Event Trojan: Asynchronous Event-based Backdoor Attacks

Anonymous ECCV 2024 Submission

Paper ID #1155

7 Overview

This supplementary document provides more event representation strategies, backdoor attack training details, experimental results, and visualization examples that accompany the paper:

- Sec. 8 illustrates the process of training a backdoor model on the event data.
- Sec. 9 presents more details about the event data and popular event representation strategies.
- Sec. 10 provides detailed experimental results of 22 classifiers on N-Caltech101 and N-Cars datasets.
- Sec. 11 shows more visualization results of triggered samples poisoned by three types of triggers: representation trigger, immutable trigger, and mutable trigger, respectively. Additionally, the point sets of the poisoned event data generated by immutable and mutable triggers are depicted in Fig. S4.

8 Backdoor attack on event vision models

In Fig. 2 of the main paper, we show the details of training event vision model and the design of our proposed two triggers. For training a backdoored event vision model, we need first to generate some poisoned samples by Fig. 2 (c) or (d) of the main paper. Then, we can follow the pipeline shown in Fig. S1 to train a victim model and evaluate the attacking performance. The backdoored model can correctly classify benign event streams, such as the motorbike and airplane shown in the first row of Fig. S1. However, once the attacker injects the specific trigger into event samples, this model will output the predetermined label. For instance, the poisoned motorbike and ferry (in the second row of Fig. S1) are all misclassified as accordions. This kind of potential risk could severely impact the performance of autonomous driving systems.

9 Event data

As depicted in Sec. 3.2 of the main paper, event data consists of a series of independent and discrete events (x_k, y_k, t_k, p_k), a kind of sparse sequence data. In contrast to conventional images, event data is recorded by the event camera

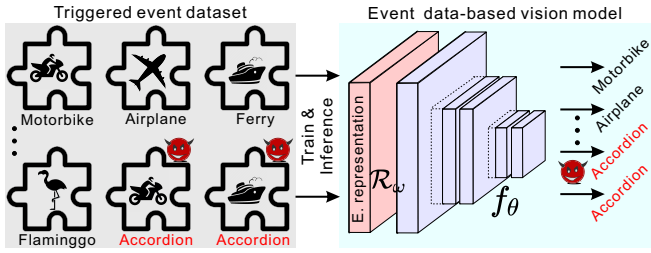

Fig. S1: The framework of the backdoor attack on event vision models. Each "puzzle piece" represents an event data stream. $\mathcal{R}_\omega$ denotes the module of event representation with parameters ω (E. representation), and f_θ represents the victim model with parameters θ.

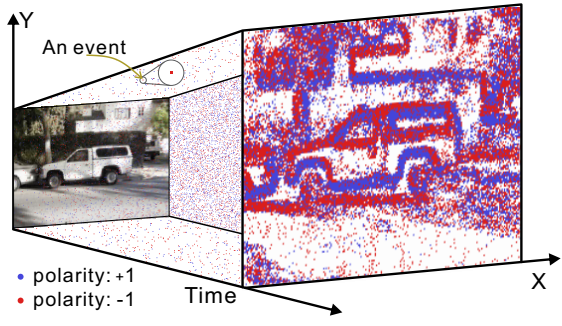

Fig. S2: Compared with conventional cameras, an event camera obtains the data (*e.g.*, an event) asynchronously. The event data consists of all discrete events within a certain time period.

with asynchronous sensors that respond to brightness changes in a scene asynchronously and independently for each pixel, as shown in Fig. S2. Hence, the event data is a variable data-rate sequence of digital "events", *i.e.*, $\mathcal{E} = \{\boldsymbol{e}_k\}_{k=1}^{N}$, where N depends on the number of brightness changes in the scene. The faster the brightness changes, the more events per second are generated. The event data reacts rapidly to visual stimuli because the events are timestamped at microsecond resolution and transmitted with less than a millisecond latency.

To accommodate the input requirements of deep neural networks, the event stream needs to be transformed into the corresponding representations, also known as event representation[1]. Injecting triggers into the original event data ensures that the effectiveness of the proposed *Event Trojan* is not compromised by various event representation methods. As Table 6 illustrates, our method maintains attacking effectiveness across different event representations. The representation schemes we consider in our work are listed as follows:

[1] https://github.com/LarryDong/event_representation

- **Event Frame (EF).** EF is a simple representation strategy that considers the polarity (positive / none / negative) within the event data to set the pixel value (+1 / 0 / -1) in the images [1,2]. Furthermore, some variant versions [3] convert events by counting events or accumulating polarity pixel-wise into an image compatible with image-based vision models.
- **Time Surface (TS).** A TS representation [4] is also a 2D image where each pixel stores a single time value, *e.g.*, the time stamp of the last event at the selected pixel address. Thus, the event stream is converted into an image where only the most recent recorded timestamps at each pixel position are taken into account. It can be formulated as:

$$TS(x, y) = p \times \exp^{-(t_{max}-t)/\tau}, \tag{1}$$

  where τ is a time constant.
- **Voxel Grid (VG).** VG [5] is a space-time (3D) histogram of events, where each voxel represents a particular pixel and time interval. This representation preserves better the temporal information of the events by avoiding collapsing them on a 2D representation. The VG representation can be generated by:

$$\begin{aligned} V(x, y, t) &= \Sigma p_k \phi(x - x_k)\phi(y - y_k)\phi(t - t_k^*), \\ \phi(a) &= max(0, 1 - |a|), \\ t_k^* &= (B - 1)(t_k - t_1)/(t_N - t_1), \end{aligned} \tag{2}$$

  where B bins are used to discretize the time dimension and N denotes the length of a set of input events.
- **Tencode.** Tencode [6] considers both polarities and timestamps of the event stream to conduct the event representation. A temporal resolution Δt is defined to discretize the normalized time stamps in order to produce a three-channel frame I by:

$$\begin{aligned} I[x, y, :] &= (255, \frac{255 * (t_{max} - t)}{\Delta t}, 0) \leftarrow (x, y, t, +1), \\ I[x, y, :] &= (0, \frac{255 * (t_{max} - t)}{\Delta t}, 255) \leftarrow (x, y, t, +1), \end{aligned} \tag{3}$$

  where t_{max} represents the timestamp of the latest event in the temporal resolution Δt

10 Detailed experimental results

Table S1 shows the detailed quantitative results of each classifier shown in Fig. 4 and Fig. 5 of the main paper, respectively. It's clear that the mutable trigger achieves better attacking performance than the immutable trigger in almost all cases on two public datasets. On the other hand, these victim models achieve better performance on the N-Cars dataset [10] than that on the N-Caltech101

dataset [9], primarily because N-Cars [10] has a larger number of training samples and fewer categories. On Transformer-based models, ViTs [11] perform worst because the extracted sequence features may not adequately satisfy the downstream tasks especially when the event data contains much background activity noise and fewer training samples (N-Caltech101 [9]). Due to the fact that poisoning the event representations to initiate backdoor attacks is impossible in real-world application scenarios, we haven't conducted more explorations about representation triggers in the following experiments. Only the classical backdoor method: BadNets [14] and the latest work: FIBA [15] are chosen in our experiments (Table 1 in the main paper).

Table S1: Quantitative results of the immutable trigger and mutable trigger imposed on 22 classifiers on the N-Caltech101 [9] and N-Cars [10] datasets, respectively.

	N-Caltech101 [9]				N-Cars [10]			
	Immutable Trigger		Mutable Trigger		Immutable Trigger		Mutable Trigger	
	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR
ResNet-18 [16]	0.8561	0.9673	0.8621	0.9971	0.9223	0.9967	0.9272	1.0000
ResNet-34 [16]	0.8698	0.7557	0.8598	0.9741	0.9190	0.9974	0.9279	0.9934
ResNet-50 [16]	0.8572	0.8194	0.8443	0.9747	0.9281	0.9933	0.9176	0.9981
ResNet-101 [16]	0.8578	0.9954	0.8534	0.9954	0.9159	0.9985	0.9132	0.9725
ResNet-152 [16]	0.8446	0.9839	0.8218	0.9885	0.9144	0.9859	0.9374	0.9949
VGG-16 [17]	0.7064	0.1812	0.8526	0.9765	0.9211	0.9970	0.9293	1.0000
VGG-19 [17]	0.6766	0.1692	0.8521	0.9719	0.4893	1.0000	0.9280	1.0000
EfficientNet-B0 [18]	0.8589	0.8630	0.8448	0.9443	0.9453	0.9964	0.9395	0.9933
EfficientNet-B1 [18]	0.8704	0.9386	0.8586	0.9644	0.9391	0.9988	0.9402	0.9972
EfficientNet-B2 [18]	0.8607	0.8291	0.8529	0.9718	0.9347	0.9988	0.9219	0.9908
EfficientNet-B3 [18]	0.8876	0.9828	0.8747	0.9868	0.9538	0.9993	0.9434	0.9904
EfficientNet-B4 [18]	0.8704	0.9025	0.8761	0.9718	0.9391	0.9926	0.9454	0.9955
Inception-v3 [19]	0.6852	0.6451	0.8477	0.6891	0.9495	0.9972	0.9327	0.9909
ViT-S [20]	0.5086	0.1474	0.4731	0.8773	0.8453	0.9729	0.8717	1.0000
ViT-B [20]	0.4851	0.0401	0.5189	0.9943	0.8113	0.9584	0.8815	1.0000
ViT-L [20]	0.4908	0.0860	0.4736	0.9874	0.8542	0.9807	0.8809	0.9987
Swin-S [21]	0.7494	0.2161	0.8899	0.9994	0.7974	0.5091	0.9476	1.0000
Swin-B [21]	0.7655	0.1799	0.9203	0.9977	0.7357	0.4105	0.9457	1.0000
Swin-L [21]	0.6247	0.3115	0.9203	0.9983	0.7981	0.5186	0.9536	1.0000
DeiT-S [22]	0.4868	0.0860	0.6640	1.0000	0.8532	0.9845	0.9030	0.9991
DeiT-B [22]	0.4300	0.1067	0.8165	1.0000	0.8280	0.9721	0.8865	0.9997
DeiT-L [22]	0.7425	0.1508	0.8773	1.0000	0.8641	0.9829	0.8978	0.9995

11 Visualization of triggers

Fig. S3 and Fig. S4 show more visualization examples of the benign event, poisoned event, and corresponding triggers. In Fig. S3, it's clear that the stealthiness of the poisoned event compromised by representation trigger (R. trigger) is lower than our *Event Trojan*. Notably, in BadNets [14], the noticeable white

patch in the top-left is easily detectable by users. FIBA [15] embeds a random image into the frequency domain of the event representations, yielding better performance than BadNets. However, it is still quite noticeable when compared to benign events. Our *Event Trojan* is designed to inject triggers directly into the event data, thereby avoiding abnormal anomalies in the corresponding event representations. Fig. S4 presents some point sets of the poisoned event data compromised by our two types of triggers. The mutable trigger exhibits a more stealthy pattern than the immutable trigger.

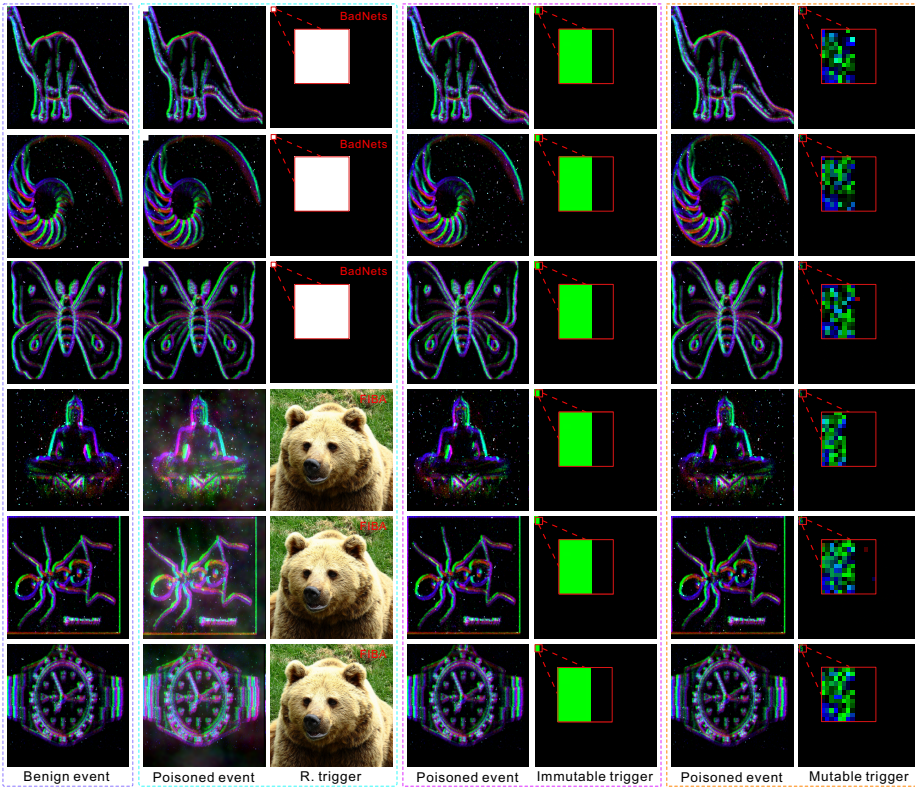

Fig. S3: From left to right, we show benign events, poisoned events with representation trigger (R. trigger), poisoned events with immutable trigger, and poisoned events with mutable trigger, respectively. Trigger details are zoomed in on the red square for better visibility. For the representation trigger, we show two types of triggers generated by BadNets [14] (first 3 rows) and FIBA [15] (last 3 rows), respectively.

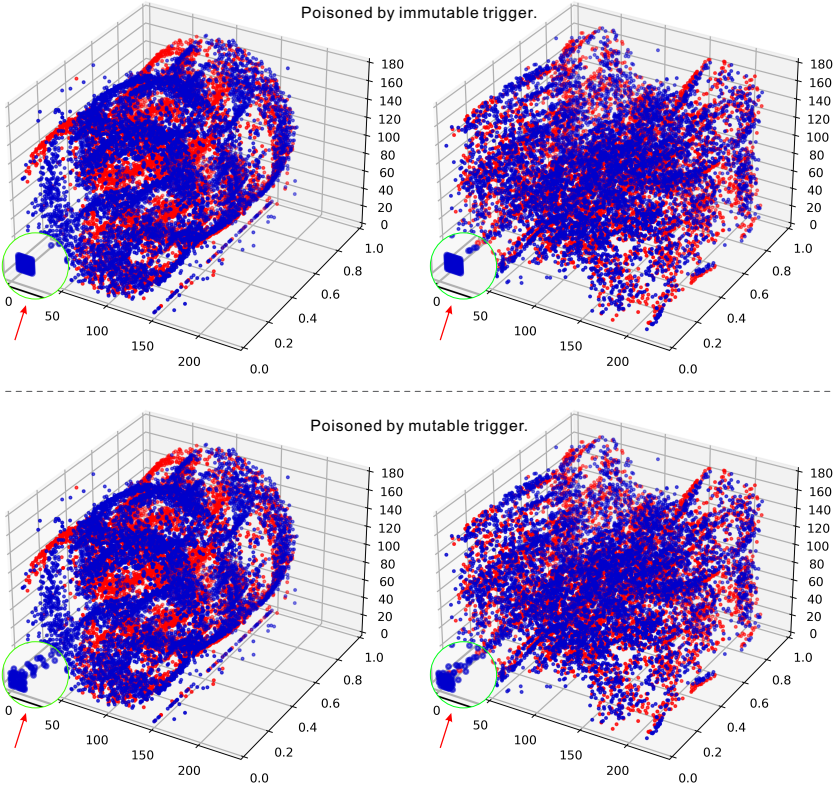

Fig. S4: Point sets of triggered samples poisoned by our immutable and mutable triggers. For better visualization, we normalize these event data in the time dimension. Details are zoomed in on the green circle ◯. Blue means the polarity $p = 1.0$ while red denotes the $p = -1.0$.

References

1. Liu M, Delbruck T. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In BMVC, 2018. 3
2. Rebecq H, Horstschaefer T, Scaramuzza D. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. 2017. 3
3. Maqueda A I, Loquercio A, Gallego G, et al. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018: 5419-5427. 3
4. Lagorce X, Orchard G, Galluppi F, et al. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2016, 39(7): 1346-1359. 3
5. Zhu A Z, Yuan L, Chaney K, et al. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 989-997. 3
6. Huang Z, Sun L, Zhao C, et al. EventPoint: Self-Supervised Interest Point Detection and Description for Event-based Camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023: 5396-5405. 3
7. Schaefer S, Gehrig D, Scaramuzza D. Aegnn: Asynchronous event-based graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022: 12371-12381.
8. Gehrig D, Loquercio A, Derpanis K G, et al. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019: 5633-5643.
9. Orchard G, Jayawant A, Cohen G K, et al. Converting static image datasets to spiking neuromorphic datasets using saccades. Frontiers in neuroscience, 2015, 9: 437. 4
10. Sironi A, Brambilla M, Bourdis N, et al. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018: 1731-1740. 3, 4
11. Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 4
12. Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2021: 10012-10022.
13. Touvron H, Cord M, Douze M, et al. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*. PMLR, 2021: 10347-10357.
14. Gu T, Dolan-Gavitt B, Garg S. Badnets: Identifying vulnerabilities in the machine learning model supply chain[J]. arXiv preprint arXiv:1708.06733, 2017. 4, 5
15. Feng Y, Ma B, Zhang J, et al. Fiba: Frequency-injection based backdoor attack in medical image analysis[C] In Proc. CVPR. 2022: 20876-20885. 4, 5
16. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016: 770-778. 4
17. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. 4
18. Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 2019: 6105-6114. 4

19. Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016: 2818-2826. 4
20. Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16×16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 4
21. Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2021: 10012-10022. 4
22. Touvron H, Cord M, Douze M, et al. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*. PMLR, 2021: 10347-10357. 4