Spiking Transformers for Event-based Single Object Tracking (Supplementary Material)

Jiqing Zhang[1], Bo Dong[2,⋆], Haiwei Zhang[1], Jianchuan Ding[1], Felix Heide[2], Baocai Yin[1], Xin Yang[1,⋆]
[1]Dalian University of Technology, [2] Princeton University
https://github.com/Jee-King/CVPR2022_STNet

In this Supplemental Material, we present additional information and experimental results in support of the findings from the main paper. We first describe the architecture details of the two main components of the proposed STNet: SNNformer Feature Extractor (SFE) and Temporal-Spatial Feature Fusion (TSFF); see section 2. Next, in section 3, we provide the raw RSRs and RPRs of the top five methods on the FE240hz [13] dataset. We provide additional evaluations for non-rigid and rigid objects on the VisEvent [7] dataset, and the corresponding results are shown in section 4. In Section 5, we demonstrate the impacts of the three essential hyperparameters, which are the decay factor α, the spiking threshold $\mathcal{V}_{th}$, and the number of accumulated event-frames, n. Section 6 provides ablation experiments validating the architecture choices we made for the proposed model. In Section 7, we provide insights into the capability of the proposed model in filtering noise. Finally, we provide a *Supplemental Video* to intuitively demonstrate the effectiveness of the proposed STNet under four different degraded conditions; see section 8.

1. Summary of Symbols

In Figure 1, 2, and 4, the following symbols are used:
$\mathcal{C}_k$ denotes a convolutional layer with a kernel size of $k \times k$;
ψ is an operator consisting of Batch Normalization (BN) and a ReLU activation function;
μ is a MEAN operation;
$\mathcal{A}$ denotes adaptive average pooling;
σ is a sigmoid function;
$\mathcal{R}$ denotes a reshape function;
$\cdot$ denotes matrix multiplication;
φ is a softmax function;
$\mathcal{M}$ is a three-layer MLP operator with one linear input layer, one ReLU activation function, and one linear output layer;
$\mathcal{C}$ denotes a concatenation operation;
$\otimes$ and $\oplus$ denote the element-wise multiplication and addition, respectively;
p denotes max pooling;
t denotes a tanh function.

2. SFE and TSFF

We provide two detailed illustrations for the proposed SFE and TSFF in Figure 1 and Figure 2, respectively. Compared to their counterparts in the main manuscript, these two schematics provide more details in architecture, which are essential in understanding our design intuitively.

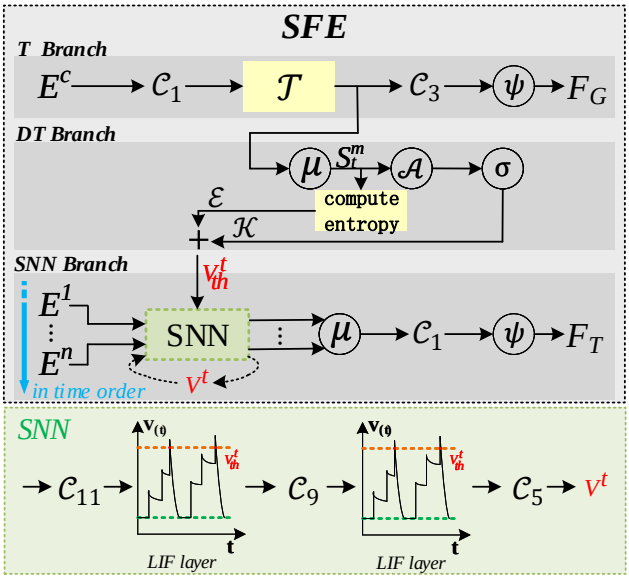


Figure 1. The detailed architecture of the SNNformer Feature Extractor (SFE). $\mathcal{T}$ is a reduced Swin-transformer [10]. The compute entropy module is implemented based on Eq. 15.

3. Raw RSRs and RPRs

The raw RSRs and RPRs of the top five methods on the FE240hz [13] dataset under four different adverse conditions are provided in Table 1. Specifically, the four adverse

⋆ Xin Yang (xinyang@dlut.edu.cn) is the corresponding authors. Xin Yang and Bo Dong lead this project.

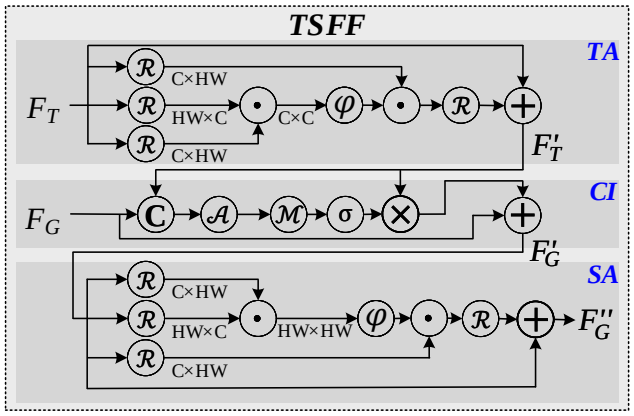


Figure 2. The detailed architecture of the Temporal-Spatial Feature Fusion (TSFF) module.

conditions are: (a) scenes with objects similar to the object being tracked (SOSOBT); (b) severe camera motion; (c) scenes illuminated with a strobe light; and (d) high dynamic range (HDR) scenes.

We report the raw RSRs and RPRs with respect to the tracking speed of the top five methods on the FE240hz [13] dataset in Table 2. The results validate that the proposed STNet offers the best performance in both accuracy and speed. We note that the time for preprocessing events is not included when estimating the tracking speeds of all trackers.

	Methods	KYS [3]	PrDiMP [6]	STRAK-S [12]	TransT [4]	STNet
RSR	SOSOBT	0.446	0.524	0.471	0.516	**0.535**
	severe camera motion	0.297	0.286	0.269	0.330	**0.351**
	strobe light	0.335	0.391	0.321	0.382	**0.392**
	HDR	0.503	0.505	0.493	0.480	**0.508**
	over all	0.553	0.552	0.554	0.567	**0.585**
RPR	SOSOBT	0.628	0.737	0.627	0.742	**0.746**
	severe camera motion	0.565	0.538	0.429	0.562	**0.622**
	strobe light	0.483	0.565	0.397	0.549	**0.574**
	HDR	0.825	0.815	0.755	0.820	**0.832**
	over all	0.878	0.868	0.837	0.890	**0.896**

Table 1. The RSRs and RPRs of the top five competing approaches on the FE240hz [13] dataset under four degraded conditions. SOSOBT means scenes with objects similar to the object being tracked.

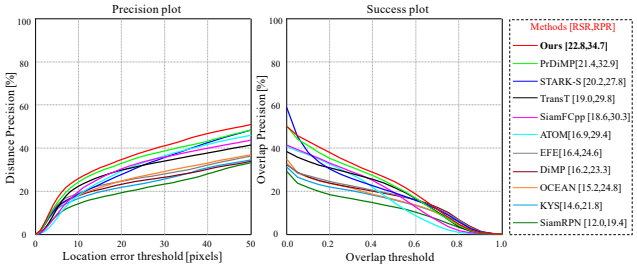
Methods	KYS [3]	PrDiMP [6]	STRAK-S [12]	TransT [4]	STNet
RSR	0.553	0.552	0.554	0.567	**0.585**
RPR	0.878	0.868	0.837	0.890	**0.896**
Speed(fps)	27.8	36.2	41.3	43.8	**85.3**

Table 2. The RSRs and RPRs with respect to tracking speed of the top five competing approaches on the FE240hz [13] dataset.

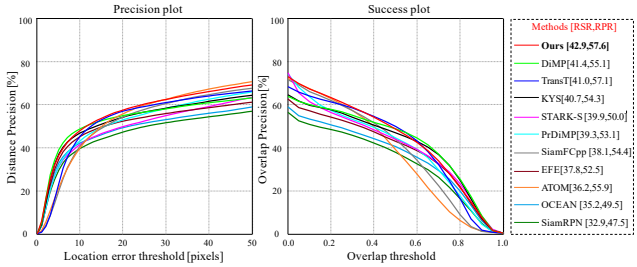
4. Additional Evaluations

The FE240hz and EED datasets only contain rigid objects. To assess the effectiveness of the proposed STNet with non-rigid objects, we use the VisEvent [7] dataset. However, at the moment of conducting our experiments, VisEvent has much inaccurate or missing information, making it hard to be used directly. We manually check all the sequences of the VisEvent dataset and remove the problematic ones (*e.g.*, missing event data or misaligned timestamps between frame and event domains). Eventually, we obtain 377 sequences for training and 172 for testing. Among the 172 testing sequences, 63 of them contain non-rigid objects.

We train the proposed STNet on the 377 sequences and test the trained STNet in three different ways: first, we use all 172 testing sequences to assess the effectiveness of the STNet with both rigid and non-rigid objects presented, and the corresponding results are reported in the main paper; Second, we evaluate the trained STNet with the 63 sequences that only contain non-rigid objects, and Figure Figure 3 (a) shows the experimental results under this setting; Third, the effectiveness of the STNet on single rigid object tracking is assessed with the rest 109 sequences, and the corresponding results are shown in Figure Figure 3 (b). The overall tracking performance is reported in Table Table 4. Based on these experimental results, the proposed STNet offers the best performance in rigid and non-rigid object tracking compared to other competing approaches.



(a) Precision (left) and Success (right) plot on non-rigid objects



(b) Precision (left) and Success (right) plot on rigid objects

Figure 3. Precision (left) and Success (right) plot on the rigid and non-rigid objects the VisEvent [7] dataset.

5. Impact of Hyperparameters

There are three essential hyperparameters in the proposed STNet: the decay factor α of the *leaky integrate-and-fire (LIF)* [8] spiking neuron, used in Eq. 10; The spiking threshold V_{th} in Eq. 13, defining the spiking triggering potential value; The n in Eq. 7, which is the number of aggregated event-frames. Here, we study their impact on single object tracking performance with the FE240hz dataset in accuracy and efficiency.

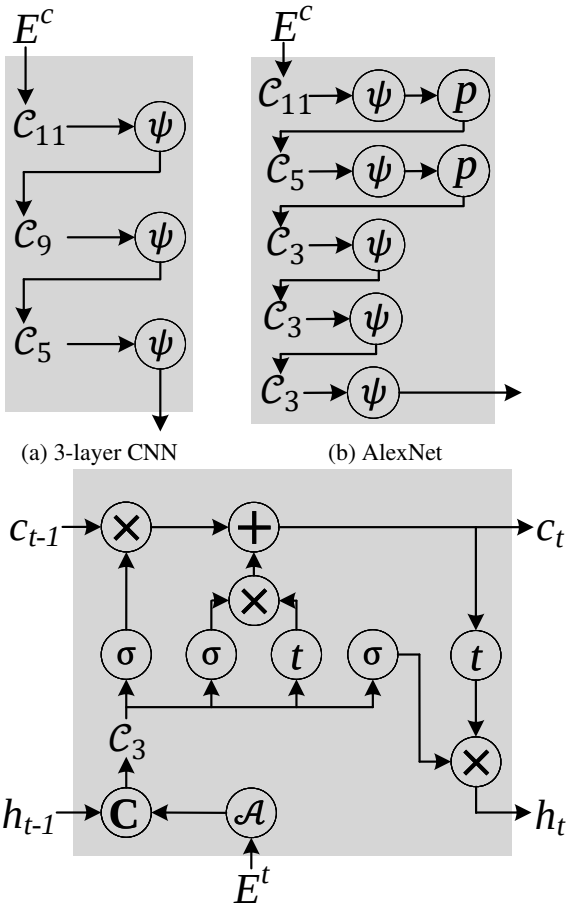
Based on the experimental results shown in Table 3, the values of α and $\mathcal{V}_{th}$ used in our experiments (*i.e.*, 0.2 and 0.3) offer the best results in both tracking accuracy and efficiency. For the value of n, with a larger value, it increases the tracking accuracy but slows down the tracking efficiency, and vice versa. In our experiments, we set it to 5, offering the best trade-off between the tracking accuracy and efficiency.

	α			V_{th}			n		
	0.1	0.2	0.3	0.1	0.3	0.5	3	5	10
RSR	57.9	58.5	57.7	58.1	58.5	57.0	56.9	58.5	58.8
RPR	88.4	89.6	88.4	88.7	89.6	88.3	88.0	89.6	90.3
FPS	85.3	85.3	85.3	85.3	85.3	85.3	97.4	85.3	56.2

Table 3. Hyperparameter evaluations on the FE240hz dataset.

6. Models in Ablation Experiments

In our ablation experiments, we replace the 3-layer SNN model in the SNN branch of the SFE module with the following three models: (a) a 3-layer CNN; (b) an AlexNet; and (c) an LSTM. The ablated (and replaced) model parts are listed in Figure 4 (a), (b), and (c), respectively.



(a) 3-layer CNN (b) AlexNet

(c) A cell in LSTM

Figure 4. The detailed architectures of the three models used in our ablation experiments; (a) 3-layer CNN, (b) AlexNet, (c) LSTM.

7. Robustness to Noise

The spiking mechanism of SNNs acts as a natural noise filter. To get insights into this capability, we use the SNN branch (SB) of the SFE module to conduct the following experiments: (i) replacing the SNN with the LSTM shown in Figure 4 (c); (ii) using the original SNN network, but with a fixed spiking threshold, 0.3; (iii) using the original SNN and with the proposed dynamic spiking threshold scheme. The experimental results are shown in Figure 5, indicating that (iii) offers the best noise filtering results and (i) results in the worst performance. Based on (ii) and (iii), we see the dynamic threshold plays an essential role in filtering out noise. The corresponding visualizations are obtained based on the output features of the SNN branch, (F_T), for (ii) and (iii). For (i), we use the LSTM output feature vector, which is the mean of the output h_t of all cells, for visualization. The heatmaps shown in Figure 5 are plotted by *Seaborn* library [1], which directly reflects the raw value of the feature maps.

8. Supplemental Video

To demonstrate the effectiveness of the proposed STNet, we provide additional qualitative comparisons of STNet compared to state-of-the-art trackers under different degraded conditions in our *Supplemental Video*, including (a) scenes with objects similar to the object being tracked; (b) severe camera motion; (c) scenes illuminated with a strobe light; and (d) high dynamic range (HDR) scenes. We refer to our video for additional details. The *Supplemental Video* is available at https://youtu.be/m5LtG4oUQNs

Dataset	Metrics	SiamRPN [9]	ATOM [5]	DiMP [2]	SiamFC++ [11]	OCEAN [14]	KYS [3]	PrDiMP [6]	STRAK-S [12]	TransT [4]	EFE [13]	STNet
non-rigid	RSR ↑	12.0	16.9	16.2	18.6	15.2	14.6	21.4	20.2	19.0	16.4	**22.8**
	RPR ↑	19.4	29.4	23.3	30.3	24.8	21.8	32.9	27.8	29.8	24.6	**34.7**
rigid	RSR ↑	32.9	36.2	41.4	38.1	35.2	40.7	39.3	39.9	41.0	37.8	**42.9**
	RPR ↑	47.5	55.9	55.1	54.4	49.5	54.3	53.1	50.0	57.1	52.5	**57.6**

Table 4. State-of-the-art comparison for the non-rigid objects and rigid objects on the VisEvent [7] dataset in representative success rate (RSR) and representative precision rate (RPR).

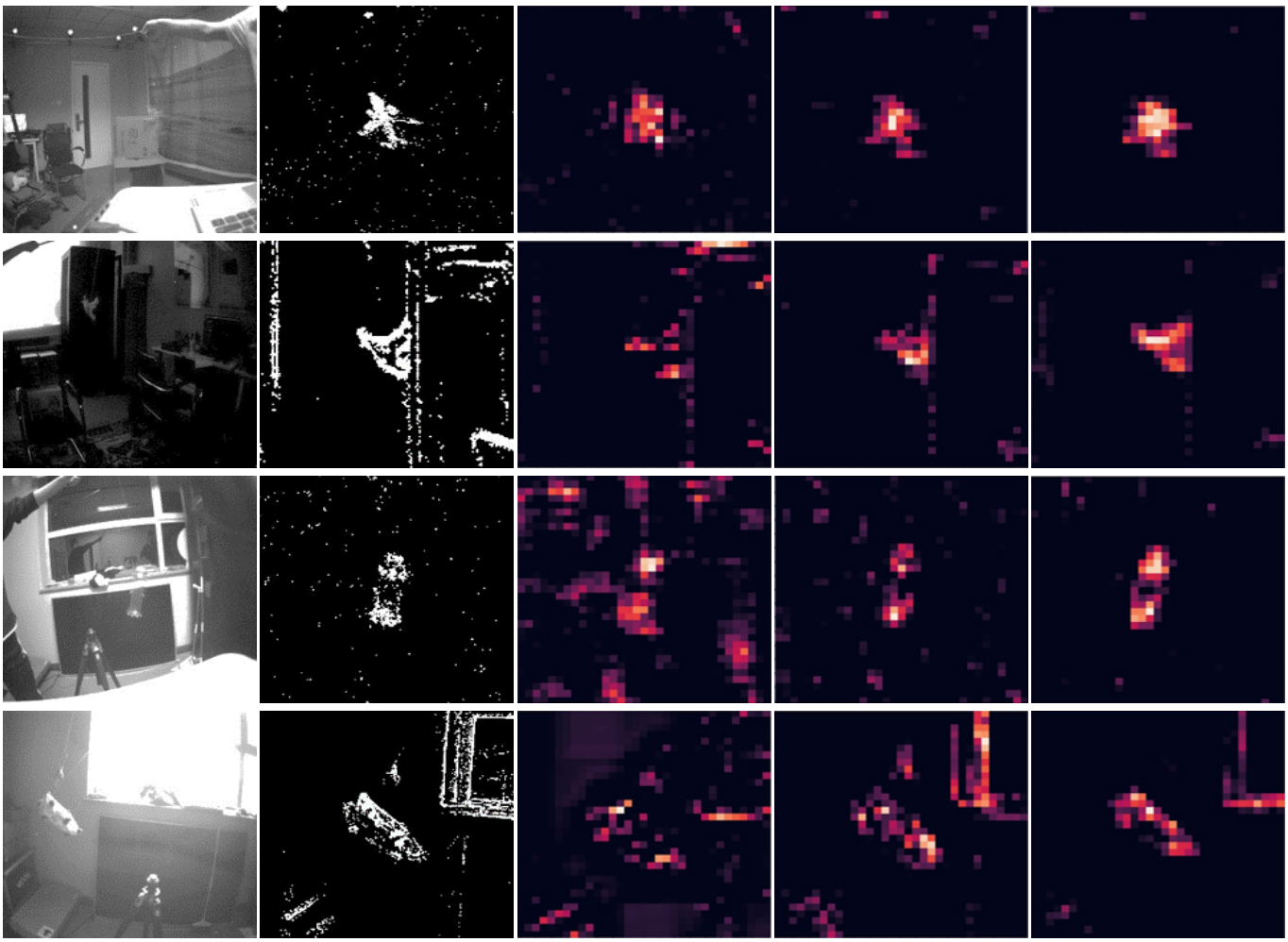


Figure 5. Qualitative comparison of three competing models in noise robustness. We use the SNN branch (SB) of the SFE module to conduct the following experiments: (i) replacing the SNN with an LSTM; (ii) using the original SNN network, but with a fixed spiking threshold, 0.3; (iii) using the original SNN and with the proposed dynamic spiking threshold scheme.

References

[1] Seaborn: Statistical data visualization. https://seaborn.pydata.org/. 3

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019. 4

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *ECCV*, 2020. 2, 4

[4] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, 2021. 2, 4

[5] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, 2019. 4

[6] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, 2020. 2, 4

[7] Wang et al. Visevent: Reliable object tracking via collaboration of frame and event flows. *arXiv*, 2021. 1, 2, 4

[8] Wulfram Gerstner and Werner M. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. 2002. 3

[9] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018. 4

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, 2021. 1

[11] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, 2020. 4

[12] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. *ICCV*, 2021. 2, 4

[13] Jiqing Zhang, Xin Yang, Yingkai Fu, Xiaopeng Wei, Baocai Yin, and Bo Dong. Object tracking by jointly exploiting frame and event domain. In *ICCV*, 2021. 1, 2, 4

[14] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020. 4