.

Limitation. The power efficiency of SBMVTrack is currently evaluated theoretically based on operation counts and spike activity, while its practical efficiency may vary with hardware architectures, memory access, and specific implementation details. In future work, we plan to further validate its actual power consumption and runtime efficiency on neuromorphic hardware platforms.

5 Conclusion

We presented SBMVTrack, a power-efficient fully spiking framework for UAV tracking. SBMVTrack explicitly regulates layer-wise spike activity through Energy-Weighted Spike Budgeting (EWSB) and improves the stability of target representations through Masked Multi-View Target Modeling (MVTM). Extensive experiments on multiple UAV tracking benchmarks demonstrate that SBMVTrack achieves performance competitive with advanced ANN trackers while substantially reducing theoretical power consumption, resulting in a favorable accuracy-power trade-off. We hope this work provides a new perspective on energy-aware training of SNNs for UAV vision tasks.

AI USE STATEMENT

In this work, we used GPT-5.5 to assist with English translation and language polishing of the manuscript, as well as to check formatting. We have not used generative AI tools for theoretical derivation, data analysis, experimental design, or result interpretation. We have reviewed all AI-assisted work: the translation and polishing output was manually checked by the authors for accuracy. We take responsibility for the final content of this work, including text, claims, and experimental results.

ETHICS STATEMENT

UAV visual tracking can support applications such as navigation and emergency response, but can also enable intrusive surveillance. Deployment should respect privacy, obtain appropriate authorization, and comply with applicable restrictions on UAV operation and data use. The theoretical power estimates reported here do not establish measured power savings on deployed hardware.

REPRODUCIBILITY STATEMENT

Section 3 specifies the architecture, neuron dynamics, and training objectives. Section 4.1 reports the training datasets, optimization settings, hyperparameters, hardware, and template-update policy. Sections 4.2 and 4.3 report benchmark comparisons and ablation studies, respectively. The source code will be released upon acceptance.

REFERENCES

Wenrui Cai, Qingjie Liu, and Yunhong Wang. Hiptrack: Visual tracking with historical prompts. In *CVPR*, pp. 19258–19267, 2024.

Ziang Cao, Changhong Fu, Junjie Ye, Bowen Li, and Yiming Li. Hift: Hierarchical feature transformer for aerial tracking. In *ICCV*, pp. 15457–15466, 2021.

Ziang Cao, Ziyuan Huang, Liang Pan, Shiwei Zhang, Ziwei Liu, and Changhong Fu. Tctrack: Temporal contexts for aerial tracking. In *CVPR*, pp. 14798–14808, 2022.

Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qiuhong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: A simplified architecture for visual object tracking. In *ECCV*, pp. 375–392, 2022.

Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *CVPR*, pp. 14572–14581, 2023.

Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Discriminative scale space tracking. *IEEE TPAMI*, 39(8):1561–1575, 2016.

Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *ECCV*, pp. 370–386, 2018.

Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, et al. Lasot: A high-quality large-scale single object tracking benchmark. *IJCV*, 129:439–461, 2021.

Changhong Fu, Ziang Cao, Yiming Li, Junjie Ye, and Chen Feng. Onboard real-time aerial tracking with efficient siamese anchor proposal network. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2021.

Goutam Yelluru Gopal and Maria A Amer. Separable self and mixed attention transformers for efficient object tracking. In *WACV*, pp. 6708–6717, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE TPAMI*, 37(3):583–596, 2014.

Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *ISSCC*, pp. 10–14, 2014.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE TPAMI*, (5):1562–1577, 2019a.

Ziyuan Huang, Changhong Fu, Yiming Li, Fuling Lin, and Peng Lu. Learning aberrance repressed correlation filters for real-time uav tracking. In *ICCV*, pp. 2891–2900, 2019b.

Ben Kang, Xin Chen, Simiao Lai, Yang Liu, Yi Liu, and Dong Wang. Exploring enhanced contextual information for video-level object tracking. In *AAAI*, pp. 4194–4202, 2025.

Ben Kang, Jie Zhao, Xin Chen, Wanting Geng, Bin Zhang, Lu Zhang, Dong Wang, and Huchuan Lu. Uetrack: A unified and efficient framework for single object tracking. In *CVPR*, pp. 20890–20901, 2026.

Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pp. 734–750, 2018.

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.

Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, pp. 4282–4291, 2019.

Shuiwang Li, Yangxiang Yang, Dan Zeng, and Xucheng Wang. Adaptive and background-aware vision transformer for real-time uav tracking. In *ICCV*, pp. 13989–14000, 2023.

Shuiwang Li, Xiangyang Yang, Xucheng Wang, Dan Zeng, Hengzhou Ye, and Qijun Zhao. Learning target-aware vision transformers for real-time uav tracking. *IEEE TGRS*, 62:1–18, 2024a.

Siyi Li and Dit-Yan Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *AAAI*, volume 31, pp. 4140–4146, 2017.

Xiaohai Li, Bineng Zhong, Qihua Liang, Guorong Li, Zhiyi Mo, and Shuxiang Song. Mambalct: Boosting tracking via long-term context state space model. In *AAAI*, volume 39, pp. 4986–4994, 2025.

Yiming Li, Changhong Fu, Fangqiang Ding, Ziyuan Huang, and Geng Lu. Autotrack: Towards high-performance visual tracking for uav with automatic spatio-temporal regularization. In *CVPR*, pp. 11923–11932, 2020.

Yongxin Li, Mengyuan Liu, You Wu, Xucheng Wang, Xiangyang Yang, and Shuiwang Li. Learning adaptive and view-invariant vision transformer for real-time uav tracking. In *ICML*, pp. 28403–28420, 2024b.

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 *ECCV*, pp. 740–755, 2014.

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

Yihao Luo, Min Xu, Caihong Yuan, Xiang Cao, Liangqi Zhang, Yan Xu, Tianjiang Wang, and Qi Feng. Siamsnn: Siamese spiking neural networks for energy-efficient object tracking. In *International conference on artificial neural networks*, pp. 182–194. Springer, 2021.

Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10:1659–1671, 1997.

Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. In *ICCV*, pp. 13444–13454, October 2021.

Matthias Mueller, Neil G. Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, volume 7, pp. 445–461, 2016.

Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, pp. 300–317, 2018.

David Nordström, Johan Edstedt, Fredrik Kahl, and Georg Bökman. Mum: Multi-view masked image modeling for 3d vision. In *CVPR*, pp. 21736–21747, 2026.

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. In *CVPR*, pp. 658–666, 2019.

Yimeng Shan, Zhenbang Ren, Haodi Wu, Wenjie Wei, Rui-Jie Zhu, Shuai Wang, Dehao Zhang, Yichen Xiao, Jieyuan Zhang, Kexin Shi, et al. Sdtrack: A baseline for event-based tracking via spiking neural networks. In *CVPR*, 2026.

Liangtao Shi, Bineng Zhong, Qihua Liang, Ning Li, Shengping Zhang, and Xianxian Li. Explicit visual prompts for visual object tracking. In *AAAI*, volume 38, pp. 4838–4846, 2024.

Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Transformer tracking with cyclic shifting window attention. In *CVPR*, pp. 8791–8800, 2022.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Ning Wang, Wengang Zhou, Qi Tian, Richang Hong, Meng Wang, and Houqiang Li. Multi-cue correlation filters for robust visual tracking. In *CVPR*, pp. 4844–4853, 2018.

Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Advances in Neural Information Processing Systems*, 35:3502–3516, 2022.

Longyin Wen, Pengfei Zhu, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Chenfeng Liu, Hao Cheng, Xiaoyu Liu, Wenya Ma, et al. Visdrone-sot2018: The vision meets drone single-object tracking challenge results. In *ECCV*, pp. 0–0, 2018.

You Wu, Xucheng Wang, Xiangyang Yang, Mengyuan Liu, Dan Zeng, Hengzhou Ye, and Shuiwang Li. Learning occlusion-robust vision transformers for real-time uav tracking. In *CVPR*, pp. 17103–17113, 2025.

Shuiying Xiang, Tao Zhang, Shuqing Jiang, Yanan Han, Yahui Zhang, Xingxing Guo, Licun Yu, Yuechun Shi, and Yue Hao. Spiking siamfc++: Deep spiking neural network for object tracking. *Nonlinear dynamics*, 112(10):8417–8429, 2024.

Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, volume 34, pp. 12549–12556, 2020.

Chaocan Xue, Bineng Zhong, Qihua Liang, Yaozong Zheng, Ning Li, Yuanliang Xue, and Shuxiang Song. Similarity-guided layer-adaptive vision transformer for uav tracking. In *CVPR*, pp. 6730–6740, 2025.

Jingjun Yang, Liangwei Fan, Jinpu Zhang, Xiangkai Lian, Hui Shen, and Dewen Hu. Fully spiking neural networks for unified frame-event object tracking. In *NeurIPS*, 2025.

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 *ICLR*, volume 2024, pp. 52885–52907, 2024.

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 TPAMI*, 47(4):2973–2990, 2025.

Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *ECCV*, pp. 341–357, 2022a.

Junjie Ye, Changhong Fu, Guangze Zheng, Danda Pani Paudel, and Guang Chen. Unsupervised domain adaptation for nighttime aerial tracking. In *CVPR*, pp. 8896–8905, 2022b.

Dan Zeng, Mingliang Zou, Xucheng Wang, and Shuiwang Li. Towards discriminative representations with contrastive instances for real-time uav tracking. In *ICME*, pp. 1349–1354, 2023.

Jiqing Zhang, Malu Zhang, Yuanchen Wang, Qianhui Liu, Baocai Yin, Haizhou Li, and Xin Yang. Spiking neural networks with adaptive membrane time constant for event-based tracking. *IEEE TIP*, 34:1009–1021, 2025.

Qiuyang Zhang, Jiujun Cheng, Qichao Mao, Cong Liu, Yu Fang, Yuhong Li, Mengying Ge, and Shangce Gao. Spiketrack: A spike-driven framework for efficient visual tracking. In *CVPR*, pp. 6802–6811, 2026.

Haojie Zhao, Dong Wang, and Huchuan Lu. Representation learning for visual object tracking by masked appearance transfer. In *CVPR*, pp. 18696–18705, 2023.

Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. Odtrack: Online dense temporal token learning for visual tracking. In *AAAI*, 2024.

Pengzhi Zhong, Jiwei Mo, Dan Zeng, Feixiang He, and Shuiwang Li. Fully spiking neural networks with target awareness for energy-efficient uav tracking. *arXiv preprint arXiv:2603.27493*, 2026.

Jiawen Zhu, Huayi Tang, Xin Chen, Xinying Wang, Dong Wang, and Huchuan Lu. Two-stream beats one-stream: Asymmetric siamese network for efficient visual tracking. In *AAAI*, pp. 10959–10967, 2025.

A Appendix

A.1 Additional Implementation Details

A.1.1 Hardware and Software Environment

All experiments are conducted on a workstation equipped with an Intel i9-10850K CPU, 16 GB system memory, and a single NVIDIA TitanX GPU with 12 GB memory. The implementation is based on Python 3.8 and PyTorch 1.12.1+CU113 under Ubuntu 20.04.

A.1.2 Power evaluation.

Power (mJ). Following the theoretical power evaluation protocol commonly used in prior SNN methods Lei et al. (2025); Shan et al. (2026); Yao et al. (2025; 2024); Yang et al. (2025); Zhang et al. (2026), we estimate the power consumption of ANN and SNN trackers. Theoretical power is calculated as:

$$E_{\mathrm{ANN}} = \mathrm{FLOPs} \times E_{\mathrm{MAC}}, \tag{22}$$

$$E_{\mathrm{SNN}} = \mathrm{FLOPs} \times E_{\mathrm{AC}} \times \mathrm{SFR} \times T \times D, \tag{23}$$

where SFR denotes the average spike firing rate, T is the number of timesteps, and D denotes the maximum integer activation level during training. All theoretical power values are estimated using a 32-bit floating-point implementation under 45-nm technology Horowitz (2014), where $E_{\mathrm{MAC}} = 4.6\,\mathrm{pJ}$ and $E_{\mathrm{AC}} = 0.9\,\mathrm{pJ}$. The reported SFR is averaged across UAVTrack112 Fu et al. (2021), UAVDT Du et al. (2018), VisDrone2018 Wen et al. (2018), and UAV123 Mueller et al. (2016). To provide a fine-grained characterization of spike activity, Tab. 12 reports the layer-wise firing rates of SBMVTrack and SBMVTrack-S across all backbone stages and prediction-head branches.

Table 8: Effect of EWSB loss weights. We fix $r_{\mathrm{tar}} = 0.12$ and disable MVTM for all experiments. Avg. SFR denotes the average spike firing rate, and Avg. SAR denotes the average saturation ratio.

λ_{sat}	$\lambda_{\mathrm{budget}}$	UAV123		UAVDT		Avg. SFR↓	Avg. SAR↓	Power (mJ) ↓
		Prec.	Succ.	Prec.	Succ.			
0.0	0	86.1	67.0	78.5	61.2	0.188	0.044	5.80
	1	86.5	67.4	81.5	63.6	0.169	0.035	5.10
	2	86.8	67.4	82.4	63.6	0.164	0.033	4.99
	3	87.1	67.8	79.9	61.8	0.157	0.031	4.79
0.5	1	87.1	67.9	82.9	64.1	0.140	0.017	4.42
	2	86.4	67.1	76.1	59.6	0.123	0.009	3.93
	3	87.0	67.5	77.2	60.5	0.124	0.007	3.95

Avg. SAR. The average saturation ratio is used to measure the proportion of neurons with excessive spike activation. Specifically, for the l-th spiking layer, it is calculated as:

$$R_l^{\mathrm{sat}} = \frac{1}{N_l} \sum_{i=1}^{N_l} \mathbb{I}(S_{l,i} > \tau_{\mathrm{sat}}), \tag{24}$$

where N_l denotes the number of spike elements in the l-th layer, $\mathbb{I}(\cdot)$ is the indicator function, and τ_{sat} is the saturation threshold. The overall average saturation ratio is obtained by averaging the layer-wise values weighted by their computational costs:

$$R^{\mathrm{sat}} = \sum_{l=1}^{L} \omega_l R_l^{\mathrm{sat}}. \tag{25}$$

A.2 More Implementation Details

A.2.1 Randomized Patchwork Module (RPM)

Following SpikeFET Yang et al. (2025), we adopt the Randomized Patchwork Module (RPM) to organize the two temporal templates and the search region into a joint spatial input. Let $Z_1, Z_2 \in \mathbb{R}^{T\times C\times H_z\times W_z}$ denote the templates and $X \in \mathbb{R}^{T\times C\times H_x\times W_x}$ denote the search region, where $H_x = 2H_z$ and $W_x = 2W_z$.

In our implementation, the two templates are first stacked vertically, with Z_1 above Z_2, to form a template block of size $T \times C \times 2H_z \times W_z$. This block has the same height as the search region and is then placed randomly to its left or right. The resulting input therefore has size $T \times C \times H_x \times (W_x + W_z)$. With two 128×128 templates and a 256×256 search region, RPM produces a 256×384 spatial input without resizing the individual regions. This arrangement preserves the spatial structure within each view while varying the relative placement of the template block and the search region.

A.2.2 Cox-Process Masking

In the mask generation stage, we employ the same spatial Cox-process masking operator used in ORTrack Wu et al. (2025) to produce structured occlusion perturbations. In SBMVTrack, the generated masks are extended from a single template to three temporal views, including the initial template Z_1, online template Z_2, and search region X, and are used with the masked feature reconstruction and cross-view identity-consistency objectives introduced in MVTM.

Given a view $V_i \in \{Z_1, Z_2, X\}$ with spatial resolution $H_i \times W_i$, we first divide it into non-overlapping $b \times b$ patches. The resulting patch grid contains:

$$N_i = \frac{H_i W_i}{b^2} \tag{26}$$

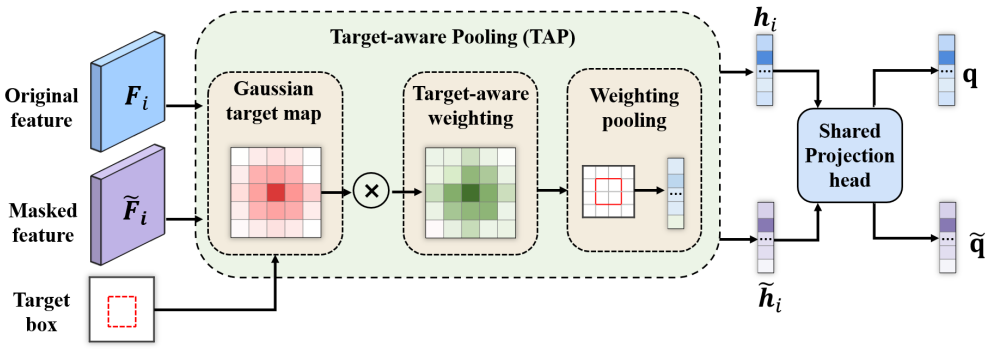

Figure 5: Overview of the Target-Aware Pooling (TAP). TAP takes the original feature map $\mathbf{F}_i$ and masked feature map $\widetilde{\mathbf{F}}_i$ as inputs. The target-aware Gaussian weight map generated from the bounding box is used to aggregate target-related features, producing target representations $\mathbf{h}_i$ and $\widetilde{\mathbf{h}}_i$.

spatial locations. Let ρ_i denote the expected masking ratio. The number of sampled occluded patches is controlled by a Poisson random variable:

$$\Gamma_i \sim \text{Poisson}(\zeta_i), \qquad \zeta_i = \rho_i N_i. \tag{27}$$

Conditioned on Γ_i, a non-homogeneous Poisson point process is sampled over the spatial region $\mathcal{B}_i$. Its intensity function is defined as:

$$\lambda_i(\mathbf{p}) = \frac{\Gamma_i \exp\left(-\frac{\|\mathbf{p}-\mathbf{c}_i\|_2^2}{2\sigma_c^2}\right)}{\displaystyle\int_{\mathcal{B}_i} \exp\left(-\frac{\|\mathbf{u}-\mathbf{c}_i\|_2^2}{2\sigma_c^2}\right) d\mathbf{u}}, \tag{28}$$

where $\mathbf{c}_i$ denotes the spatial center of the corresponding view and σ_c controls the spatial concentration of the sampled patches. Since the intensity is higher around the central region, the resulting masks are more likely to cover spatially correlated target-related patches than uniform random masking.

We use the thinning algorithm to sample a point pattern Ξ_i from λ_i. A binary occlusion map $O_i \in \{0,1\}^{H_i/b \times W_i/b}$ is then constructed, where $O_{i,p} = 1$ indicates that the corresponding patch is masked. The masked view is obtained by:

$$\widetilde{V}_i = V_i \odot \left[(\mathbf{1} - O_i) \otimes \mathbf{1}_{b\times b}\right], \tag{29}$$

where $\odot$ and $\otimes$ denote the Hadamard product and Kronecker product, respectively.

Following ORTrack Wu et al. (2025), we directly adopt its default Cox-process masking configuration without additional tuning. Specifically, the patch size and spatial scale are set to $b = 16$ and $\sigma_c = 64$, respectively, and the expected masking ratio is set to 0.3. We apply the same masking ratio to all three temporal views, i.e., $\rho_{Z_1} = \rho_{Z_2} = \rho_X = 0.3$, throughout all experiments.

A.2.3 Target-Aware Pooling

Target-Aware Pooling (TAP) extracts compact target-level representations from both the original and masked spiking feature maps. As shown in Fig. 5, TAP takes the feature maps $\mathbf{F}_i$ and $\widetilde{\mathbf{F}}_i$ as inputs, where $i \in \{1, 2, x\}$ denotes the initial template, online template, and search region, respectively. Different from global average pooling, which treats all spatial locations equally, TAP exploits the ground-truth bounding box to generate spatially adaptive weights and emphasize target-related responses while suppressing background interference.

Given a feature map $\mathbf{F}_i \in \mathbb{R}^{C\times H_i \times W_i}$ and its normalized bounding box $\mathbf{b}_i = (c_i^x, c_i^y, w_i, h_i)$, TAP first constructs an anisotropic Gaussian weight map:

Table 9: Comparison with representative deep-based trackers on DTB70, UAVDT, VisDrone2018, and UAV123. Prec. and Succ. denote precision and success rate, respectively. Power represents theoretical power consumption estimated under the same protocol. Avg. denotes the average performance over the four benchmarks. The best result is highlighted in bold.

Method	DTB70 (Prec./Succ.)	UAVDT (Prec./Succ.)	VisDrone2018 (Prec./Succ.)	UAV123 (Prec./Succ.)	Avg. (Prec./Succ.)	Power (mJ) ↓
KeepTrackMayer et al. (2021)	83.6 / 64.3	83.8 / 60.5	84.0 / 63.5	88.0 / 69.7	84.9 / 64.5	51.3
CSWinTTSong et al. (2022)	80.3 / 62.3	67.3 / 54.0	75.2 / 58.0	87.6 / 70.5	77.6 / 61.2	75.7
SimTrackChen et al. (2022)	83.2 / 64.6	76.5 / 57.2	80.0 / 60.9	88.2 / 69.2	82.0 / 63.0	93.8
OSTrackYe et al. (2022a)	82.7 / 65.0	85.0 / 63.4	84.2 / 64.8	84.7 / 67.4	84.2 / 65.2	98.9
SeqTrackChen et al. (2023)	85.6 / 65.5	78.7 / 58.8	83.3 / 64.1	86.8 / 68.6	83.6 / 64.3	302.7
MATZhao et al. (2023)	83.2 / 64.5	72.9 / 54.8	81.6 / 62.2	86.7 / 68.3	81.1 / 62.5	197.3
EVPTrackShi et al. (2024)	85.8 / 66.5	80.6 / 61.2	84.5 / 65.8	88.9 / 70.2	85.0 / 65.9	300.8
HIPTrackCai et al. (2024)	81.2 / 68.6	81.2 / 60.9	86.7 / 67.3	91.3 / 70.2	85.1 / 66.8	307.7
ODTrackZheng et al. (2024)	**90.0/ 70.0**	**85.8** / 63.8	85.6 / 64.7	89.7 / **70.4**	**87.8** / 67.2	335.8
MambaLCTLi et al. (2025)	88.3 / 68.7	84.4 / 63.6	88.1 / 65.4	88.2 / 69.0	87.2 / 66.7	119.1
MCITrackKang et al. (2025)	87.3 / 68.1	83.9 / **66.4**	81.4 / 62.1	**91.9** / **70.4**	86.1 / 66.8	176.9
SBMVTrack	84.0 / 66.8	84.3 / 64.8	**90.0** / **70.0**	87.6 / 68.0	86.5 / **67.4**	4.4
SBMVTrack-S	81.6 / 64.8	83.1 / 63.8	85.3 / 66.5	85.6 / 66.6	83.9 / 65.4	**2.5**

$$A_i(r,c) = \exp\left(-\frac{(x_c - c_i^x)^2}{2(\sigma_i^x)^2} - \frac{(y_r - c_i^y)^2}{2(\sigma_i^y)^2}\right), \tag{30}$$

where

$$x_c = \frac{c - 0.5}{W_i}, \qquad y_r = \frac{r - 0.5}{H_i}, \quad c = 1, \ldots, W_i, \quad r = 1, \ldots, H_i. \tag{31}$$

and the spatial scales are determined by:

$$\sigma_i^x = \kappa_i w_i, \qquad \sigma_i^y = \kappa_i h_i. \tag{32}$$

We set $\kappa_1 = \kappa_2 = 0.6$ for the two template views and $\kappa_x = 0.5$ for the search view. Therefore, the spatial extent and aspect ratio of the Gaussian kernel adapt to the size and shape of the target.

The target-aware representation is obtained through normalized weighted pooling:

$$\mathbf{h}_i = \frac{\sum_{r=1}^{H_i}\sum_{c=1}^{W_i} A_i(r,c)\mathbf{F}_i(:,r,c)}{\sum_{r=1}^{H_i}\sum_{c=1}^{W_i} A_i(r,c) + \epsilon}, \tag{33}$$

where ϵ is a small constant for numerical stability. The same operation is applied to the masked feature map $\widetilde{\mathbf{F}}_i$, producing $\widetilde{\mathbf{h}}_i$. Finally, a shared projection head $\mathcal{P}(\cdot)$ maps these pooled representations into the embedding space:

$$\mathbf{q}_i = \mathcal{P}(\mathbf{h}_i), \qquad \widetilde{\mathbf{q}}_i = \mathcal{P}(\widetilde{\mathbf{h}}_i). \tag{34}$$

The projected embeddings are subsequently used to compute the masked feature reconstruction loss and the cross-view identity-consistency loss. TAP and the projection head are used only during training and introduce no additional inference cost.

A.3 More Exploring Experiments

A.3.1 Effect of EWSB weights.

Tab. 8 investigates the influence of the budget and saturation loss weights in EWSB. We first set $\lambda_{\text{sat}} = 0$ and vary λ_{budget}. Increasing λ_{budget} gradually reduces the average spike firing rate, saturation ratio, and power consumption, indicating that the spike budget constraint effectively suppresses redundant spike activities. However, an excessively large constraint may introduce performance fluctuations due to over-regularization.

Table 10: Ablation study of masking and pooling strategies in MVTM. All variants use the same training objectives and inference pipeline.

Method	Pooling	UAV123		VisDrone2018	
		Prec.	Succ.	Prec.	Succ.
Random Patch	GAP	84.2	65.7	83.6	64.7
Random Patch	TAP	85.1	66.3	84.7	65.3
Cox-Process	GAP	84.4	65.0	**85.1**	66.2
Cox-Process	TAP	**87.4**	**68.0**	84.8	**66.7**

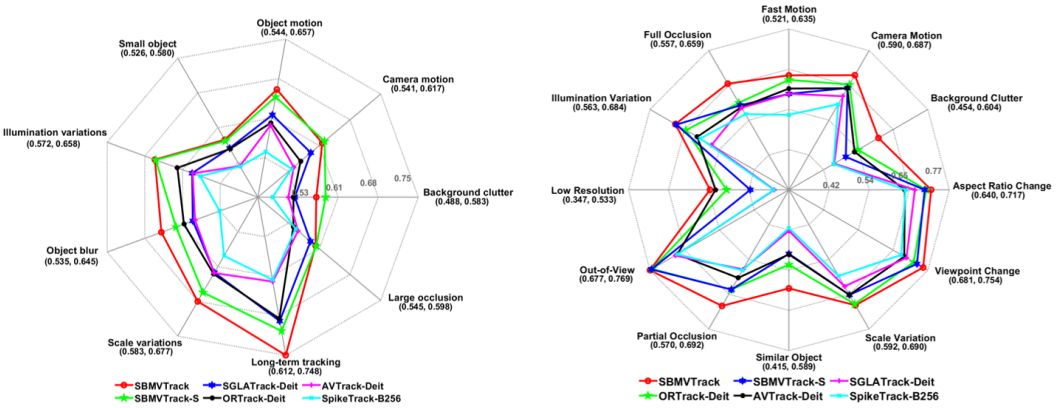

Figure 6: Attribute-based success rate analysis of SBMVTrack and compared trackers under diverse UAV tracking challenges on UAVDT (left) and VisDrone2018 (right).

We further introduce the saturation penalty by setting $\lambda_{\text{sat}} = 0.5$. Compared with using the budget constraint alone, the saturation penalty further reduces the saturation ratio and power consumption while maintaining competitive tracking performance. The setting $\lambda_{\text{sat}} = 0.5$ and $\lambda_{\text{budget}} = 1$ achieves the best overall balance, reducing the power consumption from 5.80 mJ to 4.42 mJ, while improving the tracking performance on both UAV123 and UAVDT.

A.3.2 Effect of Masking and Pooling Strategies.

Tab. 10 evaluates the influence of masking and pooling strategies in MVTM. Compared with random patch masking, the Cox-process masking introduces spatially structured perturbations and achieves better overall performance, demonstrating its effectiveness in constructing informative temporal views. Moreover, replacing global average pooling (GAP) with target-aware pooling (TAP) improves success rates on both benchmarks under both masking strategies.

A.3.3 Comparison with Deep Trackers.

Tab. 9 further compares SBMVTrack with 11 state-of-the-art deep trackers on DTB70 Li & Yeung (2017), UAVDT Du et al. (2018), VisDrone2018 Wen et al. (2018) and UAV123 Mueller et al. (2016). Although these trackers generally achieve strong tracking performance with larger models and higher computational costs, SBMVTrack obtains competitive or superior results with substantially lower power consumption. Specifically, SBMVTrack achieves the highest average success rate of 67.4% among all compared trackers while consuming only 4.4 mJ of theoretical power. On VisDrone2018, SBMVTrack obtains the best precision and success rate of 90.0% and 70.0%, respectively, demonstrating its effectiveness in challenging UAV scenarios. Compared with high-performance Transformer-based trackers, SBMVTrack maintains comparable tracking accuracy with significantly improved power efficiency. For example, SBMVTrack achieves a higher average success rate than ODTrack and MCITrack while reducing power consumption by more than 70× and 40×, respectively. Compared with MambaLCT, which is designed for efficient tracking, SBMVTrack still achieves a higher average success rate with only about 3.7% of its power con-

Table 11: Generalization of EWSB to existing SNN-based trackers. Since SpikeFET requires synchronized frame-event inputs, it is evaluated on FE108, while the RGB-based SpikeTrack is evaluated on three UAV benchmarks. Prec. and AUC denote precision and the area under the success plot, respectively.

Tracker	EWSB	Dataset	Prec. ↑	AUC ↑	Power	
					(mJ) ↓	Reduction
SpikeFET-Tiny	–	FE108	95.8	**67.9**	18.4	–
	✓	FE108	**96.0**	67.6	**14.2**	**22.8%**
SpikeTrack	–	UAVDT	**75.9**	**56.3**	8.1	–
		VisDrone2018	80.2	60.3		
		UAV123	88.7	67.5		
	✓	UAVDT	75.6	56.2	**6.3**	**22.2%**
		VisDrone2018	**80.7**	**62.0**		
		UAV123	**89.3**	**68.1**		

sumption. These results demonstrate that SBMVTrack further narrows the performance gap between SNN-based and ANN-based trackers, achieving competitive tracking accuracy while preserving the inherent power efficiency of SNNs.

A.3.4 Application to Existing SNN Trackers

To evaluate the generalizability of EWSB, we integrate it into two representative SNN-based trackers, SpikeFET-Tiny Yang et al. (2025) and SpikeTrack-B256 (T=1) Zhang et al. (2026), without modifying their original inference architectures. Since SpikeFET requires synchronized frame-event inputs, it is evaluated on FE108. In contrast, the RGB-based SpikeTrack is evaluated on UAVDT, VisDrone2018, and UAV123.

As shown in Tab. 11, EWSB reduces the theoretical power consumption of SpikeFET-Tiny from 18.4 mJ to 14.2 mJ, corresponding to a reduction of 22.8%, while maintaining comparable tracking performance. In particular, its precision slightly increases from 95.8% to 96.0%, with only a marginal AUC variation of 0.3 percentage points.

For SpikeTrack, EWSB reduces the theoretical power consumption from 8.1 mJ to 6.3 mJ, yielding a 22.2% reduction. Averaged over the three UAV benchmarks, EWSB improves precision and AUC by 0.3 and 0.7 percentage points, respectively. Although minor performance variations are observed on UAVDT, consistent improvements are achieved on VisDrone2018 and UAV123. These results demonstrate that EWSB can effectively regulate spike activity across different SNN architectures and input modalities without introducing additional inference components.

A.3.5 Attribute-based Evaluation

Fig. 6 presents attribute-based success plots on UAVDT Du et al. (2018) and VisDrone2018 Wen et al. (2018). We compare SBMVTrack with representative lightweight ANN-based trackers (ORTrack-DeiT Wu et al. (2025), AVTrack-DeiT Li et al. (2024b), and SGLATrack-DeiT Xue et al. (2025)) and the SNN-based tracker SpikeTrack-B256 Zhang et al. (2026).

SBMVTrack achieves consistently competitive performance across diverse challenging attributes. On UAVDT, it performs particularly well under long-term tracking, object motion, small object, and scale variation conditions. On VisDrone2018, it shows clear advantages in Camera motion, viewpoint change, partial occlusion, and similar object scenarios. These results demonstrate that SBMVTrack effectively improves target robustness while maintaining the power efficiency of SNNs.

Table 12: Layer-wise firing rates of SBMVTrack and SBMVTrack-S.

Stage	Module	Submodule	Layer	SBMVTrack	SBMVTrack-S
Stage 1	Downsampling		Conv	1.0000	1.0000
	Conv-Based SNN Block	SSConv	PWConv1	0.2509	0.2131
			DWConv	0.1182	0.0724
			PWConv2	0.1389	0.0868
		ChConv	Conv1	0.2509	0.3154
			Conv2	0.0477	0.0549
Stage 2	Downsampling		Conv	0.2208	0.2504
	Conv-Based SNN Block	SSConv	PWConv1	0.1759	0.1822
			DWConv	0.1554	0.1619
			PWConv2	0.1202	0.1093
		ChConv	Conv1	0.2512	0.2911
			Conv2	0.0359	0.0449
Stage 3	Downsampling		Conv	0.2351	0.2597
	Conv-Based SNN Block 1	SSConv	PWConv1	0.2084	0.1872
			DWConv	0.1895	0.1897
			PWConv2	0.1234	0.1338
		ChConv	Conv1	0.2120	0.2243
			Conv2	0.0249	0.0303
	Conv-Based SNN Block 2	SSConv	PWConv1	0.2899	0.3186
			DWConv	0.1360	0.1362
			PWConv2	0.1037	0.1108
		ChConv	Conv1	0.1980	0.2364
			Conv2	0.0185	0.0256
Stage 4	Downsampling		Conv	0.2669	0.3010
	Transformer-Based SNN Block 1	SSConv	PWConv1	0.1694	0.1938
			DWConv	0.1752	0.2062
			PWConv2	0.0763	0.0840
		E-SDSA	Q_S	0.1916	0.2227
			K_S	0.1916	0.2227
			V_S	0.1916	0.2227
			$Q_S K_S^T$	0.0096	0.0087
			Context	0.1028	0.0929
			Linear	0.3787	0.3132
	Transformer-Based SNN Block 2	SSConv	PWConv1	0.2967	0.3254
			DWConv	0.1603	0.1797
			PWConv2	0.0989	0.1023
		E-SDSA	Q_S	0.2863	0.2963
			K_S	0.2863	0.2963
			V_S	0.2863	0.2963
			$Q_S K_S^T$	0.0062	0.0057
			Context	0.0594	0.0429
			Linear	0.2834	0.2122
	Transformer-Based SNN Block 3	SSConv	PWConv1	0.3269	0.3326
			DWConv	0.1814	0.1877
			PWConv2	0.1056	0.1026
		E-SDSA	Q_S	0.3120	0.3032

Continued on next page

Table 12 – continued from previous page

Stage	Module	Submodule	Layer	SBMVTrack	SBMVTrack-S
			K_S	0.3120	0.3032
			V_S	0.3120	0.3032
			$Q_S K_S^T$	0.0054	0.0064
			Context	0.0514	0.0367
			Linear	0.2598	0.2183
	Transformer-Based SNN Block 4	SSConv	PWConv1	0.2889	0.3394
			DWConv	0.1619	0.1929
			PWConv2	0.0991	0.1186
		E-SDSA	Q_S	0.2756	0.3323
			K_S	0.2756	0.3323
			V_S	0.2756	0.3323
			$Q_S K_S^T$	0.0036	0.0072
			Context	0.0401	0.0280
			Linear	0.1309	0.2142
	Transformer-Based SNN Block 5	SSConv	PWConv1	0.2567	0.3105
			DWConv	0.1626	0.2012
			PWConv2	0.0858	0.1053
		E-SDSA	Q_S	0.2353	0.2926
			K_S	0.2353	0.2926
			V_S	0.2353	0.2926
			$Q_S K_S^T$	0.0039	0.0051
			Context	0.0478	0.0333
			Linear	0.1398	0.1308
	Transformer-Based SNN Block 6	SSConv	PWConv1	0.2431	0.2959
			DWConv	0.1612	0.1951
			PWConv2	0.0749	0.1000
		E-SDSA	Q_S	0.2189	0.2805
			K_S	0.2189	0.2805
			V_S	0.2189	0.2805
			$Q_S K_S^T$	0.0028	0.0035
			Context	0.0539	0.0397
			Linear	0.1128	0.0959
Stage 5	Downsampling		Conv	0.1765	0.2508
	Transformer-Based SNN Block 1	SSConv	PWConv1	0.1527	0.2053
			DWConv	0.0797	0.1077
			PWConv2	0.0810	0.0884
		E-SDSA	Q_S	0.2128	0.2493
			K_S	0.2128	0.2493
			V_S	0.2128	0.2493
			$Q_S K_S^T$	0.0033	0.0034
			Context	0.0142	0.0167
			Linear	0.0691	0.0708
	Transformer-Based SNN Block 2	SSConv	PWConv1	0.2496	0.2702
			DWConv	0.0362	0.0425
			PWConv2	0.0469	0.0417
		E-SDSA	Q_S	0.2044	0.2433
			K_S	0.2044	0.2433
			V_S	0.2044	0.2433
			$Q_S K_S^T$	0.0032	0.0030
			Context	0.0076	0.0075

Continued on next page

Table 12 – continued from previous page

Stage	Module	Submodule	Layer	SBMVTrack	SBMVTrack-S
			Linear	0.0615	0.0533
Head	Ctr	–	Conv1	0.1360	0.1683
		–	Conv2	0.0557	0.0595
		–	Conv3	0.0669	0.0704
		–	Conv4	0.1470	0.1374
		–	Conv5	0.1470	0.1374
	Offset	–	Conv1	0.1360	0.1683
		–	Conv2	0.0634	0.0632
		–	Conv3	0.0895	0.1002
		–	Conv4	0.1333	0.1289
		–	Conv5	0.1333	0.1289
	Size	–	Conv1	0.1360	0.1683
		–	Conv2	0.0734	0.0724
		–	Conv3	0.1174	0.1102
		–	Conv4	0.1960	0.1700
		–	Conv5	0.1960	0.1700