Event Trojan: Asynchronous Event-Based Backdoor Attacks

Ruofei Wang[1], Qing Guo[2], Haoliang Li[3], and Renjie Wan[1(✉)]

[1] Department of Computer Science, Hong Kong Baptist University, Kowloon, Hong Kong
ruofei@life.hkbu.edu.hk, renjiewan@hkbu.edu.hk
[2] IHPC and CFAR, Agency for Science, Technology and Research (A*STAR), Singapore, Singapore
guo_qing@cfar.a-star.edu.sg
[3] Department of Electrical Engineering, City University of Hong Kong, Kowloon, Hong Kong
haoliang.li@cityu.edu.hk

Abstract. As asynchronous event data is more frequently engaged in various vision tasks, the risk of backdoor attacks becomes more evident. However, research into the potential risk associated with backdoor attacks in asynchronous event data has been scarce, leaving related tasks vulnerable to potential threats. This paper has uncovered the possibility of directly poisoning event data streams by proposing *Event Trojan* framework, including two kinds of triggers, *i.e.*, immutable and mutable triggers. Specifically, our two types of event triggers are based on a sequence of simulated event spikes, which can be easily incorporated into any event stream to initiate backdoor attacks. Additionally, for the mutable trigger, we design an adaptive learning mechanism to maximize its aggressiveness. To improve the stealthiness, we introduce a novel loss function that constrains the generated contents of mutable triggers, minimizing the difference between triggers and original events while maintaining effectiveness. Extensive experiments on public event datasets show the effectiveness of the proposed backdoor triggers. We hope that this paper can draw greater attention to the potential threats posed by backdoor attacks on event-based tasks.

Keywords: Backdoor attack · Event data · Event Trojan · Immutable trigger · Mutable trigger

1 Introduction

Event data is known for its exceptional capacity to capture fast-moving objects [11]. By converting asynchronous event data from *variable data-rate*

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-72667-5_18.

A. Leonardis et al. (Eds.): ECCV 2024, LNCS 15065, pp. 315–332, 2025.
https://doi.org/10.1007/978-3-031-72667-5_18

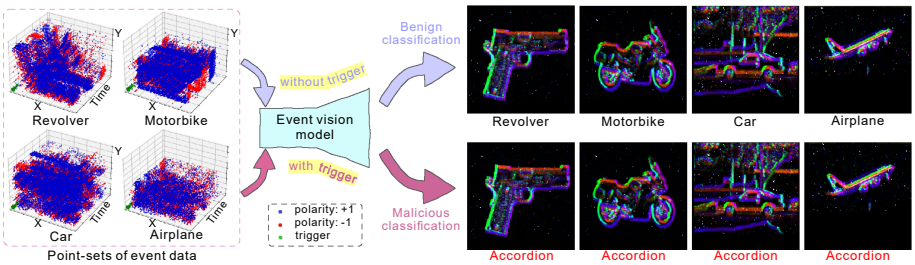

Fig. 1. Event data consists of a large number of asynchronous events, which can be manipulated to inject malicious triggers with high stealthiness, as illustrated by the green points in various point sets. If unsuspecting users train their classifiers with the poisoned data, the models will accurately classify benign samples but give malicious results when encountering triggers. The right images are rendered by EST [12]. (Color figure online)

sequences into *image-like representations*, it becomes compatible with existing deep learning frameworks used in various vision tasks, such as autonomous driving [28], object tracking [5], surveillance and monitoring [24], object/gesture recognition [31], *etc.*. However, the potential risk of backdoor attacks via event data becomes significantly evident when made compatible with deep networks.

Backdoor attacks embed triggers into original data to control the model's responses and are known for their simplicity and harmfulness [4,13]. Typical pipeline of backdoor attacks is poisoning the training data to install a malicious backdoor and then activating it by injecting the trigger into test samples during the inference phase [22]. A successful backdoor attack to its desired data should ensure that the trigger can effectively undermine the performance of the downstream models while keeping the correct prediction on benign samples [22], as shown in Fig. 1. Besides, the injected trigger should keep high stealthiness to avoid being discovered by users.

Different from conventional images, asynchronous event data consists of a variety of asynchronous events, as illustrated by the point sets in Fig. 1. For compatibility with existing deep networks, event data needs to be converted into image-like representations to serve as the inputs of deep networks [12,15,20,25,53]. A typical solution is to adopt a representation module with different task-specific models for classification [12], recognition [55], segmentation [42], *etc.*. Once the image-like representation is accessed, it can be directly injected with malicious triggers to initiate backdoor attacks for downstream models using existing image backdoor approaches [10,13]. However, as the image-like representation module and the downstream task-specific models are usually tightly bounded [36], attackers cannot get this module to launch the attack. Furthermore, since the original event streams still retain their original contents, any backdoors introduced into the event representation become invalid if the representation is reconstructed from the unaltered event streams.

As shown in Fig. 1, malicious attackers could devise highly stealthy new trigger patterns exploiting the unique characteristics of event data. Since the event data stream is hard for humans to perceive, data users may find it challenging to discern subtle alterations within the data stream. If attackers succeed in embedding harmful triggers into the event stream that naturally exhibit significant stealth, these triggers would be highly concealed and undetectable. Given the extensive use of public datasets in research and industrial applications, such malicious triggers with high stealthiness could lead to catastrophic outcomes.

In this paper, we propose *Event Trojan*, which injects triggers into the event data to enable backdoor attacks with high stealthiness and effective attack capability. An event, the basic unit of event data, consists of x-y coordinates, timestamps, and polarities. Therefore, a feasible solution is to craft the event trigger with multiple events based on predefined spatial coordinates, timestamps, and polarities, *i.e.*, immutable trigger. Then, the attackers only need to inject this trigger into event streams to conduct backdoor attacks for victim models. Although this simple trigger can effectively impair the performance of numerous event-based victim models, fixed settings of the immutable trigger lead to limited generalization ability in various cases. This is due to neglecting the event distribution in the original data. So, we propose learning the trigger from the original event data to ensure the trigger has adaptive content capable of poisoning various event data, *i.e.*, mutable trigger. Meanwhile, we design a novel loss function to optimize this trigger for high stealthiness and effectiveness. As displayed in Fig. 1, the mutable trigger (green points) shows a more realistic event form. Extensive experiments demonstrate that our proposed *Event Trojan* can easily inject triggers into the asynchronous event data and initiate effective backdoor attacks, even when defended by state-of-the-art defense methods.

Through the introduction of the *Event Trojan*, we uncover the potential dangers posed by backdoor attacks on event vision tasks and aim to increase awareness of this risk. Our contribution can be concluded as follows:

- We investigate the execution of backdoor attacks using asynchronous event data to raise awareness about the security concerns associated with event-based deep learning models.
- We propose *Event Trojan* to directly poison the event stream by injecting malicious events generated by considering multidimensional properties.
- An adaptive approach is introduced to make the injected trigger with adaptive time stamps that can maximize the attacking effectiveness.

2 Related Work

2.1 Event Data

Event data-based learning has gained increasing focus due to the advantages of the bio-inspired sensor, the event camera, which captures moving objects with high dynamic range and temporal resolution, low time latency and power consumption [1,12,37,41,55]. Gehrig *et al.* [12] design a grid-based representation

that transfers the event data stream to image-like representations, enabling many state-of-the-art vision models that can be easily worked on the event streams. Schaefer *et al.* [37] propose an asynchronous event-based graph neural network, which treats the events as temporally evolving graphs to avoid sacrificing the sparsity and high temporal resolution. Sun *et al.* [43] propose an unsupervised domain adaption method for semantic segmentation on event data, which motivates the segmentor to learn semantic information from labeled images to unlabeled events. In addition, event-based studies achieve satisfactory performances in image deblurring [16,52], optical flow [39,53], object recognition [18,55], video reconstruction [35], stereo matching [50], *etc.*. Although event-based methods have drawn more attention from researchers recently, limited security studies on this topic have been conducted.

2.2 Backdoor Attack

The backdoor attack is a typical topic to study the vulnerability of deep models [13], which is very different from the adversarial attack in two terms: attack fuse and attack process [32]. Backdoor attacks, injecting a trigger into data samples to mislead the model outputting an attacker's desired label, have been extensively studied for the model security of 2D-image models [48], 3D point cloud networks [21], neural radiance fields [7], natural language processing networks [38], speech recognizers [19], *etc.*. Gu *et al.* [13] first studied the backdoor attack in the deep learning area, injecting a checkerboard pattern as the trigger to mislead the classifier to output a given label on the triggered data. Subsequently, Chen *et al.* [4] propose a physical instance-based backdoor attack model, which employs the daily used products as triggers to avoid human censorship. Besides, some people make efforts to explore novel trigger patterns to improve the stealthiness, such as object reflection [26], image structure [29,49] and frequency perturbations [10,23], *etc.*. Apart from the classification task, the backdoor attack is also studied in terms of semantic segmentation [10], object detection [3], video recognition [51], facial recognition [47], *etc.*. Although a wide range of backdoor attack methods have been proposed to examine security issues across various tasks, it is still impossible to directly poison asynchronous event data by existing methods to execute backdoor attacks. This is due to the asynchronous property of event data, which only records information about pixels with brightness changes exceeding a certain threshold.

3 Preliminary

3.1 Backdoor Attack

Given a dataset $\mathcal{D} = \{d_i, l_i\}_{i=1}^{N}$, where d_i and l_i indicate the input data and the corresponding label. The backdoor attack aims to learn a mapping function: $f_\theta(d_i) \rightarrow l_i$ while changing this mapping to $f_\theta(d_i) \rightarrow c$ if d_i contains a trigger injected by $T(d_i)$. $f_\theta(\cdot)$ is a deep model with its learnable parameters θ. c is the

attacker-chosen label, which is employed to evaluate the attack effectiveness [6]. For training a backdoor model, attackers first need to poison some input data with a poison ratio ρ and then train the model with both benign and poisoned samples. Ultimately, this model can output accurate predictions on the benign samples while giving malicious outputs (*e.g.*, c) when the attacker injects the designed triggers into input data [23].

3.2 Background of Event Data

Event data consists of a variety of individual events, recorded as:

$$\mathcal{E} = \{\boldsymbol{e}_k\}_{k=1}^{N} = \{(x_k, y_k, t_k, p_k)\}_{k=1}^{N}, \tag{1}$$

where (x_k, y_k, t_k, p_k) indicates the x and y direction coordinates, time stamp, and polarity of a single activated event. N is the length of the event stream $\mathcal{E}$ [17]. An event, $\boldsymbol{e}_k$, has occurred when the variation of the log brightness at each pixel exceeds the threshold σ, *i.e.*, $|\log(x_k, y_k, t_k) - \log(x_k, y_k, t_{k-1})| > \sigma$ (see Fig. 2 (a)). If an event is activated, the polarity $p_k = 1.0$ when the difference between bi-temporal pixels is higher than $+\sigma$. Otherwise, p_k is set to -1.0.

Although event data significantly differs from traditional images, by transforming it into image-like representations, they can be made compatible with prevalent vision models that take images as input [2,33,36]. Many studies on event representation have been conducted recently. For example, Event Spike Tensor (EST) [12], a popular event representation method, employs differentiable kernel convolution and quantization layers to transfer the event to grid representations considering both time stamp and polarity:

$$V_{\pm}(x_w, y_h, t_n) = \sum_{e_k \in \mathcal{E}} f_{\pm}(x_k, y_k, t_k) \times \delta(x_w - x_k, y_h - y_k, t_n - t_k), \tag{2}$$

where $x_w \in \{0, 1, ..., W-1\}$, $y_h \in \{0, 1, ..., H-1\}$, $t_n \in \{t_0, t_0 + \Delta t, ..., t_0 + B\Delta t\}$, t_0 denotes the first time stamp, Δt denotes the bin size, and B indicates the number of temporal bins, $\pm$ means the two kinds of polarities. W and H are the width and height of event data, respectively. $\delta(x, y, t) = \nabla(x, y) max(0, 1 - |\frac{t}{\Delta t}|)$, where $\nabla(\cdot)$ is an indicator function.

Backdoor Attack on Event Representation. With the asynchronous event data converted into image-like representations, attackers can simply embed the image backdoor triggers as features to directly poison those representations (*e.g.*, FIBA [10] shown in Fig. 3) to initiate backdoor attacks. The overall process can be described as:

$$f_\theta(\boldsymbol{\mathcal{R}}_\omega(\mathcal{E})) \rightarrow l, \quad f_\theta(T(\boldsymbol{\mathcal{R}}_\omega(\mathcal{E}))) \rightarrow c, \tag{3}$$

where $\boldsymbol{\mathcal{R}}_\omega(\cdot)$ denotes the module for converting the event stream to image-like representations. Generally, an event vision model $F_{\{\theta,\omega\}}$ consists of both representation module $\boldsymbol{\mathcal{R}}_\omega$ and task-specific model f_θ.

However, due to the close integration of the event representation module with downstream task-specific models, attackers typically cannot access this event

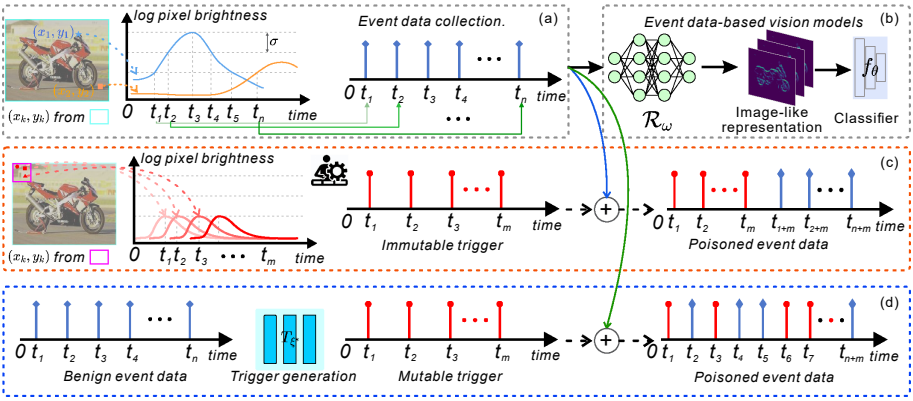

Fig. 2. The pipeline of the event data-based backdoor attacks. (a) The principle of event activation: events are generated when there are relative changes in brightness that exceed a threshold σ. (b) The flowchart of vision models based on event data. Each event stream needs to be first converted to an image-like representation by $\boldsymbol{\mathcal{R}}_\omega(\cdot)$ [12]. Generating poisoned samples by the immutable trigger (c) and mutable trigger (d), respectively. $T_{\xi^*}(\cdot)$ is the mutable trigger generator with its best parameters ξ^*. $\bigoplus$ indicates the concatenation operation.

representation. Consequently, poisoning the event representation becomes a less feasible threat operation for event vision tasks. Since event vision tasks begin with using the event stream as input, the data transmitted and utilized throughout the process is always the original event data. Malicious attackers are more likely to access this original event data during its transmission. Therefore, compared to compromising the event representation, directly poisoning the original event data presents a higher value.

4 Methodology

We introduce backdoor attacks into the context of asynchronous event data through a more practical strategy, *i.e.*, initiating backdoor attacks by poisoning the event data:

$$F_{\{\theta,\omega\}}(\mathcal{E}) \to l, \quad F_{\{\theta,\omega\}}(T(\mathcal{E})) \to c, \tag{4}$$

where $F_{\{\theta,\omega\}}$ denotes the event vision model with learned parameters, $T(\cdot)$ indicates the trigger injection function. c is the attacker-chosen label. We first discuss the threat model to event-based tasks and then introduce immutable and mutable triggers in the following sections to reveal the possible approaches for backdoor attacks on event-based vision models

4.1 Threat Model

Attacker's Capability. In practice, attackers have no ability to control the training details of event-based models (*e.g.*, model structure, loss function, *etc.*),

while accessing some training data is allowed. During inference, attackers can only access the original event data without any rights to manipulate the inference process, without any information about the event representation methods.

Attacker's Goal. The attacker aims to create a backdoored event-based model that incorporates a stealthy backdoor. This backdoor would be activated when a specific pattern is injected, resulting in the prediction that is predetermined by the attacker. Generally, attackers hope that the backdoor can be activated under any circumstances and the injected trigger won't be discovered users, *i.e.*, high *effectiveness* and *stealthiness*.

4.2 Immutable Trigger

The essence of the immutable trigger lies in the deliberate placement of malicious events at particular spatial locations and time stamps within various event data streams. After injecting these events, they can maintain consistent spatial positions and time stamps across different event streams (see Fig. 1 (d)). Given that the spatial dimension dictates the shape and the temporal dimension influences the pixel values in the event representation, these injected malicious events manifest as identical patterns in the representations. As shown in the 4^{th} column of Fig. 3, our immutable trigger still can show identical patterns across different asynchronous event data. We synthesize the immutable trigger manually with (x, y) coordinates sampled from a predefined region, timestamp by α, and polarity by β. Then, we inject it into the original event stream and modify the label as the attacker-desired target. Based on this strategy, we can generate more triggered samples according to the poison ratio ρ to train a victim model.

4.3 Mutable Trigger

The immutable trigger contaminates various asynchronous event data using fixed settings in dimensions of coordinates, timestamps, and polarities. Its fixed nature may not adequately address the distinct characteristics of different event data, potentially diminishing its effectiveness in backdoor attacks. The more malicious triggers should be designed based on the internal patterns of the original events. Thus, we introduce a mutable trigger pattern incorporating timing variations to better adapt to diverse event data.

As shown in the last column of Fig. 3, the mutable trigger possesses two characteristics: ❶ The malicious events inserted across various asynchronous event data streams possess identical spatial values, ensuring that the trigger patterns maintain the same shapes within the image-like representation; ❷ The inserted events are given adaptive time stamps (see the event trigger in Fig. 1, leading to trigger patterns with unique pixel values in the image-like representation. The entire methodology for embedding malicious triggers with adaptive time stamps is depicted in Algorithm 1. In this process, several events are strategically placed at time stamps deduced by a malicious events injector, referred to as $T_\xi(\cdot)$. First, we randomly sample m time stamps from the original event as the input of $T_\xi(\cdot)$,

Algorithm 1: Backdoor Attack via Mutable Trigger

Input: Classifier $f_\theta(\cdot)$, Trigger injector $T_\xi(\cdot)$, attacker-chosen label c, event Training dataset $\mathcal{E}_{\text{train}} = \{< \mathcal{E}_k, l_k >\}_{k=1}^{N}$, batch size b, trigger size m, learning rate γ_f and γ_T, and Maximum iteration number $MaxIters$, balance weights α and β, CrossEntropy loss function $\mathcal{L}$.

Output: $f_{\theta^*}(\cdot)$, $\boldsymbol{\mathcal{R}}_{\omega^*}(\cdot)$, and $T_{\xi^*}(\cdot)$.

Initialize θ and ξ, $\theta^* \leftarrow \theta, \xi^* \leftarrow \xi$;

Function MutableT$(\mathcal{E}, T_\xi)$**:**

  Sample m time stamps $\mathbf{t} = \{t_i\}_{i=1}^{m}$ from $\mathcal{E}$;

  Build our mutable trigger $\mathcal{T}$ with poisoned time stamps $T_\xi(\mathbf{t})$;

  Inject the mutable trigger $\mathcal{T}$ into $\mathcal{E}$ to generate poisoned $\mathcal{E}'$;

  return $\mathcal{E}'$;

End function;

for $i = 1$ to $MaxIters$ **do**

  Sample minibatch $< \mathcal{E}, l >$ from $\mathcal{E}_{train}$;

  Sample poisoned event $\mathcal{E}' = \textbf{MutableT}(\mathcal{E}, T_\xi)$;

  $\theta \leftarrow \theta - \gamma_f \partial_\theta (\mathcal{L}(f_\theta(\boldsymbol{\mathcal{R}}_\omega(\mathcal{E})), l) + \mathcal{L}(f_\theta(\boldsymbol{\mathcal{R}}_\omega(\mathcal{E}')), c))$;

  $\omega \leftarrow \omega - \gamma_f \partial_\omega (\mathcal{L}(f_\theta(\boldsymbol{\mathcal{R}}_\omega(\mathcal{E})), l) + \mathcal{L}(f_\theta(\boldsymbol{\mathcal{R}}_\omega(\mathcal{E}')), c))$;

  $\xi \leftarrow \xi - \gamma_T \partial_\xi \mathcal{L}_T(5)(T_\xi(\mathbf{t}), \mathbf{t})$;

  $\theta^* \leftarrow \theta,\ \omega^* \leftarrow \omega,\ \xi^* \leftarrow \xi$ if $i\%(len(\mathcal{E}_{train})//b)) = 0$;

which can generate a trigger $\mathcal{T}$ using the predicted malicious time stamps. Then, we inject this trigger into the original event and modify the corresponding labels according to the attacker's targets. Finally, we train the deep classifiers and our trigger injector $T_\xi(\cdot)$ jointly to encourage the produced triggers that best suit the classifiers. This scheme can utilize the classifier to guide the injector $T_\xi(\cdot)$ to learn the adaptive triggers for different events. We design a trigger optimization loss function to ensure that $T_\xi(\cdot)$ can learn the unique characteristics along the assigned dimensions. Note that our trigger generator is optimized with the classifier only during the training phase; this does not imply that the generator is tied to the classifier during inference.

Since the event stream consists of a series of individual and discrete events, we propose measuring the cosine similarity, expectation, and variance to identify the most malicious patterns from the original event. For effectiveness, we need to ensure that the $T_\xi(\cdot)$ can generate poisoned time stamps that have a distinct pattern with the original event data. So, we reduce the cosine similarity between the poisoned time stamps and benign counterparts to improve the difference between trigger and clean data. However, solely focusing on maximizing the difference between the malicious triggers and the original event data may cause the generated events to deviate significantly from the distribution of benign data, thereby impacting the performance of classifiers on clean data. We attempt to push the expectation and variance of poisoned time stamps to those of benign samples, which may encourage the $T_\xi(\cdot)$ to generate the time stamps as similar

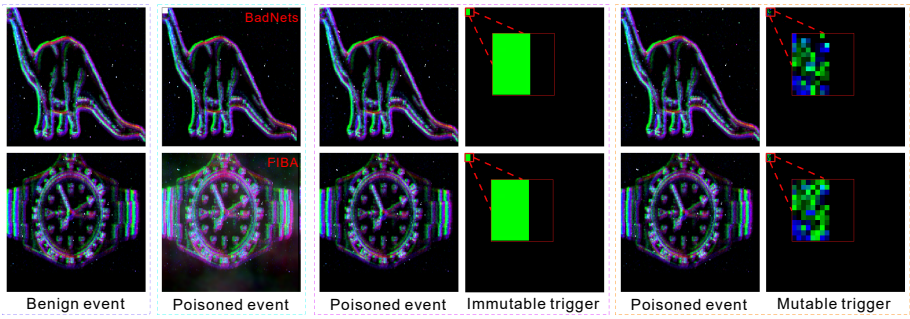

Fig. 3. Visualization results corresponding to the benign events, poisoned events, and the corresponding triggers. Trigger details are zoomed in on the red square for better visibility. For the representation trigger, we show two types of triggers in 2nd column generated by BadNets [13] (1st row) and FIBA [10] (2nd row), respectively. (Color figure online)

to the original inputs. The trigger generation loss is then formulated as:

$$\mathcal{L}_T = \lambda_1 \frac{T_\xi(\mathbf{t}) \cdot \mathbf{t}}{||T_\xi(\mathbf{t})|| \times ||\mathbf{t}||} + \lambda_2 \ \psi(T_\xi(\mathbf{t}), \mathbf{t}), \tag{5}$$

where $\mathbf{t} = \{t_i\}_{i=1}^{m}$, indicating the sampled m time stamps from the benign event data. $T_\xi(\mathbf{t})$ denotes the generated malicious time stamps. $\psi(\cdot)$ involves calculating the square difference between the malicious time stamps and clean time stamps in terms of the expectation and variance, respectively. The detailed training process is shown in line $8 \sim 14$ of Algorithm 1.

4.4 Implementation Details

We implement our method using Pytorch. The trigger injector $T_\xi(\cdot)$ is built by the Multi-Layer Perceptron (MLP) with 5 layers, each having 64 channels. The length of synthesized events, m, and poison ratio, ρ, are set to 100 and 0.1. For the immutable triggers, we set the time stamp, α, and polarity, β, as 10^{-2} and 1.0, respectively. For mutable triggers, we sample the time stamps randomly and set the balance weights λ_1 and λ_2 are 1 and 2, respectively. We use the SGD optimizer with learning rate 10^{-4} and momentum 0.9 to train classifiers and the trigger generator $\mathcal{T}_\xi(\cdot)$. The learning rate is decreased by exponential scheduler with gamma 0.5. All backdoored methods are trained for 60 epochs while finetuning for defense by 20 epochs.

5 Experiments

5.1 Setup

Dataset. To validate the effectiveness of our methods, We use the N-Caltech101 dataset [30] and the N-Cars dataset [41] in our evaluation. N-Caltech101 [30] is an

Table 1. Quantitative comparison results of different triggers imposed on event data-based deep models. We show the R. triggers (Representation triggers) obtained by BadNets [13] and FIBA [10] in Fig. 3.

Dataset	Victim Model	R. trigger BadNets		R. trigger FIBA		Immutable trigger		Mutable trigger	
		CDA↑	ASR↑	CDA↑	ASR↑	CDA↑	ASR↑	CDA↑	ASR↑
N-Caltech101 [30]	ResNet-18 [14]	57.24	0.0	82.47	43.39	85.61	96.73	86.21	99.71
	VGG-16 [40]	65.86	100.0	67.82	100.0	70.64	18.12	85.26	97.65
	Swin-S [27]	46.67	100.0	43.45	100.0	74.94	21.61	88.99	99.94
	ViT-S [8]	40.06	100.0	44.48	100.0	50.86	14.74	47.31	87.73
N-Cars [41]	ResNet-18 [14]	91.27	99.92	90.18	100.0	92.23	99.67	92.72	100.0
	VGG-16 [40]	91.83	100.0	91.98	100.0	92.11	99.70	92.93	100.0
	Swin-S [27]	84.91	100.0	90.98	100.0	79.74	50.91	94.76	100.0
	ViT-S [8]	84.73	100.0	84.53	100.0	84.53	97.29	87.17	100.0

event-based version of the frame-based Caltech101 dataset [9], which is obtained by affixing the ATIS sensor [34] to a motorized pan-tilt unit to record the moved Caltech101 examples. N-Caltech101 [30] consists of 4,356 training samples, 2,612 validating samples, and 1,741 testing samples in 101 classes. The amount of data varies greatly among different categories. N-Cars (Neuromorphic-Cars) [41] is a real-world event dataset for recognizing whether a car is present in a scene. It is recorded using an ATIS camera [34] that is mounted on a car. According to the partition in [37], the N-Cars dataset [41] includes 8392 training samples, 2462 validation samples, and 8608 testing samples in two classes.

Victim Model. For evaluating the effectiveness of the proposed methods comprehensively, we quantify the results of 22 popular classifiers with different network architectures (including ResNet [14], VGG [40], EfficientNet [45], Inception [44], ViT [8], Swin Transformer [27], DeiT [46]. All models are implemented by the official codes, with modifications made only to the input and output channels. EST [12] is selected as the event representation network.

Error Metric. We adopt the Attack Success Rate (ASR) and Clean Data Accuracy (CDA) to evaluate the effectiveness of the proposed methodology on both two datasets against different baselines. Specifically, ASR is the proportion of successfully attacked poison samples in the total poison examples, showing the effectiveness of the tested backdoor attackers. CDA is defined as the accuracy of testing on benign event data, which is used to evaluate the performance of backdoored models on untriggered data. Higher is better for both error metrics.

5.2 Evaluation

Representation Trigger. As shown in Fig. 3, we can use different representation triggers (an abnormal pixel block: BadNets [13] or a frequency perturbation: FIBA [10]) to poison the event representations. From Table 1, it can show that such a representation trigger can achieve a good attack success rate on

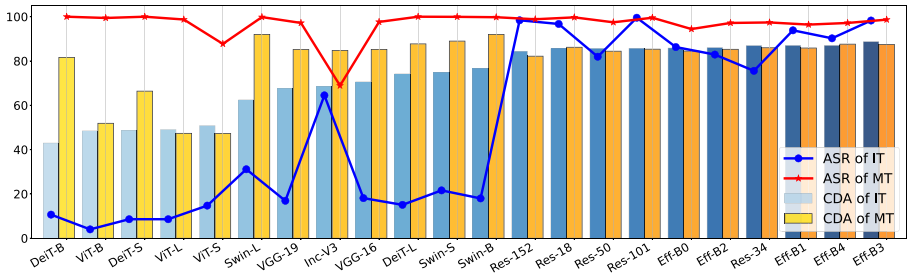

Fig. 4. Quantitative results about Immutable Trigger (IT) and Mutable Trigger (MT) evaluated by 22 deep classifiers on the event data from N-Caltech101 dataset [30]. The names of some baselines are abbreviated due to space limitation (Res: ResNet [14], Eff: EfficientNet [45], Inc: Inception [44]).

both two datasets. However, such a performance highly hinges on the image-level backdoor approaches. For example, BadNets [13] is unable to compromise ResNet-18 on the N-Caltech dataset because the small white block injected as a backdoor trigger is hard to detect by a lightweight model when processing various data with noises. FIBA [10] imposes some confusion for deep classifiers on clean data, resulting in low CDA. Furthermore, as discussed in Sect. 3.2, the event representation is inaccessible to attackers during the inference phase, which significantly undermines the effectiveness of image backdoor attack methods.

Immutable Trigger. In Fig. 3, we present the visualization results of the immutable triggers and the corresponding poisoned event data. The results demonstrate that the immutable trigger does not negatively impact the visualization of original event samples. As Fig. 4 shows, our immutable trigger successfully attacks most vision models on N-Caltech101 dataset [30], such as ResNet [14], EfficientNet [45], and Inception-V3 [44], without causing confusion on benign data. However, the other classifiers like VGG [40], ViT [8], and DeiT [46] fail to detect the injected triggers during the attack process. Detailed quantitative results are shown in Table 1. This discrepancy can be attributed to the specific characteristics of the N-Caltech101 dataset [30], which contains a significant amount of background noise and imbalanced data distribution. As a result, the fixed and unified immutable trigger may not be suitable for different event samples attacking these more deep classifiers.

On the N-Cars dataset [41], our immutable trigger achieves successful attacking performance in most cases. As shown in Fig. 5, the overall performance of each classifier on N-Cars [41] is better than on N-Caltech101 [30] due to its larger data scale. Additionally, we can find that the Transformer-based classifiers are slightly inferior to convolution-based models on the N-Cars dataset [41]. In summary, our immutable trigger successfully attacks the majority of classifiers with a high ASR, while maintaining the model's performance on the benign data.

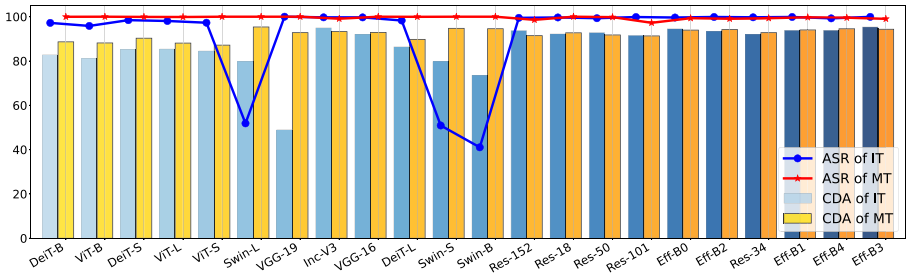

Fig. 5. Quantitative results about Immutable Trigger (IT) and Mutable Trigger (MT) evaluated by 22 deep classifiers on the event data from N-Cars dataset [41]. The names of some baselines are abbreviated due to space limitation (Res: ResNet [14], Eff: EfficientNet [45], Inc: Inception [44]).

Mutable Trigger. The last two columns of Fig. 3 present the visualization results of the mutable triggers and the corresponding poisoned event data, respectively. The mutable trigger has different pixel values across different event data and is less noticeable than the other two kinds of triggers. Table 1 presents the evaluation results of mutable triggers on the N-Caltech101 [30] and N-Cars datasets [41]. The findings indicate that the mutable trigger consistently outperforms the immutable trigger in terms of attack performance and clean data accuracy. Compared to representation triggers, our mutable trigger retains a strong attack capability and imposes less confusion on clean data.

Figure 4 and Fig. 5 show the quantitative results of each classifier with mutable triggers on the N-Caltech101 [30] and N-Cars [41] datasets. Compared to the immutable trigger, the mutable trigger achieves better attacking performance on most vision models, while keeping a high accuracy on the benign data. Only on the N-Caltech101 dataset [30], the mutable trigger does not achieve excellent attacking performance on the Inception-V3 [44]. This is mainly caused by the data scales and background noise contained in this dataset. However, this issue has been effectively resolved when users have a large number of training samples, such as the N-Cars dataset [41].

Table 2. Performance against backdoor defense method: Neural Polarizer [54].

Defense Method	R. trigger BadNets		R. trigger FIBA		Immutable trigger		Mutable trigger	
	CDA↑	ASR↑	CDA↑	ASR↑	CDA↑	ASR↑	CDA↑	ASR↑
NP (NeurIPS24 [54])	60.00	1.03	15.63	0.0	66.84	22.01	83.03	64.11

Table 3. Importance of the proposed loss function. w/o denotes the eliminated item in Eq. (5).

	w/o cos.	w/o $\psi(\cdot)$	$\mathcal{L}_T$
CDA	80.91	85.67	86.21
ASR	11.93	100.00	99.71

Table 4. Effectiveness of poisoning different dimensions of event data. $[\cdot]$ indicates the corresponding key in event data.

	$T(\mathcal{E}_{[t]})$	$T(\mathcal{E}_{[p]})$	$T(\mathcal{E}_{[x,y,t,p]})$
CDA	82.93	84.31	86.21
ASR	9.71	8.56	99.71

5.3 Ablation Studies

Backdoor Defense. To evaluate the robustness of our backdoor triggers, we adopt a state-of-the-art backdoor defense method: Neural Polarizer (NP) [54], to defend against each method on the N-Caltech dataset [30]. Neural polarizer is inspired by light polarization, which injects a new neural layer into the triggered model to filter out poisoned features. Detailed defense results are shown in Table 2, which demonstrates that we should draw greater attention to the potential risks posed by backdoor attacks on event-based models. Image backdoor attack methods inject triggers into the representations. These triggers are easily polarized since the benign features and poisoned features are separated. We inject triggers into the event data itself, where the benign and poisoned features are closely intertwined, preventing polarization.

Trigger Optimization Loss Function. To improve the effectiveness of injected triggers, we have designed a new loss (see Eq. (5)) for supervising the trigger generation. We conduct the ablation study about each component of Eq. (5) in Table 3. If we eliminate the cosine similarity between the poisoned timestamps and the original input (w/o cos.), it will be challenging for downstream models to detect the generated triggers since this term strengthens the attack ability of our trigger. Without calculating the square difference between two terms (w/o $\psi(\cdot)$), the mutable triggers are prone to be captured by downstream task models, but this also introduces some confusion on the benign samples.

Table 5. Influence caused by the size of the injected trigger. We set the $height \times width = m$ to represent the height and width of our triggers, where m is the length of synthesized events.

	1×10	5×5	10×10	20×20	30×30
CDA	0.8458	0.8435	0.8561	0.8681	0.8796
ASR	0.0820	0.0694	0.9673	0.9954	0.9994

Trigger Dimension. As we discussed before, poisoning the event data in a single dimension can also inject triggers successfully. However, chaotic distributions result in poor attacking performance. Now, we study the effectiveness of

Table 6. Experimental performance of the injected trigger under different event representations. And the time cost for each method to convert an event stream into the corresponding image-like representation.

	EST [12]	EF [25]	TS [20]	VG [53]	Tencode [15]
CDA	0.8561	0.8050	0.8016	0.8790	0.8050
ASR	0.9673	0.8830	0.8635	0.9977	0.9461
Time (s)	0.0013	0.3214	0.5102	0.3894	0.5938

this straightforward solution by poisoning the timestamps and polarities, respectively. Table 4 shows quantitative results of different trigger injection strategies tested by ResNet-18 on the N-Caltech dataset. It's clear that poison single dimension of event data cannot execute backdoor attacks successfully.

Trigger Size. Generally, trigger size plays a crucial role in determining the effectiveness and stealthiness of a backdoor attack method. An experiment has been conducted in Table 5 to show the correlation between the trigger size and their attacking effectiveness. From Table 5, the larger trigger size usually leads to higher values on CDA and ASR. However, when the trigger size increases, it also becomes obvious from Fig. 3. This is in line with observations on image-level backdoor attacks: a bigger trigger enhances effectiveness, but it also makes the trigger more noticeable. Considering its comprehensive performance, we select the small but effective size of 10×10 to design our triggers.

Event Representation. *Event Trojan* aims to embed triggers into the original data, enabling the proposed method to be effective after being converted by any event representation techniques. Since we have emphasized in the threat model that attackers have no ability to access the event representation modules. As depicted in Table 6, our approach has yielded impressive CDA and ASR results across various event representation methods such as Event Spike Tensor (EST) [12], Event Frame (EF) [25], Time Surface (TS) [20] Voxel Grid (VG) [53] and Tencode [15]. Considering the time consumption for event representations, we select the EST [12] as the event representation module in our experiments. Detailed results based on EST on more victim models are shown in Table 1.

Stealthiness. Event data is a type of multidimensional time-series data that are hardly perceptible to users. Meanwhile, image backdoor attacks (*e.g.*, BadNets [13], FIBA [10]) cannot poison the event data itself since they only inject the trigger into the corresponding representations (see Sec. 3.2). Hence, we cannot directly assess the stealthiness of various methods on the poisoned event data. A possible solution is to evaluate it by converting the event data into corresponding representations. Table 7 shows the stealthiness comparison of four kinds of poisoned event representations. Our triggers have better stealthiness than comparison methods, and the immutable trigger has a higher PSNR than the mutable trigger because of its fixed pattern.

Table 7. Stealthiness of the poisoned event representations.

	BadNets	FIBA	Immutable T.	Mutable T.
PSNR↑	39.772	27.769	75.064	65.453
SSIM↑	0.996	0.4586	1.000	1.000
LPIPS↓	0.005	0.0743	0.000	0.000

6 Conclusion

Our paper investigates the potential risks posed by backdoor attacks on event-based deep models. We propose the *Event Trojan* framework and have discussed various potential strategies for backdoor attacks and identified their pros and cons. Several designs are made to accommodate the designed trigger to maximize its attacking effectiveness. We further conduct thorough experiments to evaluate the proposed trigger injection strategies. From our experiments, while the multidimensional nature of event data makes it challenging to conduct backdoor attacks as usual, it does not indicate that users of event data can rest easy. Attackers are still capable of injecting harmful events to compromise downstream vision models. Moreover, since the current state-of-the-art defense method is ineffective against *Event Trojan* attacks, increased awareness of the security issues in event data-based models should be given.

Limitations and Future Work. This paper focuses on studying the security issues of event data-based deep neural networks against backdoor attacks. Extensive experiments are conducted on the event-based classification task to show that we should pay greater attention to the potential threat. In the future, we will study this issue caused by *Event Trojan* in more general event-based tasks.

Acknowledgement. This work was done at Renjie's Research Group at the Department of Computer Science of Hong Kong Baptist University. Renjie's Research Group is supported by the National Natural Science Foundation of China under Grant No. 62302415, Guangdong Basic and Applied Basic Research Foundation under Grant No. 2022A1515110692, 2024A1515012822, and the Blue Sky Research Fund of HKBU under Grant No. BSRF/21–22/16. It is also supported by the National Research Foundation, Singapore, and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-GC-2023-008), Career Development Fund (CDF) of Agency for Science, Technology and Research (A*STAR) (No.: C233312028), and National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative (No. DTC-RGC-04). This research is partially supported by the Changsha Technology Fund under its grant No. KH2304007.

References

1. Alonso, I., Murillo, A.C.: EV-SegNet: semantic segmentation for event-based cameras. In: Proceedings CVPRW (2019)
2. Berlincioni, L., et al.: Neuromorphic event-based facial expression recognition. In: Proceedings CVPR (2023)
3. Chan, S.H., Dong, Y., Zhu, J., Zhang, X., Zhou, J.: BadDet: backdoor attacks on object detection. In: Proceedings ECCV (2022)
4. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)
5. Delbruck, T., Lang, M.: Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor. Front. Neurosci. **7**, 223 (2013)
6. Doan, K., Lao, Y., Zhao, W., Li, P.: LIRA: learnable, imperceptible and robust backdoor attacks. In: Proceedings ICCV (2021)
7. Dong, W., Liu, J., Ke, Y., Chen, L., Sun, W., Pan, X.: Steganography for neural radiance fields by backdooring. arXiv preprint arXiv:2309.10503 (2023)
8. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. In: ICLR (2021). https://openreview.net/forum?id=YicbFdNTTy
9. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: Proceedings CVPRW (2004)
10. Feng, Y., Ma, B., Zhang, J., Zhao, S., Xia, Y., Tao, D.: FIBA: frequency-injection based backdoor attack in medical image analysis. In: Proceedings CVPR (2022)
11. Gallego, G., et al.: Event-based vision: a survey. IEEE TPAMI **44**(1), 154–180 (2020)
12. Gehrig, D., Loquercio, A., Derpanis, K.G., Scaramuzza, D.: End-to-end learning of representations for asynchronous event-based data. In: Proceedings ICCV (2019)
13. Gu, T., Dolan-Gavitt, B., Garg, S.: BadNets: identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the CVPR (2016)
15. Huang, Z., Sun, L., Zhao, C., Li, S., Su, S.: EventPoint: self-supervised interest point detection and description for event-based camera. In: Proceedings WACV (January 2023)
16. Jiang, Z., Zhang, Y., Zou, D., Ren, J., Lv, J., Liu, Y.: Learning event-based motion deblurring. In: Proceedings CVPR (2020)
17. Kim, H., Leutenegger, S., Davison, A.J.: Real-time 3D reconstruction and 6-DoF tracking with an event camera. In: Proceedings ECCV (2016)
18. Kim, J., Hwang, I., Kim, Y.M.: Ev-TTA: test-time adaptation for event-based object recognition. In: Proceedings CVPR (2022)
19. Koffas, S., Xu, J., Conti, M., Picek, S.: Can you hear it? backdoor attacks via ultrasonic triggers. In: Proceedings ACM Workshop WiseML, pp. 57–62 (2022)
20. Lagorce, X., Orchard, G., Galluppi, F., Shi, B.E., Benosman, R.B.: HOTS: a hierarchy of event-based time-surfaces for pattern recognition. IEEE TPAMI **39**(7), 1346–1359 (2016)
21. Li, X., et al.: PointBA: towards backdoor attacks in 3D point cloud. In: Proceedings ICCV (2021)
22. Li, Y., Jiang, Y., Li, Z., Xia, S.T.: Backdoor learning: a survey. TNNLS 1–18 (2022). https://doi.org/10.1109/TNNLS.2022.3182979

23. Li, Y., Li, Y., Wu, B., Li, L., He, R., Lyu, S.: Invisible backdoor attack with sample-specific triggers. In: Proceedings ICCV, pp. 16463–16472 (2021)
24. Litzenberger, M., et al.: Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor. In: 2006 IEEE intelligent transportation systems conference, pp. 653–658. IEEE (2006)
25. Liu, M., Delbruck, T.: Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In: Proceedings BMVC (2018)
26. Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: a natural backdoor attack on deep neural networks. In: Proceedings ECCV (2020)
27. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings ICCV (2021)
28. Maqueda, A.I., Loquercio, A., Gallego, G., García, N., Scaramuzza, D.: Event-based vision meets deep learning on steering prediction for self-driving cars. In: Proceedings CVPR, pp. 5419–5427 (2018)
29. Nguyen, T.A., Tran, A.T.: WaNet - imperceptible warping-based backdoor attack. In: ICLR (2021). https://openreview.net/forum?id=eEn8KTtJOx
30. Orchard, G., Jayawant, A., Cohen, G.K., Thakor, N.: Converting static image datasets to spiking neuromorphic datasets using saccades. Front. Neurosci. **9**, 437 (2015)
31. Orchard, G., Meyer, C., Etienne-Cummings, R., Posch, C., Thakor, N., Benosman, R.: HFirst: a temporal approach to object recognition. IEEE TPAMI **37**(10), 2028–2040 (2015)
32. Pan, Z., Mishra, P.: Backdoor attacks on bayesian neural networks using reverse distribution. arXiv preprint arXiv:2205.09167 (2022)
33. Perot, E., De Tournemire, P., Nitti, D., Masci, J., Sironi, A.: Learning to detect objects with a 1 megapixel event camera. NeurIPS (2020)
34. Posch, C., Matolin, D., Wohlgenannt, R.: A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. IEEE J. Solid-State Circuits **46**(1), 259–275 (2010)
35. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: Events-to-video: bringing modern computer vision to event cameras. In: Proceedings CVPR (2019)
36. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: High speed and high dynamic range video with an event camera. IEEE TPAMI **43**(6), 1964–1980 (2019)
37. Schaefer, S., Gehrig, D., Scaramuzza, D.: AEGNN: asynchronous event-based graph neural networks. In: Proceedings CVPR (2022)
38. Sheng, X., Han, Z., Li, P., Chang, X.: A survey on backdoor attack and defense in natural language processing. In: QRS (2022)
39. Shiba, S., Aoki, Y., Gallego, G.: Secrets of event-based optical flow. In: Proceedings ECCV (2022)
40. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
41. Sironi, A., Brambilla, M., Bourdis, N., Lagorce, X., Benosman, R.: HATS: histograms of averaged time surfaces for robust event-based object classification. In: Proceedings CVPR (2018)
42. Stoffregen, T., Gallego, G., Drummond, T., Kleeman, L., Scaramuzza, D.: Event-based motion segmentation by motion compensation. In: Proceedings ICCV (2019)
43. Sun, Z., Messikommer, N., Gehrig, D., Scaramuzza, D.: ESS: Learning event-based semantic segmentation from still images. In: Proceedings ECCV. Springer (2022). https://doi.org/10.1007/978-3-031-19830-4_20
44. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings CVPR (2016)

45. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: ICML (2019)
46. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021)
47. Wenger, E., Passananti, J., Bhagoji, A.N., Yao, Y., Zheng, H., Zhao, B.Y.: Backdoor attacks against deep learning systems in the physical world. In: Proceedings CVPR (2021)
48. Yu, Y., Wang, Y., Yang, W., Lu, S., Tan, Y.P., Kot, A.C.: Backdoor attacks against deep image compression via adaptive frequency trigger. In: Proceedings CVPR (2023)
49. Zhang, J., et al.: Poison ink: robust and invisible backdoor attack. IEEE TIP **31**, 5691–5705 (2022)
50. Zhang, K., et al.: Discrete time convolution for fast event-based stereo. In: Proceedings CVPR (2022)
51. Zhao, S., Ma, X., Zheng, X., Bailey, J., Chen, J., Jiang, Y.G.: Clean-label backdoor attacks on video recognition models. In: Proceedings CVPR (2020)
52. Zhou, C., Teng, M., Han, J., Xu, C., Shi, B.: DeLiEve-Net: deblurring low-light images with light streaks and local events. In: Proceedings ICCV (2021)
53. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: Proceedings CVPR (2019)
54. Zhu, M., Wei, S., Zha, H., Wu, B.: Neural polarizer: a lightweight and effective backdoor defense via purifying poisoned features. Adv. Neural Inf. Process. Syst. **36** (2024)
55. Zubić, N., Gehrig, D., Gehrig, M., Scaramuzza, D.: From chaos comes order: Ordering event representations for object recognition and detection. In: Proceedings ICCV (2023)