Asynchronous Event Error-Minimizing Noise for Safeguarding Event Dataset

Supplementary Material

Ruofei Wang[1] Peiqi Duan[2,3] Boxin Shi[2,3] Renjie Wan[1*]

[1]Department of Computer Science, Hong Kong Baptist University

[2]State Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University

[3]National Engineering Research Center of Visual Technology, School of Computer Science, Peking University

ruofei@life.hkbu.edu.hk, {duanqi0001, shiboxin}@pku.edu.cn, renjiewan@hkbu.edu.hk

6. Overview

- Sec. 7 discusses our **application scenario** again and shows the **difference** between image baselines and our unlearnable event streams.
- Sec. 8 illustrates the detailed explanation of our **algorithm**.
- Sec. 9 shows more details of our E^{2}MN and the corresponding **mixed** counterparts.
- Sec. 10 shows the exploration about the proposed **projection** strategy.
- Sec. 11 illustrates the details of five kinds of event pollution operations used in our experiments.
- Sec. 12 add more experiments on **event representations**, **time bins**, **adversarial attack strategies**, **naive baselines**, *etc*.
- Sec. 13 lists the **dataset** details for N-Caltech101, CIFAR10-DVS, DVS128 Gesture, and N-ImageNet.
- Sec. 14 depicts more **visualization** results to evaluate the imperceptibility of our E^{2}MN.
- Sec. 15 discusses the **social impact** of our UEvs.
- Sec. 16 lists our **future work**, including transferability evaluation, generation efficiency, and defense mechanism.

7. Our UEvs

We propose UEvs mainly focusing on preventing **unauthorized** event data usage. As shown in Fig. 5, the protector, *i.e.*, data owner, releases the unlearnable dataset for users, while only the authorized models can learn real semantic features from these data. The hacker's unauthorized models are prevented from learning. This mechanism effectively protects the interests of data owners and avoids the privacy leakage caused by data misuse. Fig. 5 shows the working scenario of our UEvs.

Unlearnable Examples (UEs) are proposed to prevent image dataset from unauthorized usage. Compared with UEs, UEvs show great challenges. 1) The image perturbation is directly optimized by deep models to ensure its effectiveness and imperceptibility. However, the event data cannot be directly input into deep models to generate the unlearnable version. 2) Event data shows the binary polarity and asynchronous nature that hinders the noise injection. This sparse property denotes that although we can represent the event stream via image-like features, the generated noise cannot be compatible with event data. 3) Different from image perturbation, event noise should include time stamps that ensure the unlearnable noise is closely aligned with the event data, enhancing imperceptibility.

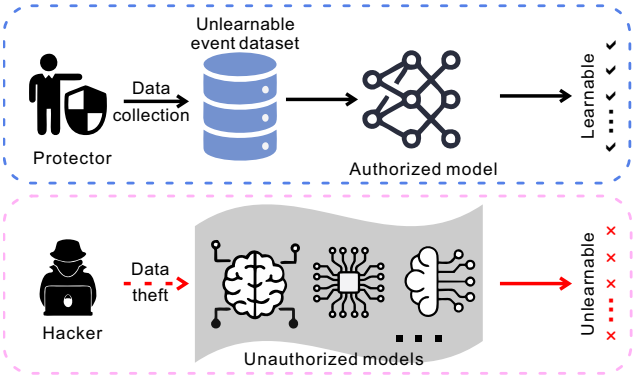

Figure 5. Our scenario of the unlearnable event streams (UEvs). For authorized training, the UEvs can be effectively used to train downstream models and achieve correct predictions. However, if hackers train their authorized networks without our authority that they cannot achieve the reliable performance.

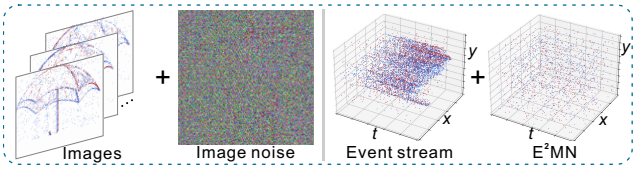

Figure 6. Comparison between the image noise and our E^{2}MN (sample-wise noise).

Although UEvs and UEs adopt the same core idea: error minimizing loss function, to generate unlearnable noise, using UEs to protect event data is impossible, as shown in Fig. 6. The noise is only injected into event representations while the original event data is not protected. Our UEvs perturb the original event data that shows better practicality.

8. UEvs algorithm

The algorithm of our UEvs is shown in Algorithm 1. Here, more detailed explanations about our algorithm are provided. To generate unlearnable event streams, the protector needs to provide the surrogate model f', target dataset $\mathcal{D}_c$, training step M, and classification accuracy γ. f' is employed to calculate the event error-minimizing noise on the dataset $\mathcal{D}_c$. M denotes the training iteration of f' in every epoch of noise generation, which is limited since more efforts should be attached on the δ optimization in $Eq.$ (2). Specifically, the training process of f' is listed in Lines#3-6. We randomly sample M batches of event streams from $\mathcal{D}_c$ and incorporate with the noise (sample-wise noise or class-wise noise) to train the surrogate model. $\mathcal{L}^*$ is our loss function (Eq. (5)), consisting of the cosine similarity loss function and cross-entropy loss function. The final loss is calculated as:

$$\begin{aligned} loss = \lambda_1([1 + \frac{f'(\mathcal{R}(\mathcal{E}))_{conv} \cdot f'(\mathcal{R}(\mathcal{E}) + \delta)_{conv}}{||f'(\mathcal{R}(\mathcal{E}))_{conv}|| \times ||f'(\mathcal{R}(\mathcal{E}) + \delta)_{conv}||}])/2) + \\ \lambda_2(-[l_i log f'(\mathcal{R}(\mathcal{E}) + \delta) + (1 - l_i) log(1 - (f'(\mathcal{R}(\mathcal{E}) + \delta))]), \end{aligned} \tag{6}$$

where $(\cdot)_{conv}$ denotes the convolution features extracted by the last convolution layer of surrogate model f', B indicates the batch size, $\mathcal{R}$ means event representation, converting an event stream into the event stack. Eq. (6) illustrates the training pipeline of f' on a batch of event streams. It is not required to calculate the cost of $f'(\mathcal{R}(\mathcal{E}))$ because this would cause the surrogate model to focus on learning the real semantic features, thereby preventing the optimization of our unlearnable noise.

After training, we generate the event error-minimizing noise for entire $\mathcal{D}_c$ according to Eq. (3), as shown in Lines#7-10. $Clip(\cdot)$ denotes clipping those noise that exceed $-\epsilon$ or $+\epsilon$ back to this region. The noise generation and surrogate model training would be terminated once the classification accuracy tested under δ is higher than γ. This termination demonstrates that the generated noise can effectively guide the model to conduct predictions without relying on image semantics. Therefore, the noise δ is able to prevent the unauthorized model from learning informative knowledge from our data.

Due to the special characteristic of event data, our noise δ is generated based on the event stack. It's necessary to conduct event reconstruction to generate the unlearnable event stream from its corresponding unlearnable event stack (Lines#13-17). To ensure the δ can be compatible with event data, we propose a projection strategy $\mathbf{P}(\cdot)$ to sparsify the noise into $\{-0.5, 0, +0.5\}$[2]. Then, we integrate the projected noise with an event stack and clip it into $[0, 1]$ to generate the unlearnable event stack. Detailed noise embedding process is shown in Table 4. Finally, a retrieval strategy $\mathcal{R}'$ is proposed to search the compressed time stamps from the original event streams to achieve the reconstruction. Based on this algorithm, our valuable event data can be protected well that prevents unauthorized data exploitation, as shown in Fig. 5.

[2]In image area, the generated noise is directly added on the images, rendering the unlearnable examples via modifying the pixel values. However, for event data, which consists solely of binary events, we can only transform the event stream into an unlearnable one on the event level via deleting events or inserting new events. Therefore, we define the projected values as $\{-0.5, 0, +0.5\}$ to be compatible with event stacks ($\{0, 0.5, 1.0\}$).

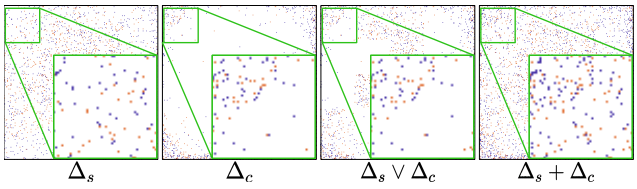

Figure 7. Illustrative examples of our E^{2}MN.

9. Class-wise and sample-wise noise

Our E^{2}MN consists of two kinds of noise: class-wise noise and sample-wise noise, which are all generated based on Eq. (2). The sample-wise noise is generated case by case, which leads to every generated noise being only workable for the single event stream. This limits the practicality of sample-wise noise, especially in the dataset scale grows or new event streams are captured. Hence, an alternative way is class-wise noise, which is generated class by class. It means that a kind of noise can be injected into different event streams sampled from the sample class. Another advantage is that class-wise noise consumes less memory than sample-wise noise in noise optimization. Although two kinds of noises achieve similar performance in Table 2, class-wise noise achieves better stealthiness than sample-wise noise as shown in Fig. 3 and Table 1.

Considering their respective advantages, we combine the two noises to explore whether it can bring more benefits. We have proposed union and addition operations in Sec. 4.3 to evaluate. For $\Delta_s \vee \Delta_c$, we randomly choose Δ_s or Δ_c to protect the event. The form of this kind of noise resembles both of them because only a simple random sampling operation is employed. For $\Delta_s + \Delta_c$, we fuse two kinds of noise by element-wise addition to perturb the event stream. As shown in Fig. 7, this configuration increases the number of perturbations introduced into the event data, resulting in higher effectiveness (see E4 of Table 3).

10. Projection discussion

Our projection strategy is designed to sparsify the noise δ to ensure compatibility with event stacks. We illustrate a detailed confusion matrix of the noise embedding in Ta-

Table 4. Confusion matrix of embedding the noise (E^{2}MN) into an event stack (E. stack). The final unlearnable event stack would be clipped into $[0, 1]$.

		δ		
		-0.5	0	$+0.5$
E. stack	0	original event	original event	event deletion
	0.5	event generation	no event	event generation
	1.0	event deletion	original event	original event

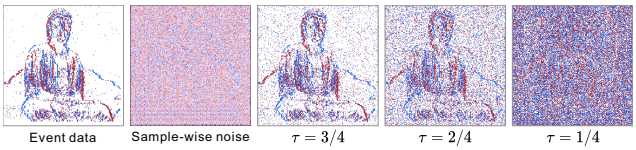

Figure 8. Visualization of event data, sample-wise noise, and the corresponding unlearnable event streams with different projection parameters τ.

ble 4. If $\delta = -0.5$ is added to a negative event (0), the event keeps its original value. If added to a positive event (1), the event is deleted. A new event with negative polarity is created when adding -0.5 to the pixel value 0.5 sampled from the event stack. In this section, we showcase the importance of the parameter τ in Eq. (4). As discussed in [22], the added noise should be imperceptible to human eyes and does not affect the normal data utility. Hence, we introduce the parameter τ to balance the imperceptibility and unlearnability of our E^{2}MN. As shown in Fig. 8, the larger τ can lead to better imperceptibility but the less unlearnable noise has been introduced, which harms the unlearnability. We have tested our sample-wise noise with $\tau = 7/8$ on the N-Caltech101 dataset and obtained the accuracy of ResNet18 by 4.88, which is higher than the accuracy (0.52) tested by $\tau = 3/4$. This demonstrates the great challenges in balancing the imperceptibility and unlearnability of our E^{2}MN.

11. Baseline setting

To further evaluate the effectiveness of our UEVs, we introduce the straightforward event distortions as our baselines, which are inspired from event data augmentation [17] and backdoor attacks [55]. Data augmentation is proposed to enrich the training data for improving the model's performance, which usually employs data distortion operations to augment the sample. Generally, the quality of these augmented samples is lower than the original ones. Therefore, we propose simple coordinate shifting (CS), timestamp shifting (CS), polarity inversion (PI), and area shuffling (AS) based on [17] to corrupt the event streams for preventing unauthorized usage. Additionally, we also propose the manual pattern (MP) based on backdoor attacks [55] to perturb our event streams. We inject a pre-defined pattern for those event streams sampled from the same class to prevent unauthorized data usage, which can be viewed as a class-wise noise. According to Table 2, we can find that compromising the quality of our event datasets can degrade the performance of downstream models. However, the unlearnability is rather limited and does not prevent the downstream models from learning informative knowledge.

12. Additional experiments

Various event representations. In our main experiments, we adopt the voxel-grid event stack as our event representation to evaluate the effectiveness. To test the generalizability of our UEVs among different representations, the event frame (**EF**) [43] and Time surface (**TS**) [50] are adopted. As shown in the **E1** of Table 5, our UEVs can still prevent unauthorized event data usage under **EF** and **TS** representations, showing high robustness and generalizability.

Generalizability. According to [9], we set the time bin to 16 to represent the event stream. To evaluate the generalizability of our UEVs on different time intervals, we change the size of Δ_t to $\mathbf{0.5}\times$ and $\mathbf{2}\times$ to conduct ablation studies. as shown in **E2** of Table 5, our UEVs still shows high protection ability to prevent the unauthorized event data usage.

Diverse Adversarial attacks. Apart from the PGD [38] and FGSM [16] attacks, we add new adversarial attack methods: C&W [4] and MIFGSM (MIF.) [6]. CW attack is an optimization-based method that crafts minimal perturbations to mislead neural networks while remaining imperceptible. MIFGSM is an iterative adversarial attack that enhances the basic FGSM by incorporating the momentum. Compared with MIFGSM, CW attack achieves better unlearnable functionality. As shown in **E3** of Table 5, our method can still achieve reliable unlearnable performance while adopting different adversarial attacks.

Table 5. Quantitative results tested by Res50 on N-Caltech101.

	E1		E2		E3		E4
	EF	TS	$0.5\Delta_t$	$2\Delta_t$	C&W	MIF.	UEs
Δ_c	5.17	10.09	5.46	4.94	8.73	15.43	5.51
Δ_s	8.85	16.48	5.17	5.05	12.15	14.47	18.38

Naive image baselines. Image-based methods are unable to directly secure event data due to data differences, which can only safeguard the corresponding event representation. In **E4** of Table 5, although the image approach, UEs [22], performs well in event representations, it cannot prevent malicious users from misusing the **original event**.

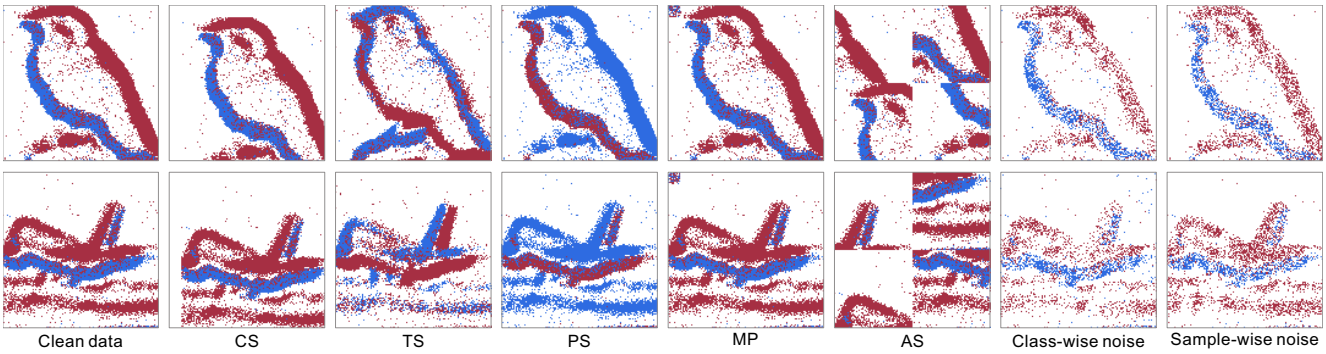

Figure 9. Visualization results of various noise forms on the CIFAR10-DVS dataset [31]. Blue/Red points denote the events with $p = +1/-1$. Our noise E^{2}MN does not introduce much noise in the background region, which maintains good imperceptibility.

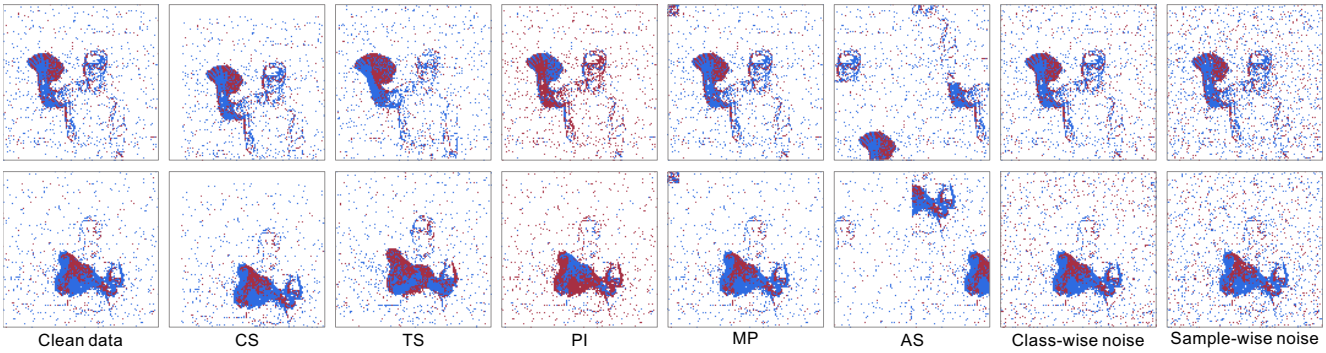

Figure 10. Visualization results of various noise forms on the DVS128 Gesture dataset [1]. Blue/Red points denote the events with $p = +1/-1$. Our noise E^{2}MN does not corrupt the target objects, and the noise distributed in the background region appears quite realistic.

Event to image reconstruction. We employ an event-to-image method (E2VID [44]) to evaluate the generazibility. As shown in Fig. 11, our method prevents E2VID [44] from reconstructing details from the protected event data, thereby providing solutions for privacy preserving.

Figure 11. Visualization frames reconstructed by E2VID [44].

Unlearnable cluster. We extend our class-wise noise into a cluster-wise one. We ❶ employ K-Means+ResNet50 to cluster the N-Caltech101 into **10** classes; ❷ train a surrogate model on 10 classes to generate **cluster-wise noise**; ❸ train ResNet18 on the whole classes (**101**) with cluster-wise noise. The cluster version of our method can reduce the classification accuracy from 0.787 to 0.189.

13. Dataset details

To evaluate the effectiveness of our method, we employ four popular event-based datasets in our experiments, including N-Caltech101 [42], CIFAR10-DVS [31], DVS128 Gesture [1], and N-ImageNet [27]. N-Caltech101 is the neuromorphic version of the image dataset, Caltech101 [11], which has 101 classes and 4356, 2612, and 1741 samples for training, validation, and testing, respectively. CIFAR10-DVS is generated based on image datasets CIFAR-10 [29], where the training set, validation set and testing set contain 7000, 1000, and 2000 samples, respectively. DVS128 Gesture contains 11 classes from 29 subjects under 3 illumination conditions, which has 1176 training samples and 288 testing samples. The N-ImageNet (mini) dataset is derived from the ImageNet dataset. It utilizes an event camera to capture RGB images shown on a monitor. This dataset includes 100 object classes, with each class having 1,300 streams for training and 50 streams for validation. Details of each dataset are shown in Table 6.

14. Visualization comparison

To evaluate the imperceptibility of our E^{2}MN, we show more visualization results in Fig. 9 and Fig. 10. In Fig. 9, we sample event streams from the CIFAR10-DVS dataset [31] to generate the unlearnable ones via five straightforward distortions and our two kinds of noise. It's clear that there

Table 6. Details of four used datasets in our experiments.

	N-Caltech101	CIFAR10-DVS	DVS128 Gesture	N-ImageNet
Type	Simulated	Simulated	Real	Simulated
Calsses	101	10	11	1000
Resolution	180×240	128×128	128×128	480×640
Event Camera	ATIS camera	DVS128	DVS128	Samsung DVS Gen3
Train	4536	7000	1176	130000
Val	2612	1000	288	50000
Test	1741	2000	288	50000

is an x-y offset in the unlearnable event streams generated by CS compared to the clean data. TS alters time stamps to pollute the input event streams, resulting in a noticeable difference between the distorted samples and the clean data. PI reverses the polarity of the event stream to degrade the quality, thereby reducing the quality of the training data. AS rearranges the event data at the block level to produce unlearnable data, which exhibits low imperceptibility. Our class-wise noise and sample-wise noise perturb the event stream with imperceptible noise that shows better invisibility than other comparison methods. The visualization results sampled from DVS128 Gesture dataset [1] are shown in Fig. 10. It's clear that our E^{2}MN achieves the best unlearnability while maintaining good imperceptibility.

To visualize the influence caused by our E^{2}MN, we employ GradCAM to highlight several unlearnable event streams in Fig. 12. Figures (A), (B), (C), (D), and (E) are rendered by A-shuffle, M-pattern, P-inverse, Class-wise noise, and sample-wise noise, respectively. Our method builds a shortcut between the input samples and labels that suppresses the model learning semantic features, resulting in a lower response on the foreground regions.

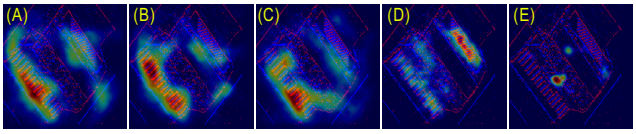

Figure 12. GradCAM of event distortions and our methods.

15. Social impact

The social impact of UEVs is multifaceted, addressing key issues around event data security, privacy, and ethics. By making event datasets unlearnable, our method helps protect individuals' data from being used without authorization. This is particularly important in an era where event data privacy concerns are paramount, and unauthorized data usage can lead to significant privacy breaches. The method provides a robust protection way for event data owners, ensuring that their event streams cannot be exploited by unauthorized entities. This fosters greater trust in event data sharing. With the application of UEVs, there is a push towards more ethical event data practices. Entities will need to obtain proper authorization and consent before using event data, promoting a culture of respect for data ownership and user rights.

Overall, our UEVs contributes significantly to the advancement of secure and trustworthy data sharing, promoting a safer and more ethical event data ecosystem.

16. Future work

UEVs is the first method designed for generating unlearnable event streams, which provides a possible solution to prevent the unauthorized usage of our valuable event data. We mainly focus on studying the unlearnability of event streams in the main paper, while causing some limitations in terms of transferability evaluation, generation efficiency, and defense mechanism. To address these issues beyond our research topic in this work, we will explore the following directions in the future:

- Transferability evaluation: We plan to extend the evaluation of our method from classification task to other event datasets and event vision tasks, enhancing the transferability. This comprehensive testing is helpful to demonstrate the effectiveness of UEVs that promote the trustworthy event data sharing. However, the effectiveness of UEVs may be decreased across different vision tasks. A possible solution is that adopt the foundation model as our surrogate model to calculate the event error-minimizing noise, which is trained with a large amount of data that has strong generalization capabilities. It can be applied to a variety of tasks without training independent models for each specific task.
- Generation efficiency: According to our experiments, we find that the efficiency of the sample-wise noise generation depends on the scale of the event dataset. The larger the scale of the event dataset, the lower the efficiency of the sample-wise noise generation. We propose addressing this issue via a noise generator. We train this generator with the surrogate model jointly, which aims to enable the generated noise to minimize the cost of the surrogate model. This training pipeline avoids storing the generated medium noise, which saves efficiency significantly. Once the optimization has been finished, we can employ this generator to generate sample-wise noise for each input sample with high efficiency.
- Defense mechanism: It's crucial to investigate potential defense mechanisms against the generation of malicious unlearnable event streams. If we release our unlearnable dataset online, hackers could manipulate these samples to force their models to learn the information. By understanding possible defense mechanisms, we can develop more reliable methods for creating unlearnable event datasets, thereby preventing unauthorized usage. Furthermore, elucidating these defense mechanisms can help users improve the dataset quality, ultimately saving training time and computing resources.