# Exploring Vulnerabilities in Spiking Neural Networks: Direct Adversarial Attacks on Raw Event Data

Yanmeng Yao[1*], Xiaohan Zhao[1*], and Bin Gu[23✉]

[1] Nanjing University of Information Science and Technology, Nanjing, China
[2] School of Artificial Intelligence, Jilin University, China
[3] Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates
{yanmengyao1213, ritianzhaoxx, jsgubin}@gmail.com

**Abstract.** In the field of computer vision, event-based Dynamic Vision Sensors (DVSs) have emerged as a significant complement to traditional pixel-based imaging due to their low power consumption and high temporal resolution. These sensors, particularly when combined with Spiking Neural Networks (SNNs), offer a promising direction for energy-efficient and fast-reacting vision systems. Typically, DVS data are converted into grid-based formats for processing with SNNs, with this transformation process often being an opaque step in the pipeline. As a result, the grid representation becomes an intermediate yet inaccessible stage during the implementation of attacks, highlighting the importance of attacking raw event data. Existing attack methodologies predominantly target grid-based representations, hindered by the complexity of three-valued optimization and the broad optimization space associated with raw event data. Our study addresses this gap by introducing a novel adversarial attack approach that directly targets raw event data. We tackle the inherent challenges of three-valued optimization and the need to preserve data sparsity through a strategic amalgamation of methods: 1) **Treating Discrete Event Values as Probabilistic Samples**: This allows for continuous optimization by considering discrete event values as probabilistic space samples. 2) **Focusing on Specific Event Positions**: We prioritize specific event positions that merge original data with additional target label data, enhancing attack precision. 3) **Employing a Sparsity Norm**: To retain the original data's sparsity, a sparsity norm is utilized, ensuring the adversarial data's comparability. Our empirical findings demonstrate the effectiveness of our combined approach, achieving noteworthy success in targeted attacks and highlighting vulnerabilities in models based on raw event data.

**Keywords:** Adversarial attacks · Spiking neural networks · Raw events

## 1 Introduction

Dynamic Vision Sensors (DVSs) [28], as a leading example of event cameras, signify a major advancement in bio-inspired vision technology. These sensors detect brightness changes and emit asynchronous spikes when these changes surpass a specific threshold.

* Equal contributions

Their output, stored in sparse coordinate (COO) format [3, 40, 41][4], offers notable advantages over traditional frame-based cameras, including high temporal resolution, low power consumption, and minimal data redundancy [14].

The unique characteristics of this output are well-aligned with the functionalities of Spiking Neural Networks (SNNs), which emulate the animal brain's information processing using binary spikes across layers, incorporating temporal dynamics. This alignment is further enhanced by the synergy with neuromorphic devices, which enable efficient asynchronous processing of event data [19,45]. However, the high temporal resolution and inherent sparsity of event data present challenges such as increased memory overhead and decreased training efficiency. To address these issues, raw event data is often transformed from sparse COO format to denser grid-based formats, such as event-count frames, through event aggregation in many vision pipelines.

Security concerns in SNNs processing event datasets are paramount, particularly in the context of adversarial attacks on the raw event data, an area that remains underexplored in SNN adversarial attack research. Attacking raw event data presents a unique set of challenges. First, it requires navigating an extensive optimization space, often exceeding 10,000 timestamps per sample at the microsecond ($\mu s$) level, making it impractical to exhaustively search for adversarial samples. Second, it involves addressing the high temporal resolution, leading to significant memory challenges and computational costs when using gradient methods for optimization. Third, maintaining the inherent sparsity of the original event data in adversarial samples is crucial, as samples with too many or too few events are easily detectable. Additionally, the optimization problem in SNN adversarial attacks involves three-valued optimization, distinct from the pixel-based optimization in Artificial Neural Networks (ANNs), requiring more targeted analysis and solutions. These factors collectively introduce a layer of complexity that necessitates advanced techniques to craft effective and subtle adversarial samples.

Thus, due to the above challenges, prevailing studies instead focus on attacking grid representations, which is an intermediate representation of event data during the optimization process. Studies in this domain are broadly categorized into two approaches: attacks on continuous grid representations [5, 22, 29, 30] and attacks on binary grid representations [6,26,33]. Attacks on continuous grid representations align with traditional attack methods [5], such as Projected Gradient Descent (PGD) [17] and Fast Gradient Sign Method (FGSM) [23]. Conversely, binary grid representation attacks address additional challenges posed by discrete values, for instance, SpikeFool [6] rounds smooth values to integers iteratively, updating these values using gradients from their integer counterparts; DVS-attacks [33] employs heuristic methods that bypass optimization; Spike-Compatible Gradient [26] directly optimizes in the discrete space using spike-compatible gradients. However, the process of grid representation may either be nonexistent [18] or seamlessly integrated and concealed within the vision pipeline (as illustrated in Fig. 1). This implies that even with complete knowledge of an online model, an attacker is likely unable to launch an attack in the form of grid-based representation, as the model does not provide an interface for grid-based representations.

In response to the challenges of three-valued optimization, expansive optimization spaces, discrete values, and the requirement for high sparsity in event data, we pro-

[4] 'Raw event data' refers to this COO format data collected by event cameras in this paper.

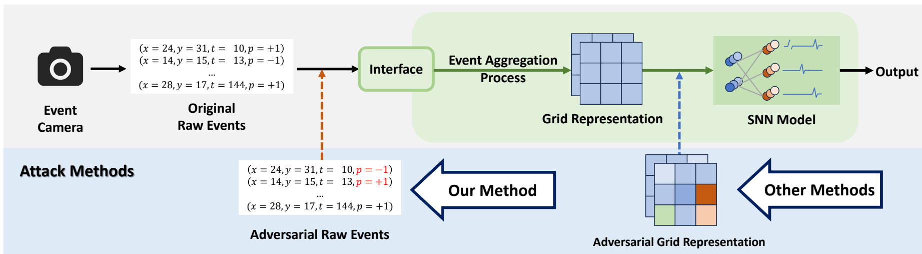



**Fig. 1:** Comparison of our method with others. Green arrows indicate operations not accessible in the final attack implementation. The grid representation process is integrated into the vision pipeline, exposing only an interface for raw event data processing. Our method produces adversarial samples directly compatible with this interface, whereas other methods generate samples as intermediate results within the vision pipeline.

pose a systematic solution. By viewing discrete values as samples from a continuous probability space, we employ the Gumbel-Softmax sampling technique and a straight-through estimator to convert the discrete optimization problem into a continuous one. We also focus on specific locations that combine the position of original events and additional positions of events with the target label to optimize, significantly narrowing the optimization space while preserving data sparsity and improving attack performance. Additionally, we introduce an extra penalty term in the loss function to further enhance performance. Our empirical results demonstrate that this strategy achieves superior performance in maintaining greater sparsity than other methods in targeted attacks on SNN models. Further ablation experiments demonstrate that the organic combination of these techniques plays an important role in achieving efficient adversarial attacks. To the best of our knowledge, our work represents the first successful implementation of an attack on raw event data targeting SNNs. Our contributions are summarized below:

1. **Gradient Optimization in Discrete Space:** To overcome the problem of gradient optimization being inapplicable in discrete spaces, we employ the Gumbel-Softmax sampling strategy and a straight-through estimator to transform the discrete optimization problem into a continuous one.
2. **Efficient Position Select:** To address the challenge of optimizing within vast spaces, we focus on specific locations, which combine the position of original events and additional positions of events with the target label to optimize.
3. **Effective Combination:** We incorporate an extra penalty term within the objective problem, and extensive empirical evaluations and detailed ablation studies illustrate that the strategic integration of the core technologies above yielded significant results on the raw event data adversarial attacks.

## 2 Preliminaries

### 2.1 Event Data

**Raw Event Data.** Event cameras operate on an innovative principle where individual pixels trigger events independently in response to logarithmic brightness changes, denoted as $L(x, y, t)$. These events occur when:

$$|L(x, y, t) - L(x, y, t - \varDelta t)| \geq C, \tag{1}$$

where $C$ represents the contrast threshold, and $\varDelta t$ is the elapsed time since the last event. Typically, $p$ is utilized to signify the polarity of the brightness change:

$$p = \text{sign}(L(x, y, t) - L(x, y, t - \varDelta t)). \tag{2}$$

Over a specified time interval $\varDelta\xi$, the camera generates a sequence of events $\mathcal{E}$:

$$\mathcal{E} = \{e_k\}_{k=1}^{N} \stackrel{\text{def}}{=} \{(x_k, y_k, t_k, p_k)\}_{k=1}^{N}. \tag{3}$$

Given the camera's asynchronous nature, events within the time window $\varDelta\xi$ are encapsulated as a sparse COO tensor. The sparse COO tensor comprises indices $I = \{(x_k, y_k, t_k)\}_{k=1}^{N}$ and values $V = \{p_k\}_{k=1}^{N}$, representing the locations and polarities of events, respectively.

**Grid Representation.** Due to the limitations of most networks in handling sparse COO format tensors, a mapping $\mathcal{M} : \mathcal{E} \rightarrow \mathcal{T}$ is required to translate the event set $\mathcal{E}$ into a dense tensor $\mathcal{T}$. Conceptually, events can be visualized as points within a four-dimensional manifold defined by spatial coordinates ($x$, $y$), time, and polarity. These points are aggregated through *kernel convolutions* as detailed in [15]:

$$S_{\pm}(x, y, t) = \sum_{e_k \in \mathcal{E}_{\pm}} f_{\pm}(x_k, y_k, t_k) k(x_l - x_k, y_m - y_k, t_n - t_k), \tag{4}$$

where this aggregation occurs in continuous space and time for both positive ($\mathcal{E}+$) and negative ($\mathcal{E}-$) event polarities. The function $k(x, y, t)$ denotes the aggregation kernel, which represents the presence of spikes. Commonly used kernels include the Dirac pulse ($\delta$), $k(x, y, t) = \delta(x, y)\delta(t)$, *alpha*-kernel, $k(x, y, t) = \delta(x, y)\frac{et}{\tau}exp(-t/\tau)$ [24, 34], and exponential kernel $k(x, y, t) = \delta(x, y)\frac{1}{\tau}\exp(\frac{-t}{\tau})$ [39]. The function $f_{\pm}(x, y, t)$ represents the measurement of each position. Examples of such functions are event polarity $f_{\pm}(x, y, t) = \pm 1$, event count $f_{\pm}(x, y, t) = 1$, and normalized time stamp $f_{\pm}(x, y, t) = \frac{t - t_0}{\varDelta t}$. The grid representation of these events is typically achieved by sampling the convolution signals at regular intervals:

$$S_{\pm}[x_l, y_m, t_n] = \sum_{e_k \in \mathcal{E}_{\pm}} f_{\pm}(x_k, y_k, t_k) k(x_l - x_k, y_m - y_k, t_n - t_k). \tag{5}$$

In practical applications, the spatiotemporal coordinates of event data denoted as $x_l$, $y_m$, $t_n$, are systematically organized into a voxel grid. This arrangement is defined such that $x_l \in \{0, 1, ..., W-1\}$, $y_m \in \{0, 1, ..., H-1\}$, and $t_n \in \{t_0, t_0 + \varDelta t, \ldots, t_0 + B\varDelta t\}$, effectively compressing the high temporal resolution of the original event data into $B$ discrete temporal bins, each with a size of $\varDelta t$. Noticeably, when $\varDelta t = 1$, $k(x, y, t) = \delta(x, y)\delta(t)$, and $f_{\pm}(x, y, t) = 1$, raw event representation simplifies to its dense form $\mathcal{E}^*$, filling indices with values and setting others to 0.

Due to the close relationship between the grid representation process and the model, the grid representation is often integrated with the model itself. Therefore, in the final implementation of an attack, even in a white-box setting, intermediate results like grid representation may not be externally exposed. This is because modularization eliminates the need for such an interface (please refer to the green part of Fig. 1 and Fig. 3).

**Table 1:** Comparison of SNN attack methods on event data.

| Methods | Gradient-Based | Data Format | |
|---|---|---|---|
| | | Binary Frame | Raw Event |
| SpikeFool [6] | ✓ | ✓ | |
| Spike-Compatible [26] | ✓ | ✓ | |
| DVS-attacks [33] | | ✓ | |
| **Ours** | ✓ | ✓ | ✓ |

### 2.2 Spiking Neural Network

Spiking Neural Networks, drawing inspiration from biological neural circuits, are engineered to replicate these circuits' information transmission through spike-based communication among layers. Central to the function of SNNs are spiking neurons, governed by discrete-time equations as described in [12]:

$$H[t] = f(V[t-1], X[t]), \tag{6}$$
$$S[t] = \Theta(H[t] - V_{th}), \tag{7}$$
$$V[t] = H[t] + V_{\text{reset}} S[t]. \tag{8}$$

Here, $X[t]$ denotes the input current at time $t$. $H[t]$ and $V[t]$ represent the membrane potential after neuronal dynamics and following a spike's trigger at time $t$, respectively. $V_{th}$ stands for the firing threshold, with the Heaviside step function $\Theta(x)$ defined as $\Theta(x) = 1$ for $x \geq 0$ and $\Theta(x) = 0$ for $x < 0$. The output spike at time $t$, $S[t]$, is 1 if a spike occurs and 0 otherwise. $V_{\text{reset}}$ indicates the reset potential after a spike.

The neuronal dynamics function $f(\cdot)$ in Eq. (6) varies across different spiking neuron models. The most prevalent models are Integrate-and-Fire (IF) [16] and Leaky Integrate-and-Fire (LIF) [20], formulated in Eq. (9) and Eq. (10), respectively:

$$H[t] = V[t-1] + X[t], \tag{9}$$
$$H[t] = V[t-1] + \frac{1}{\tau}(X[t] - (V[t-1] - V_{\text{reset}})). \tag{10}$$

In these equations, $\tau$ represents the membrane time constant. For effective backpropagation optimization in SNNs, the surrogate gradient method is utilized. This approach redefines $\Theta'(x)$ as $\sigma'(x)$ during error back-propagation, where $\sigma(x)$ is the surrogating function.

In SNNs, neuron communication is mediated by spikes, a process that risks information loss with increasing network depth. As depicted in the left of Fig. 2, the forward calculation in SNNs resembles the behavior of a Recurrent Neural Network (RNN), with the dynamic nature of information processing. In SNNs, IF and LIF neurons commonly substitute for the activation layer found in traditional neural network architectures. However, this substitution presents a critical hurdle: the non-differentiability of the spiking mechanism. To address this issue during the backpropagation process, surrogate gradient functions are commonly utilized to approximate the derivative of the

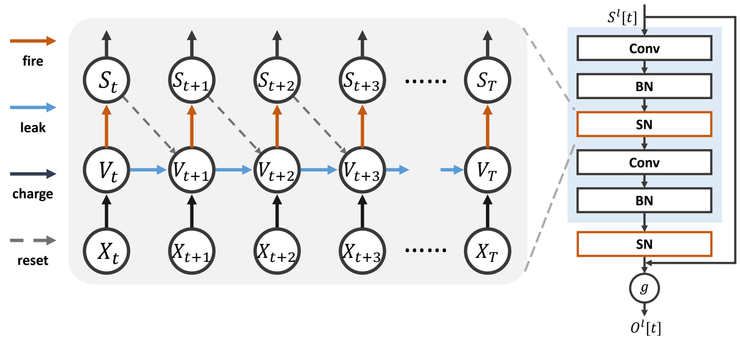



**Fig. 2:** Illustration of the SNN model. The left side details the forward pass within a layer where $T$ is the frame number, while the right side showcases a residual block of the SEW ResNet.

spike function. As shown in the right of Fig. 2, the depicted residual block in SEW ResNet [13], which is a sample of substituting traditional activation functions with spiking neurons.

# 3 Methods

## 3.1 Problem Formulation

**ANN Adversarial Attack.** Adversarial attacks aim to subtly manipulate input data, causing a model to output incorrect decisions while these changes remain undetectable to human observers. Significant progress has been made in developing adversarial techniques for ANNs [2, 10].

Let $\mathcal{F}_{\mathcal{A}}(x) : \mathbb{R}^d \to \mathbb{R}^K$ be a well-trained ANN classifier, where $x$ denotes the input image, and the output is a class label from $K$ possible classes. We have $d = w \times h \times c$, where $w, h, c$ denotes the width, height, and channel of the image. The adversarial attack seeks to craft a perturbation $\Delta x$ with minimal norm ($\|\Delta x\|_{\hat{p}} \leq \epsilon$) that misleads $\mathcal{F}_{\mathcal{A}}$, causing it to classify an image $x_0$ from class $y_0$ as a different class $y$. The formal optimization goal is:

$$\underset{\Delta x}{\operatorname{argmin}} \ \|\Delta x\|_{\hat{p}}, \quad \text{s.t. } \mathcal{F}_{\mathcal{A}}(x_{\text{adv}}) = y, \ \ 0 \leq x_0 + \Delta x \leq 1. \tag{11}$$

Considering the complexity of ensuring $\mathcal{F}_{\mathcal{A}}(x_0 + \Delta x) = y$, an alternative proposed by [8] is to minimize:

$$\mathcal{L} := l(\mathcal{F}_{\mathcal{A}}(x_{\text{adv}}), y), \quad \text{s.t. } \|\Delta x\|_{\hat{p}} \leq \epsilon, \ \ 0 \leq x_0 + \Delta x \leq 1, \tag{12}$$

where $l(\cdot)$ is the loss function, $\epsilon$ denotes the maximal allowable perturbation under the $\ell_{\hat{p}}$ norm. The constraints ensure that the perturbation is within the allowable limit and that the adversarial image remains valid.

**Transformation to Grid-based Representation.** The exploration of adversarial scenarios in SNNs often focuses on grid-based representations, which can be either continuous or binary. The optimization challenges of continuous grid-based representations align with those encountered in ANNs, so it is not discussed here.

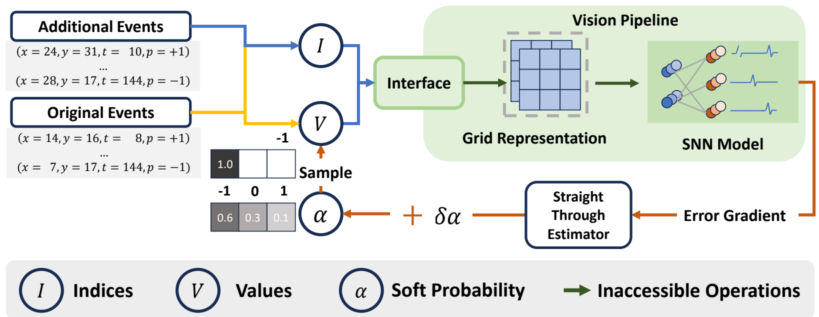



**Fig. 3:** Schematic of the adversarial attack methodology. The gray dashed box indicates that in some studies, grid representation may not exist. Original and target-labeled event indices form the position of the attack. During each iteration, discrete values $V$ are sampled from the categorical distribution with soft probability $\alpha$.

In binary grid-based representations, a location within a time window is assigned a value of 1 if at least one spike occurs, and 0 otherwise, which simplifies the representation of spikes over time. $\tilde{x} \in \{0,1\}^m$ denotes a binary grid-based representation sample, where $m = T \times W \times H \times 2$ encapsulates the dimensions of the temporal ($T$), width ($W$), and height ($H$) aspects of each frame, along with the channels. Let $\mathcal{F}_{\mathcal{S}}(\tilde{x}) : \{0,1\}^m \to \mathbb{R}^K$ be a well-trained SNN classifier. The constraint $0 \leq x_{\text{adv}} \leq 1$ was transformed to $\tilde{x}_{\text{adv}} \in \{0,1\}^m$. Thus, we can obtain the objective problem

$$\mathcal{L} = l(\mathcal{F}_{\mathcal{S}}(\tilde{x}_{\text{adv}}), y), \quad \text{s.t. } \|\Delta\tilde{x}\|_{\hat{p}} \leq \epsilon, \;\; \tilde{x}_{\text{adv}} \in \{0,1\}^m. \tag{13}$$

**Transformation to Raw Event Data.** Attacking on the raw event $\mathcal{E} \in \{-1,1\}^n$, including indices $I$ and values $V$, is much more complex than the above data formats, where $n \gg N \times W \times H \times 2$ due to high temporal resolution. $N$ is the number of events. Let $\mathcal{F}_{\mathcal{S}}(\mathcal{E}) : \{-1,1\}^n \to \mathbb{R}^K$ denote a well-trained SNN classifier exclusively accepting raw event data. The constraint needs to be further transformed.

According to the nature of raw event data, we set the range of event values to $\{-1,1\}$, with $-1$ indicating reduced light and $1$ indicating increased light. Therefore, the constraint $0 \leq x_{\text{adv}} \leq 1$ becomes

$$V \in \{-1,1\}^n.$$

Given that some samples encompass hundreds of thousands of events, event-wise optimization is much more difficult than those of pixel-wise images and grid-based representation samples.

To align with the standard constraint $\|\Delta\tilde{x}\|_{\hat{p}} \leq \epsilon$ in vanilla adversarial attacks, we introduce a novel constraint of the distance between adversarial and original events in dense representations, defined as

$$\|\Delta\mathcal{E}^*\|_{\hat{p}} \leq \epsilon := \|\mathcal{E}^*_{\text{adv}} - \mathcal{E}^*\|_{\hat{p}} \leq \epsilon,$$

where $\cdot^*$ represent dense representations.

To preserve the imperceptibility of perturbations within the sparse event data, raw events can not be added or removed too much. Thus, we need to introduce a novel constraint to maintain the event count within a reasonable range, which satisfy

$$|\mathcal{E}_{\text{adv}}| \in [\theta_l, \theta_h],$$

where $\theta_l$ and $\theta_h$ represent lower and upper bounds. Similar to the $\ell_0$ constraint within ANNs, the optimization problem with this constraint is non-trivial.

Finally, we transform the distance constraint into the penalty term. The objective problem can be reformulated as follows:

$$\begin{aligned} &\mathcal{L} = l(\mathcal{F}_{\mathcal{S}}(\mathcal{E}_0 + \Delta\mathcal{E}), y) + \lambda \|\Delta\mathcal{E}^*\|_{\hat{p}}, \\ &\text{s.t. } \ V \in \{-1, 1\}^N, \\ &\qquad |\mathcal{E}_{\text{adv}}| \in [\theta_l, \theta_h], \end{aligned} \tag{14}$$

where $y_0 = \mathcal{F}_{\mathcal{S}}(\mathcal{E}_0) \neq y$, $\lambda$ serves as the weight parameter for the sparsity penalty and $\hat{p} = 1$ in this article. Consequently, gradient-based adversarial attacks on raw event data exhibit greater complexity than attacks on grid representations.

### 3.2 Reparameterization

**Gumbel-Softmax Technique.** Handling raw event data, irrespective of the grid representation format (event count frame, Event Spike Tensor [15], voxel-grid [44], or two-channels [32, 43]), involves working with indices $I$ and values $V$ containing only integers. The binary nature of elements $p_k$ in $V$, constrained to either $-1$ or $1$, poses a significant challenge for gradient-based optimization. To navigate this, we treat each discrete value as sampled from a categorical distribution $\pi_\alpha$, with $\alpha$ representing log probabilities:

$$p_i = \pi_\alpha = \arg\max_j [\alpha_j^i] - 1, \tag{15}$$

where $\alpha_j^i$ indicates the probability of $p_i$ being $j-1$. $p_i = 0$ means such indices-value pair will be removed when formulating adversarial samples. To enable differentiable optimization in this discrete space, we employ the Gumbel-Softmax technique [21]. This approach allows for the sampling of soft one-hot values $\hat{p}^i$:

$$\hat{p}_j^i = \frac{\exp((\alpha_j^i + g_j)/\kappa)}{\sum_{l=0}^{2} \exp((\alpha_l^i + g_l)/\kappa)} \quad \text{for } i = 1, ..., k, \tag{16}$$

where $\hat{p}_j^i$ represents the probability of $p_i$ being $j-1$. As $\kappa$ approaches zero, the Gumbel-Softmax distribution's samples closely resemble one-hot vectors, aligning with $\pi_\alpha$.

**Straight-Through Estimator.** In the forward pass, Gumbel-Softmax functions as a surrogate for argmax. Here, hard values $p_i = \pi_\alpha$ are used, while the backward pass employs a straight-through estimator for gradient propagation:

$$\nabla_{\hat{p}_j^i} = \begin{cases} \nabla_{p_i}, & \text{if } j = \arg\max_j [\alpha_j^i], \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

**Algorithm 1** Adversarial Sample Generation

**Require:** original indices $I$, values $V$, target event indices $I_{\text{target}}$ and values $V_{\text{target}}$, number of total iterations $T$, learning rate $\gamma$, sample number $M$, target label $y$.
**Ensure:** adversarial sample $\mathcal{E}_{\text{adv}}$.
Initialize: $\alpha = \text{onehot}(\text{concat}(V, 0))$, $I_{\text{adv}} = \text{concat}(I, I_{\text{target}})$
**for** $t = 1$ to $T$ **do**
  **for** $i = 1$ to $M$ **do**
    **forward:**
      Sample $p$ and $\hat{p}$ according to Eq. (15) and Eq. (16)
      Formulate event data $\mathcal{E}^i_{\text{adv}} = (I_{\text{adv}}, \{p_k\}_{k=1}^M)$
      Compute attack loss $\mathcal{L}_i$
    **if** $\mathcal{F}(\mathcal{E}^i_{\text{adv}}) = y$ **then**
      **return** $\mathcal{E}^i_{\text{adv}}$
    **end if**
    **backward:**
      Compute $\nabla_p \mathcal{L}_i$
      Compute $\nabla_{\hat{p}} \mathcal{L}_i$ using straight through estimator from Eq. (17)
  **end for**
  Compute: $\delta\alpha = \frac{1}{M} \sum_{i=1}^M \nabla_{\hat{p}} \mathcal{L}_i \cdot \frac{\partial \hat{p}}{\partial \alpha}$
  Update: $\alpha \leftarrow \alpha - \gamma\delta\alpha$
**end for**

This method ensures effective optimization within a discrete framework, aligning hard values in the forward pass with the model's needs and optimizing the backward pass for efficient gradient propagation.

### 3.3 Efficient Position Selection

The high temporal resolution and inherent sparsity of event data make an attack on the full data space computationally infeasible. To mitigate this, our strategy focuses on a subset of the event data. We commence the attack by integrating additional positions from an image with a designated target label. This is executed by merging indices $I_{\text{original}}$ from $\mathcal{E}_{\text{original}}$ and $I_{\text{target}}$ from $\mathcal{E}_{\text{target}}$, padding additional values with zeros (refer to Fig. 3 for a visual representation of the process). This precision-targeted method significantly curtails computational demands, reducing the original vast optimization space into a relatively small subspace, while ensuring the attack's efficacy. Moreover, We can maintain a controlled sparsity level, ensuring that $|\mathcal{E}_{\text{adv}}|$ does not exceed $|\mathcal{E}_{\text{original}}| + |\mathcal{E}_{\text{target}}|$.

## 4 Experiment

In this section, we explored targeted attacks on datasets captured by DVSs, specifically targeting SNN models. We conducted experiments using both raw event data and grid-based representations. Additionally, we analyzed the effect of Straight-Throught estimators on attack success rate (ASR). Experimental results demonstrated the effectiveness of our method, offering insights into the model's vulnerabilities and potential defense mechanisms. The additional results are detailed in the appendix.

### 4.1 Experimental Setup

**General Setup** Our experiments were conducted using the PyTorch [38] and SpikingJelly framework [11]. The computational setup included systems running Ubuntu 22.04 with 4 NVIDIA GeForce 1080Ti GPUs and Ubuntu 20.04 with 3 NVIDIA GeForce RTX 3090 GPUs. To ensure consistency, the global random seed was set to $0$. In addition, the definitions of notations and parameter settings are detailed in Section A of the appendix. Codes are available from https://github.com/JY-ura/GumbelSoftmaxAttack.

**Datasets and Models** In this study, we utilized three DVS datasets to comprehensively evaluate the performance of all algorithms. We randomly selected 100 correctly classified samples from the following datasets. Details are described below.
1) **CIFAR10-DVS.** The CIFAR10-DVS dataset [25] comprises the conversion of 10,000 frame-based images sourced from the CIFAR-10 dataset into an equivalent number of event streams, including 10 different classes. A resolution of $128\times128$ pixels characterizes each stream.
2) **DVS gesture.** The DVS gesture dataset [1] is a collection of 1077 samples for training and 264 for testing, divided into 11 hand gesture categories. A resolution of $128\times128$ pixels characterizes each stream.
3) **N-MNIST.** The N-MNIST dataset [36] encompasses 60,000 training samples and 10,000 testing samples, mirroring the structure of the original MNIST [9]. Each image is captured at an identical visual scale of $34\times34$ pixels.

Due to page limitations, we focused on training two distinct network models in the main text, the SEW ResNet-34 [13] and the Spiking ResNet-18, using the datasets previously described. We followed the training parameters of [13], ensuring the accuracy of the models. The training parameters and accuracy of the model are shown in Tab. 2. The grid-based representation employed utilizes the Dirac pulse kernel, detailed in Sec. 2.1. The time interval $\Delta t$ is determined based on a fixed frame count of 16. Additionally, our analysis primarily focuses on the way are split by number.

**Baseline** To assess the efficiency of adversarial attacks on SNNs using event data, we conducted a comprehensive comparison with several SOTA methods, including Spike-Compatible [26] and SpikeFool [7], which target event attacks. They served as a baseline for binary grid representation attacks. Additionally, we adapted FGSM, a one-step perturbation attack, to suit data for raw event attacks, setting the boundary values at -1 and 1.

In the context of adversarial attacks on SNNs, it is crucial to delineate the specific nature of the 'event' that each algorithm targets. Unlike the grid representation, we targeted adversarial attacks in the full vision pipeline, processing data in a COO format. This allows direct modification of event polarity from raw event data. As a reference, we provided extra experiments on grid representation with Spike-Compatible and SpikeFool.

**Table 2:** Parameters and model accuracy on the different datasets.

| | CIFAR10-DVS | DVS gesture | NMNIST |
|---|---|---|---|
| Input Size | 2×128×128 | 2×128×128 | 2×34×34 |
| SEW ResNet-34 | 67.20% | 89.06% | 99.06% |
| Spiking ResNet-18 | 68.36% | 93.75% | 98.95% |
| SEW ResNet-34† | 63.10% | 93.75% | 98.84% |
| Spiking ResNet-18† | 68.16% | 94.53% | 95.75% |

The frame number is 16 and the event is split by number.
† Models trained on a binary frame.

### 4.2 Experimental Results

**Performance on Raw Events.** To our knowledge, this is the first study exploring adversarial attacks on raw events in SNNs. Thus, we adopted FGSM as a baseline for raw event attacks. To ensure a fair comparison, we incorporated location indices from targeted samples into FGSM for targeted attacks, which can effectively improve the ASR of FGSM. As demonstrated in Tab. 3, our proposed algorithm significantly outperforms FGSM, achieving success rates that are 50%-70% higher across all datasets and models, which demonstrate the effectiveness of our proposed method in targeted attacks.

Direct application of gradient optimization, as in FGSM, results in notable inaccuracies with raw events. While effective for ANNs, FGSM's performance is reduced in SNNs, emphasizing the challenges in SNN adversarial attacks. However, as indicated in Tab. 3, our application of the Gumbel-Softmax technique parameterizes the binary space, enhancing ASR effectively. The $\Delta$ metric, representing the proportion of events added or removed relative to the total number of adversarial samples, shows our method's superior performance in terms of event ratio compared to FGSM. The visual representation of DVS data is illustrated in Fig. 3, where we display sample effects from both the NMNIST and DVS gesture datasets, with more in the appendix.

The Mean-Square Error (MSE) metric reveals the disparity between adversarial and original sample values, with its definition detailed subsequently,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}[\mathcal{E}^*_{\text{adv}} - \mathcal{E}^*_0]^2_i, \tag{18}$$

where $n$ is the dimension of $\mathcal{E}^*$. The maximum of MSE is 4 because $\mathcal{E}^* \in \{-1, 0, 1\}^n$. Thus, we obtain more subtle perturbations on the adversarial samples in the raw values.

**Performance on Binary Grid Representations.** To facilitate a fair and comprehensive evaluation against the baseline algorithm, we included results from attacks on binary grid representations for reference. For this purpose, the model was retrained using binary grid representations, with detailed results presented in Tab. 2. Performance comparisons on SEW ResNet-34 and Spiking ResNet-18 are shown in Tab. 4, respectively.

During the experiment, we noted that SpikeFool struggled with disappearing gradients. This issue recurred frequently during attacks, resulting in subpar performance

**Table 3:** Attack performance on raw event data.

| Dataset | Method | SEW ResNet-34 | | | Spiking ResNet-18 | | |
|---|---|---|---|---|---|---|---|
| | | ASR | MSE | $\Delta$ | ASR | MSE | $\Delta$ |
| $\mathcal{C}$ | FGSM | 19.70% | 0.87 | 34% | 16.90% | 0.96 | 40% |
| | Ours | **72.23%** | **0.76** | **15%** | **64.79%** | **0.74** | **17%** |
| $\mathcal{G}$ | FGSM | 15.56% | 0.98 | 64% | 13.33% | 0.82 | 49% |
| | Ours | **86.67%** | **0.76** | **19%** | **62.22%** | **0.67** | **19%** |
| $\mathcal{N}$ | FGSM | 11.22% | 0.83 | 53% | 13.27% | 0.82 | 66% |
| | Ours | **53.06%** | **0.77** | **16%** | **78.57%** | **0.77** | **16%** |

$\mathcal{C}$: CIFAR10-DVS; $\mathcal{G}$: DVS gesture; $\mathcal{N}$: NMNIST.

**Table 4:** Attack performance on binary grid representation.

| Method | SEW ResNet-34† | | | Spiking ResNet-18† | | |
|---|---|---|---|---|---|---|
| | $\mathcal{C}$ | $\mathcal{G}$ | $\mathcal{N}$ | $\mathcal{C}$ | $\mathcal{G}$ | $\mathcal{N}$ |
| SpikeFool | 29.60% | 37.62% | 39.80% | 12.97% | 38.18% | 25.74% |
| Spike-Compatible | **96.98%** | 96.13% | **99.84%** | **92.86%** | **96.78%** | **94.86%** |
| Ours | 95.45% | **97.85%** | 98.10% | 91.80% | 95.74% | 86.87% |

$\mathcal{C}$: CIFAR10-DVS; $\mathcal{G}$: DVS gesture; $\mathcal{N}$: NMNIST.

in targeted attacks. This performance was notably inferior compared to the other two methods evaluated. Conversely, Spike-Compatible effectively addressed the gradient vanishing issue by employing a Restricted Spike Flipper. In addition, by transforming continuous gradients into discrete values and utilizing other techniques, it achieved commendable attack performance.

Notably, our method also excelled in attacking binary grid representations. It's important to note that binary grid representation is inherently simpler than raw event processing, as indicated in Tab. 4. However, in scenarios where the online model only provides the interface to raw event data, the utility of methods like SpikeFool and Spike-Compatible becomes invalid.

**Impact of Straight-Through Estimator.** We examined the impact of the Straight-Through estimator on adversarial ASR. The optimization of DVS data presents difficulties for gradient-based approaches due to its inherent discrete nature. Continuous gradient optimization is achieved by using an estimator while maintaining discrete values during forward transfer.

We provided the result without the straight-through estimator in Tab. 5, achieving a 100% ASR on both the DVS gesture and NMNIST datasets, which highlights the gap between continuous and discrete values. The success of the 'soft' version approach in our research highlights a potential vulnerability within SNNs. Despite their unique architecture, this suggests that SNNs' security may be breached under specific conditions.

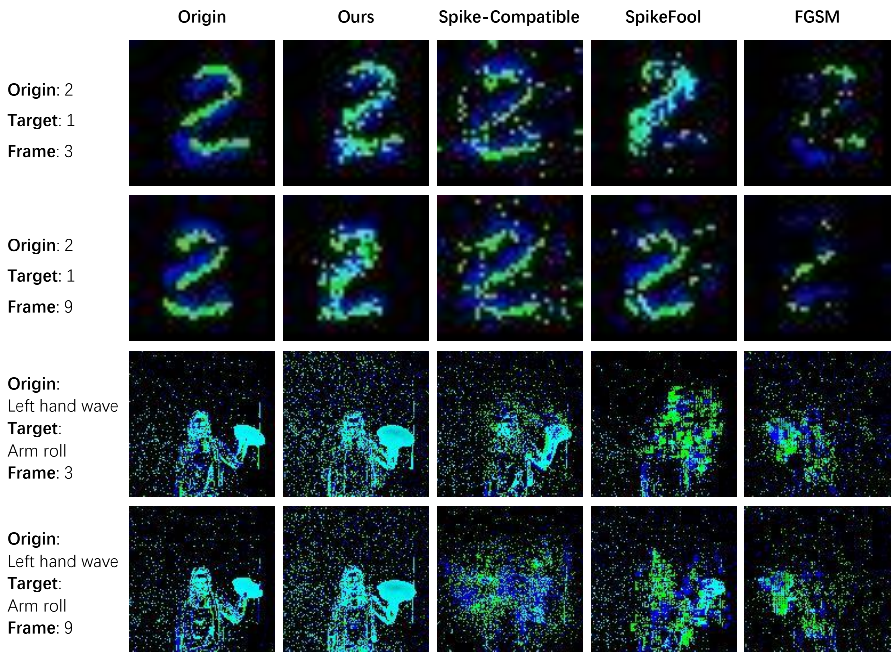



**Fig. 4:** Visualization of NMNIST and DVS gesture datasets across selected frames. Frames 3 and 9 are displayed for each method and dataset (Frame 3 in lines 1 and 3; Frame 9 in lines 2 and 4). Different color channels are indicated by blue and green, with brighter colors signifying multiple pulse emissions at the same location within a time window.

## 5 Related Work

**Continuous Grid Representation Attack.** Not all research targeting SNN attacks has been focused on event datasets. For instance, Rate Gradient Approximation Attack [5] is tailored for rate-coded SNNs with the rate gradient approximation method and hinges on the absence of temporal information in such coding schemes. The Spiking Feature Transferable Attack (SFTA) [29] aims to enhance the transferability of adversarial samples by harnessing robust spike features from intermediate layers. Additionally, the Spike Probabilistic Attack (SPA) discusses efficiency by leveraging Poisson encoding for adversarial sample generation. These approaches rely on optimizing within continuous, dense input. Thus, event data are transformed into grid representations in their work.

**Binary Grid Representation Attack.** The discrete nature of event data presents unique challenges for adversarial attacks. DVS-attack [33] introduces a series of heuristic attack methods that sidestep the need for optimization. Spike-Compatible [26] optimizes within the discrete space by mapping continuous gradients to discrete gradients suitable for spike-based processing, employing a Gradient-to-Spike Converter that probabilistically samples from continuous gradients, extracts the gradient sign, and clips

**Table 5:** Impact of Straight-Through Estimator on ASR. 'Soft' uses continuous values, while 'Hard' uses binary values.

| Dataset | SEW ResNet-34 | | Spiking ResNet-18 | |
|---|---|---|---|---|
| | Soft | Hard | Soft | Hard |
| CIFAR10-DVS | **96.97%** | 72.23% | **88.73%** | 64.79% |
| DVS gesture | **100.00%** | 86.67% | **100.00%** | 62.22% |
| NMNIST | **100.00%** | 53.06% | **100.00%** | 78.57% |

the updated sample to $\{-1, 1\}$. SpikeFool [6] adapts traditional ANN attack methodologies, such as DeepFool [35] and Adversarial Patches [4], by rounding intermediate results to integers and utilizing a straight-through estimator to facilitate gradient updates. Nonetheless, these methods typically do not operate directly on raw event data; instead, they optimize over binary grid representations that integrate event data across time, thus circumventing the high temporal resolution of raw event data.

**Transferability to Raw Event Data.** Although some studies can reconstruct raw event data from grid representations (for example, using Poisson coding to generate spike sequences from grid data), they do not directly optimize raw event data. Consequently, the sparsity of both the perturbation and the adversarial sample is neither guaranteed nor controllable, underscoring a gap in direct raw event data attack methodologies. Furthermore, grid-based representations might not be utilized in the study of SNNs [18], rendering attacks targeting grid representations ineffective.

**Conversion from Grid Representation to Raw Event.** The generation of grid representations from raw event data encompasses many methodologies, thoroughly outlined in Sec. 2.1. When employing a cumulative kernel, it is relatively feasible to find raw event data corresponding to a grid-based representation. However, the process of converting the grid-based representation back into its raw event data by other kernel functions or ANN-based techniques [15] is non-trivial. This complexity implies that attack strategies developed for grid-based representations cannot be seamlessly adapted or directly applied to raw event data.

## 6 Conclusion

This study advances adversarial attacks on SNNs processing event data. We shifted focus from grid-based data representations to directly targeting raw event data, addressing challenges of high temporal resolution and discrete nature. Our approach, using the Gumbel-Softmax sampling strategy and efficient position selection, effectively transformed the discrete optimization problem into a continuous form, enabling efficient targeting of specific spike positions while maintaining data sparsity, as well as reducing memorial and computational burden compared to optimizing on full event space. Experimental results demonstrate the effectiveness of our proposed method in achieving targeted attacks, showcasing its potential in the field of adversarial machine learning on SNNs.

## Acknowledgements

This work is partially supported by the research project "ENERGY-BASED PROBING FOR SPIKING NEURAL NETWORKS" performed at Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), in collaboration with Technology Innovation Institute (TII) (Contract No. TII/ARRC/2073/2021).


## References


1. Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al.: A low power, fully event-based gesture recognition system. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7243–7252 (2017)
2. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: European conference on computer vision. pp. 484–501. Springer (2020)
3. Bader, B.W., Kolda, T.G.: Efficient matlab computations with sparse and factored tensors. SIAM Journal on Scientific Computing **30**(1), 205–231 (2008)
4. Brown, T.B., Mané, D., Roy, A., Abadi, M., Gilmer, J.: Adversarial patch. arXiv preprint arXiv:1712.09665 (2017)
5. Bu, T., Ding, J., Hao, Z., Yu, Z.: Rate gradient approximation attack threats deep spiking neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7896–7906 (2023)
6. Büchel, J., Lenz, G., Hu, Y., Sheik, S., Sorbaro, M.: Adversarial attacks on spiking convolutional neural networks for event-based vision. arXiv preprint arXiv:2110.02929 (2021)
7. Büchel, J., Lenz, G., Hu, Y., Sheik, S., Sorbaro, M.: Adversarial attacks on spiking convolutional neural networks for event-based vision. arXiv preprint arXiv:2110.02929 (2021)
8. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. Ieee (2017)
9. Deng, L.: The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine **29**(6), 141–142 (2012)
10. Dong, X., Chen, D., Bao, J., Qin, C., Yuan, L., Zhang, W., Yu, N., Chen, D.: Greedyfool: Distortion-aware sparse adversarial attack. Advances in Neural Information Processing Systems **33**, 11226–11236 (2020)
11. Fang, W., Chen, Y., Ding, J., Yu, Z., Masquelier, T., Chen, D., Huang, L., Zhou, H., Li, G., Tian, Y.: Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. Science Advances **9**(40), eadi1480 (2023)
12. Fang, W., Yu, Z., Chen, Y., Huang, T., Masquelier, T., Tian, Y.: Deep residual learning in spiking neural networks. Advances in Neural Information Processing Systems **34**, 21056–21069 (2021)
13. Fang, W., Yu, Z., Chen, Y., Huang, T., Masquelier, T., Tian, Y.: Deep residual learning in spiking neural networks. Advances in Neural Information Processing Systems **34**, 21056–21069 (2021)
14. Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., et al.: Event-based vision: A survey. IEEE transactions on pattern analysis and machine intelligence **44**(1), 154–180 (2020)
15. Gehrig, D., Loquercio, A., Derpanis, K.G., Scaramuzza, D.: End-to-end learning of representations for asynchronous event-based data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5633–5643 (2019)

16. Gerstner, W., Kistler, W.M., Naud, R., Paninski, L.: Neuronal dynamics: From single neurons to networks and models of cognition. Cambridge University Press (2014)
17. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
18. Hagenaars, J., Paredes-Vallés, F., De Croon, G.: Self-supervised learning of event-based optical flow with spiking neural networks. Advances in Neural Information Processing Systems **34**, 7167–7179 (2021)
19. He, Y., Zhu, L., Zhu, Y., Chen, C., Jiang, S., Liu, R., Shi, Y., Wan, Q.: Recent progress on emerging transistor-based neuromorphic devices. Advanced Intelligent Systems **3**(7), 2000210 (2021)
20. Izhikevich, E.M.: Which model to use for cortical spiking neurons? IEEE transactions on neural networks **15**(5), 1063–1070 (2004)
21. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)
22. Krithivasan, S., Sen, S., Rathi, N., Roy, K., Raghunathan, A.: Efficiency attacks on spiking neural networks. In: Proceedings of the 59th ACM/IEEE Design Automation Conference. pp. 373–378 (2022)
23. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: Artificial intelligence safety and security, pp. 99–112. Chapman and Hall/CRC (2018)
24. Lee, J.H., Delbruck, T., Pfeiffer, M.: Training deep spiking neural networks using backpropagation. Frontiers in neuroscience **10**, 508 (2016)
25. Li, H., Liu, H., Ji, X., Li, G., Shi, L.: Cifar10-dvs: an event-stream dataset for object classification. Frontiers in neuroscience **11**, 309 (2017)
26. Liang, L., Hu, X., Deng, L., Wu, Y., Li, G., Ding, Y., Li, P., Xie, Y.: Exploring adversarial attack in spiking neural networks with spike-compatible gradient. IEEE transactions on neural networks and learning systems (2021)
27. Liang, L., Xu, K., Hu, X., Deng, L., Xie, Y.: Toward robust spiking neural network against adversarial perturbation. Advances in Neural Information Processing Systems **35**, 10244–10256 (2022)
28. Lichtsteiner, P., Posch, C., Delbruck, T.: A $128\times128$ 120 db $15\mu$s latency asynchronous temporal contrast vision sensor. IEEE journal of solid-state circuits **43**(2), 566–576 (2008)
29. Lin, X., Dong, C., Liu, X.: Sfta: Spiking neural networks vulnerable to spiking feature transferable attack. In: 2022 IEEE 21st International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS). pp. 140–149. IEEE (2022)
30. Lin, X., Dong, C., Liu, X., Zhang, Y.: Spa: An efficient adversarial attack on spiking neural networks using spike probabilistic. In: 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid). pp. 366–375. IEEE (2022)
31. Lin, Y., Ding, W., Qiang, S., Deng, L., Li, G.: Es-imagenet: A million event-stream classification dataset for spiking neural networks. CoRR **abs/2110.12211** (2021), https://arxiv.org/abs/2110.12211
32. 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 of the IEEE conference on computer vision and pattern recognition. pp. 5419–5427 (2018)
33. Marchisio, A., Pira, G., Martina, M., Masera, G., Shafique, M.: Dvs-attacks: Adversarial attacks on dynamic vision sensors for spiking neural networks. In: 2021 International Joint Conference on Neural Networks (IJCNN). pp. 1–9. IEEE (2021)
34. Mohemmed, A., Schliebs, S., Kasabov, N.: Span: A neuron for precise-time spike pattern association. In: Neural Information Processing: 18th International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings, Part II 18. pp. 718–725. Springer (2011)

35. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)
36. Orchard, G., Jayawant, A., Cohen, G.K., Thakor, N.: Converting static image datasets to spiking neuromorphic datasets using saccades. Frontiers in neuroscience **9**, 437 (2015)
37. Orchard, G., Jayawant, A., Cohen, G.K., Thakor, N.: Converting static image datasets to spiking neuromorphic datasets using saccades. Frontiers in neuroscience **9**, 437 (2015)
38. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
39. Ponulak, F.: Resume-new supervised learning method for spiking neural networks. institute of control and information engineering, poznoń university of technology. Tech. rep., Tech. rep (2005)
40. Vervliet, N., Debals, O., De Lathauwer, L.: Tensorlab 3.0—numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization. In: 2016 50th Asilomar Conference on Signals, Systems and Computers. pp. 1733–1738. IEEE (2016)
41. Vervliet, N., Debals, O., De Lathauwer, L.: Tensorlab 3.0—numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization. In: 2016 50th Asilomar Conference on Signals, Systems and Computers. pp. 1733–1738. IEEE (2016)
42. Zhou, Z., Zhu, Y., He, C., Wang, Y., Yan, S., Tian, Y., Yuan, L.: Spikformer: When spiking neural network meets transformer. arXiv preprint arXiv:2209.15425 (2022)
43. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Ev-flownet: Self-supervised optical flow estimation for event-based cameras. arXiv preprint arXiv:1802.06898 (2018)
44. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 989–997 (2019)
45. Zhu, J., Zhang, T., Yang, Y., Huang, R.: A comprehensive review on emerging artificial neuromorphic devices. Applied Physics Reviews **7**(1) (2020)

## Ethical Influence

This study explores adversarial attacks against event-based DVS data and SNNs. This method can manipulate input data so that models produce false outputs, affecting the security of individuals and organizations using these techniques. Compared with traditional grid-based attack methods, this method acts directly on raw event data, providing a more direct and potentially more effective attack approach. This innovation can lead to new security vulnerabilities that need to be considered when designing and deploying SNNs.

## Appendix Outline

The organization of the Appendix is as follows:

– Section A provides the setting of hyper parameters.

- Section B provides additional results on additional datasets and models aside from the main paper and untargeted attack results.
- Section C details the statistical results of perturbation on the raw event dataset.
- Section D provides an ablation study on different hyper-parameters in the experiment, including $m$, $\kappa$, and range of $\alpha$.

# A Hyper Parameters

Before discussing the additional experiment, let's define some symbols and parameter Settings (as shown in Tab. 6). For hyperparameters such as learning rate, we selected values using 5-fold cross-validation that maximizes the attack success rate (ASR).

**Table 6:** Parameter setting.

| Parameter | Value | Meaning |
|---|---|---|
| $\gamma^{\dagger}$ | 1 | Learning rate. |
| $\lambda$ | 0.1 | Regular term coefficient. |
| $m$ | 40 | The total number of samples at a time. |
| $\kappa^{\dagger}$ | 2 | Gumbel-Softmax temperature coefficient. |

† Cosine annealing was used.

# B Additional Results

## B.1 Untargeted Attacks

We present further findings for untargeted attacks. The data in Tab. 7 illustrate our method's performance on raw event datasets using the SEW ResNet-34 and Spiking ResNet-18 models, respectively. Additionally, Tab. 8 provides insights into performance on binary grid representation. These results collectively demonstrate that our approach not only excels in handling raw event datasets but also maintains effectiveness with binary grid representation in untargeted attack scenarios.

## B.2 Performance on Spiking VGG-11 Model

In this section, our focus was on the Spiking VGG-11 model and three DVS datasets. Consistent with the main paper, we also conducted comparative experiments with FGSM on raw events. We detail the training parameters and the model's accuracy in Tab. 9. The performance on the Spiking VGG-11 model is presented in Tab. 10. For untargeted attacks, we modified the number of events by no more than 15%, achieving a 100% success rate on the NMNIST dataset. In targeted attacks, we adjusted the number of events by up to 20%, significantly outperforming the FGSM method, which proved ineffective on the NMNIST dataset.

**Table 7:** Comparison of our method vs. FGSM in the untargeted scenario, with $\Delta$ indicating event addition or removal in adversarial samples.

| Dataset | Method | SEW ResNet-34 | | | Spiking ResNet-18 | | |
|---|---|---|---|---|---|---|---|
| | | ASR | MSE | $\Delta$ | ASR | MSE | $\Delta$ |
| $\mathcal{C}$ | FGSM | 36.57% | 0.94 | 42% | 34.44% | 0.94 | 26% |
| | Ours | **89.44%** | **0.80** | **14%** | **75.91%** | **0.85** | **15%** |
| $\mathcal{G}$ | FGSM | 36.90% | 1.06 | 26% | 33.87% | 1.23 | 39% |
| | Ours | **91.35%** | **0.91** | **12%** | **83.80%** | **0.82** | **12%** |
| $\mathcal{N}$ | FGSM | 20.51% | 0.91 | 20% | 25.67% | 1.85 | 43% |
| | Ours | **91.84%** | **0.77** | **12%** | **99.49%** | **0.76** | **13%** |

$\mathcal{C}$: CIFAR10-DVS; $\mathcal{G}$: DVS gesture; $\mathcal{N}$: NMNIST.

**Table 8:** Comparison with the attack success rate of the binary grid representation method on different models in the untargeted scenario.

| Method | SEW ResNet-34† | | | Spiking ResNet-18† | | |
|---|---|---|---|---|---|---|
| | $\mathcal{C}$ | $\mathcal{G}$ | $\mathcal{N}$ | $\mathcal{C}$ | $\mathcal{G}$ | $\mathcal{N}$ |
| SpikeFool | 48.61% | 60.40% | 85.15% | 29.70% | 41.69% | 71.29% |
| Spike-Compatible | **97.86%** | **97.99%** | **100%** | 95.96% | **98.12%** | **99.10%** |
| Ours | 95.18% | 97.14% | **100%** | **96.01%** | 97.44% | 98.63% |

$\mathcal{C}$: CIFAR10-DVS; $\mathcal{G}$: DVS gesture; $\mathcal{N}$: NMNIST.

In addition, we provided more visualizations on three DVS datasets, including Fig. 11 on the CIFAR10-DVS dataset, Fig. 10 on the DVS gesture dataset, and Fig. 9 on the NMNIST dataset. We chose 5 frames out of 16, and although there are some event losses, it can be seen from the graph that the perturbations are subtle and these degrees of perturbations are acceptable.

### B.3 Performance on Spikformer Architecture

We extended our experiments to include a state-of-the-art architecture, Spikformer [42], with results displayed in Tab. 11. Also, as SNN is an emerging field, major architectural innovations are limited, with many relying on replacing activation with spiking neurons. Exploring these would likely yield less insightful outcomes.

Experiments on two larger datasets, ESImageNet [31] and NCaltech101 [37], are detailed in Tab. 11. Our approach maintains efficacy across larger scales. Unlike training, adversarial attack overhead generally scales linearly with the problem dimension. By focusing solely on existing events and one additional event's polarity, we significantly reduce overhead by over 1000 times. For context, typical high-resolution raw event data consists of about 800,000 events. Compared to 150,528 pixels in standard ANN adversarial images, the overhead is not unacceptable.

**Table 9:** Parameters and model accuracy on the different datasets.

| | CIFAR10-DVS | DVS gesture | NMNIST |
|---|---|---|---|
| Input Size | 2×128×128 | 2×128×128 | 2×34×34 |
| Spiking VGG-11 | 72.50% | 95.00% | 99.01% |

**Table 10:** Comparison of our method vs. FGSM on Spiking VGG-11, with $\Delta$ indicating event addition or removal in adversarial samples.

| Dataset | Method | Target | | | Untarget | | |
|---|---|---|---|---|---|---|---|
| | | ASR | MSE | $\Delta$ | ASR | MSE | $\Delta$ |
| $\mathcal{C}$ | FGSM | 1.56% | 1.22 | 22% | 10.94% | 0.86 | 26% |
| | Ours | **55.00%** | **0.76** | **16%** | **78.33%** | **0.75** | **13%** |
| $\mathcal{G}$ | FGSM | 6.25% | 1.37 | 31% | 28.12% | 0.84 | 26% |
| | Ours | **52.94%** | **0.77** | **17%** | **82.35%** | **0.85** | **15%** |
| $\mathcal{N}$ | FGSM | 0.00% | – | – | 0.00% | – | – |
| | Ours | **75.00%** | **0.79** | **14%** | **100.00%** | **0.75** | **13%** |

$\mathcal{C}$: CIFAR10-DVS; $\mathcal{G}$: DVS gesture; $\mathcal{N}$: NMNIST.

Our ablation study varied accumulation numbers, shown in Tab. 11. The attack success rate remains stable, confirming that while the capture of temporal dynamics is a concern for trainers, modifying the polarity of events is sufficient to mount effective attacks.

### B.4 Additional Baseline

In this section, we incorporated PGD as an additional baseline. The adversarial samples are rounded after each update due to the discrete nature of event data. The results in Tab. 12 demonstrate that our soft-version attack outperforms PGD. Furthermore, the result of the black-box method using the Zeroth-Order method as a gradient approximation shows the transferability of our method. We also evaluated our method on a robust SNN model [27], with superior performance compared to PGD, as shown in an additional column in Tab. 12.

**Table 11:** Attack Spikformer model with different grid representations on ESImageNet and NCaltech101 datasets. Ours denotes the hard version.

| Dataset | Method | T=16 | T=32 | T=64 |
|---|---|---|---|---|
| ESImageNet | FGSM | 1.58% | 6.34% | 7.93% |
| | Ours | **57.37%** | **58.06%** | **60.93%** |
| NCaltech101 | FGSM | 1.30% | 3.85% | 10.26% |
| | Ours | **64.93%** | **65.38%** | **65.82%** |

**Table 12:** Comparison of different methods on robust model.

| Method | NCaltech101 Robust | NCaltech101 Non-robust | DVS gesture Robust | DVS gesture Non-robust |
|---|---|---|---|---|
| PGD | 55.95% | 63.77% | 52.70% | 78.40% |
| $\text{Ours}_{\text{soften}}$ | **58.01%** | **74.32%** | **52.70%** | **80.18%** |
| $\text{Ours}_{\text{hard}}$ | 38.13% | 59.40% | 46.83% | 65.61% |
| Black box | 31.89% | 46.13% | 39.92% | 46.53% |

## C Event perturbation

In this section, we examine the extent of perturbation in raw event samples during adversarial attacks. Initially, the original sample is represented as a sparse matrix within a full space, characterized by the dimensions (T, frame size, frame size), where 'T' denotes the number of events. The sparsity of these original samples is evident from the **origin ratio**. In targeted attacks, despite adding indices from the target sample for optimization, the adversarial samples maintain a sparsity level only marginally higher than the original, as indicated by the **adv ratio**. It's important to note that this ratio represents the proportion of events relative to the total space in both adversarial and original samples. This observation underscores the effectiveness of our method in minimizing the increase in event count while executing targeted attacks.

Furthermore, the similarity in the statistical zero norms between original and adversarial samples, as shown in Tab. 13, indicates that events are neither significantly lost nor increased during the attack. This highlights the effectiveness of our targeted index addition strategy in handling targeted attacks. Given the variability in event numbers across different datasets and within individual samples, we conducted a thorough analysis on a random selection of 100 correctly classified samples from three datasets. The detailed results of this analysis are presented in Tab. 13.

**Table 13:** The event magnitude of the sample. ALL indicates the number of events in the whole space, origin indicates the number of events in the original sample, and adv indicates the number of events in the adversarial sample.

| | ALL($\times 10^7$) | origin($\times 10^5$) / adv($\times 10^5$) | origin ratio($\times 10^{-4}$) / adv ratio($\times 10^{-4}$) | $\ell_0$($\times 10^5$) / perturbation ratio($\times 10^{-4}$) |
|---|---|---|---|---|
| $\mathcal{C}$ | $330.765 \pm 103.874$ | $2.019 \pm 0.633$ | 0.610 | $1.95 \pm 0.389$ |
| | | $2.67 \pm 0.56$ | 0.808 | 0.591 |
| $\mathcal{G}$ | $523.041 \pm 258.686$ | $3.192 \pm 1.578$ | 0.610 | $3.461 \pm 1.298$ |
| | | $4.507 \pm 1.584$ | 0.862 | 0.662 |
| $\mathcal{N}$ | $0.456 \pm 0.126$ | $0.039 \pm 0.011$ | 8.651 | $0.038 \pm 0.008$ |
| | | $0.053 \pm 0.011$ | 11.5 | 8.53 |

$\mathcal{C}$: CIFAR10-DVS; $\mathcal{G}$: DVS gesture; $\mathcal{N}$: NMNIST.

# D Ablation Experiments

## D.1 Efficient Position Selection

**Indices Proportion of Target Events.** Utilizing the Straight-Through Gumbel-Softmax estimator, we facilitate the optimization of parameters in a continuous space, subsequently generating the necessary event data. Given the constrained variability range for each event in adversarial attacks, the selection of event indices emerges as a critical factor. Strategic location selection markedly enhances the ASR. Data presented in Tab. 14 to Tab. 18 illustrate a substantial influence of increasing index ratios on the ASR across various datasets and models. Notably, a 50% or 70% increase in indices leads to a significant uplift in ASR. The symbol $\Delta$ denotes the perturbation change relative to the adversarial sample size. A smaller $\Delta$ increment signifies a more challenging attack, particularly experiments without indices. Moreover, the Mean Squared Error (MSE) metric quantifies the deviation between the original and adversarial events. An analysis of MSE variations reveals that the discrepancy between adversarial and original events narrows with an increase in indices, indicating a more stealthy attack profile.

**Indices of Random Events and Random Indices of Irrelevant Events.** Moreover, we conducted two experimental series: one involving indices for random events excluding those of the target label ($\mathrm{y}_{RL}$), and the other for random indices not about any events ($\mathrm{y}_{RP}$), with $\mathrm{y}_{target}$ representing indices from target-label events. To generate diverse results, we selected three distinct seeds: 0, 42, and 1024. The findings, as detailed from Tab. 19 to Tab. 21, reveal that ASR for indices associated with added target label events is consistently higher compared to those for other random events and random indices. The results are better only on the NMNIST dataset Spiking Resnet-18 model (as shown in Tab. 21).

**Table 14:** Effect of increasing location proportion of target sample on attack success rate on **CIFAR10-DVS** dataset **SEW Resnet-34** model.

| Proportion | ASR | $\Delta$(%) | MSE |
|---|---|---|---|
| 0% | 8.20% | 0.21±0.00 | 1.06±0.001 |
| 10% | 13.11% | 0.15±0.02 | 1.01±0.018 |
| 30% | 16.39% | 0.07±0.04 | 0.91±0.060 |
| 50% | 26.23% | 0.06±0.05 | 0.84±0.063 |
| 70% | 51.15% | 0.09±0.07 | 0.80±0.070 |
| 100% | 72.23% | 0.15±0.03 | 0.76±0.043 |

## D.2 Influence of Sparsity

This section investigates the impact of $\lambda$, a regularization term in the loss function, on the efficacy of the attack. Theoretically, an increase in $\lambda$ is expected to augment

**Table 15:** Effect of increasing location proportion of target sample on attack success rate on **DVS gesture** dataset **SEW Resnet-34** model.

| Proportion | ASR | $\Delta(\%)$ | MSE |
|---|---|---|---|
| 0% | 20.65% | 0.27±0.08 | 1.29±0.1832 |
| 10% | 26.09% | 0.18±0.08 | 1.19±0.1689 |
| 30% | 53.26% | 0.10±0.06 | 1.01±0.1931 |
| 50% | 64.13% | 0.11±0.08 | 0.92±0.2 |
| 70% | 75.00% | 0.13±0.09 | 0.88±0.2021 |
| 100% | 86.67% | 0.19±0.07 | 0.76±0.2607 |

**Table 16:** Effect of increasing location proportion of target sample on attack success rate on **NMNIST** dataset **SEW Resnet-34** model.

| Proportion | ASR | $\Delta(\%)$ | MSE |
|---|---|---|---|
| 0% | 25.71% | 0.24±0.06 | 1.17±0.1753 |
| 10% | 34.69% | 0.17±0.06 | 1.09±0.1589 |
| 30% | 35.71% | 0.08±0.06 | 1.00±0.1869 |
| 50% | 38.65% | 0.15±0.17 | 0.93±0.1731 |
| 70% | 46.73% | 0.28±0.22 | 0.89±0.1731 |
| 100% | 53.06% | 0.16±0.05 | 0.77±0.1427 |

the similarity between the adversarial and original samples, potentially reducing the attack's success rate. However, from the result in Fig. 5, as $\lambda$ increases, there is a noticeable decline in the attack success rate across most experimental groups. Meanwhile, the number of events added or removed in the adversarial sample remains largely consistent, which can be attributed to the sampling process. Although the regular term $\lambda$ does not significantly influence the sparsity of events, it is observed that the original samples are sparse. From the result in Fig. 5, we can obtain that the change in the number of events is stable and sparse. For a more comprehensive analysis of the perturbations, please refer to Tab. 13.

### D.3 Parametric Ablation

**Influence of $\kappa$.** The influence of $\kappa$ is provided in Fig. 7. The results indicate that the influence of $\kappa$ on the attack performance is significant. Since $\kappa$ controls the overall closeness of approximation to the categorical distribution. A lower $\kappa$ value will result in an abrupt change of gradient, whereas a higher $\kappa$ will result in a notable dispatch between the input value and the value for gradient calculation during the approximation process. Therefore, $\kappa$ is pivotal in balancing the gradient's effective updating with the accurate approximation of the sampling distribution.

**Table 17:** Effect of increasing location proportion of target sample on attack success rate on **DVS gesture** dataset **Spiking Resnet-18** model.

| Proportion | ASR | $\Delta(\%)$ | MSE |
|---|---|---|---|
| 0% | 3.30% | 0.22±0.02 | 1.12±0.1042 |
| 10% | 5.49% | 0.16±0.05 | 1.03±0.1399 |
| 30% | 8.79% | 0.19±0.21 | 0.89±0.1503 |
| 50% | 26.37% | 0.14±0.09 | 0.78±0.1072 |
| 70% | 40.66% | 0.15±0.09 | 0.75±0.0951 |
| 100% | 62.22% | 0.19±0.06 | 0.67±0.0619 |

**Table 18:** Effect of increasing location proportion of target sample on attack success rate on **NMNIST** dataset **Spiking Resnet-18** model.

| Proportion | ASR | $\Delta(\%)$ | MSE |
|---|---|---|---|
| 0% | 55.10% | 0.23±0.03 | 1.15±0.1526 |
| 10% | 61.22% | 0.17±0.05 | 1.09±0.1674 |
| 30% | 66.33% | 0.08±0.07 | 0.90±0.1369 |
| 50% | 69.39% | 0.10±0.10 | 0.97±0.1530 |
| 70% | 75.51% | 0.19±0.16 | 0.86±0.1481 |
| 100% | 78.57% | 0.16±0.04 | 0.77±0.1830 |

Furthermore, Fig. 6 presents the results of the cosine annealing variant of $\kappa^\dagger$. The cosine annealing formula is defined as

$$\kappa_i^\dagger = \begin{cases} \kappa_{init} + \frac{1}{2} \times (\kappa_{\text{final}} - \kappa_{\text{init}}) \times \hat{\kappa}, & \text{if } i < \kappa_{\text{decay}}, \\ \kappa_{init}, & \text{otherwise}, \end{cases} \tag{19}$$

where $\hat{\kappa} = \left(1 + \cos\left(\frac{\pi \times i}{\kappa_{\text{decay}}}\right)\right)$, and $\kappa_{\text{init}} \leq \kappa_{\text{final}}$. $i$ is the current number of iterations, $\kappa_{\text{decay}}$ is the number of annealing steps, and $\kappa^\dagger$ remain $\kappa_{\text{init}}$ when the number of iterations reaches this value.

$\kappa$, when combined with cosine annealing, generally yields higher attack success rates compared to configurations without it. Initially, efficient gradients are crucial for optimization during the early stages of the attack. As the attack progresses, it becomes more important for the Gumbel-Softmax sampling outcomes to closely resemble the classification distribution. To achieve this balance, we employ cosine annealing to optimize the temperature parameters, effectively adapting $\kappa$ throughout different stages of the attack. For the $\kappa^\dagger$ parameter combined with cosine annealing, it is evident from Fig. 6 that setting $\kappa^\dagger$ to a value of 2 results in a more efficient attack. This observation highlights the optimal setting for $\kappa^\dagger$ in enhancing the efficacy of our adversarial attack strategy.

**Influence of $m$.** This section presents an ablation study focusing on the influence of the sample number, denoted as $m$. Detailed results of this study are provided in

**Table 19:** Performance of adding indices with target label events and adding indices with non-target events on CIFAR10-DVS dataset.

| | | SEW Resnet-34 | | | Spiking Resnet-18 | | |
|---|---|---|---|---|---|---|---|
| | SEED | ASR | $\Delta$ | MSE | ASR | $\Delta$ | MSE |
| $y_{target}$ | – | 72.23% | 0.15±0.03 | 0.76±0.05 | 64.79% | 0.17±0.04 | 0.74±0.09 |
| $y_{RL}$ | 0 | 24.59% | 0.12±0.09 | 0.77±0.09 | 15.94% | 0.10±0.05 | 0.81±0.09 |
| $y_{RP}$ | 0 | 21.31% | 0.13±0.01 | 0.78±0.06 | 26.09% | 0.13±0.01 | 0.79±0.06 |
| $y_{RL}$ | 42 | 22.86% | 0.16±0.05 | 0.74±0.07 | 20.78% | 0.14±0.06 | 0.75±0.07 |
| $y_{RP}$ | 42 | 21.43% | 0.12±0.00 | 0.77±0.04 | 20.78% | 0.13±0.02 | 0.80±0.08 |
| $y_{RL}$ | 1024 | 21.43% | 0.14±0.05 | 0.76±0.05 | 18.46% | 0.11±0.04 | 0.79±0.08 |
| $y_{RP}$ | 1024 | 25.71% | 0.12±0.00 | 0.76±0.05 | 20.00% | 0.13±0.02 | 0.79±0.08 |

**Table 20:** Performance of adding indices with target label events and adding indices with non-target events on DVS gesture dataset.

| | | SEW Resnet-34 | | | Spiking Resnet-18 | | |
|---|---|---|---|---|---|---|---|
| | SEED | ASR | $\Delta$ | MSE | ASR | $\Delta$ | MSE |
| $y_{target}$ | – | 86.67% | 0.19±0.07 | 0.76±0.26 | 62.22% | 0.19±0.06 | 0.67±0.07 |
| $y_{RL}$ | 0 | 28.26% | 0.18±0.11 | 0.92±0.13 | 8.79% | 0.23±0.13 | 0.68±0.12 |
| $y_{RP}$ | 0 | 23.91% | 0.12±0.02 | 0.84±0.11 | 9.89% | 0.12±0.00 | 0.80±0.07 |
| $y_{RL}$ | 42 | 31.52% | 0.16±0.13 | 0.97±0.17 | 6.45% | 0.21±0.11 | 0.70±0.10 |
| $y_{RP}$ | 42 | 20.65% | 0.13±0.02 | 0.86±0.12 | 10.75% | 0.12±0.00 | 0.77±0.04 |
| $y_{RL}$ | 1024 | 22.22% | 0.31±0.25 | 0.92±0.04 | 6.38% | 0.16±0.12 | 0.77±0.15 |
| $y_{RP}$ | 1024 | 20.43% | 0.13±0.03 | 0.83±0.11 | 14.89% | 0.12±0.00 | 0.77±0.03 |

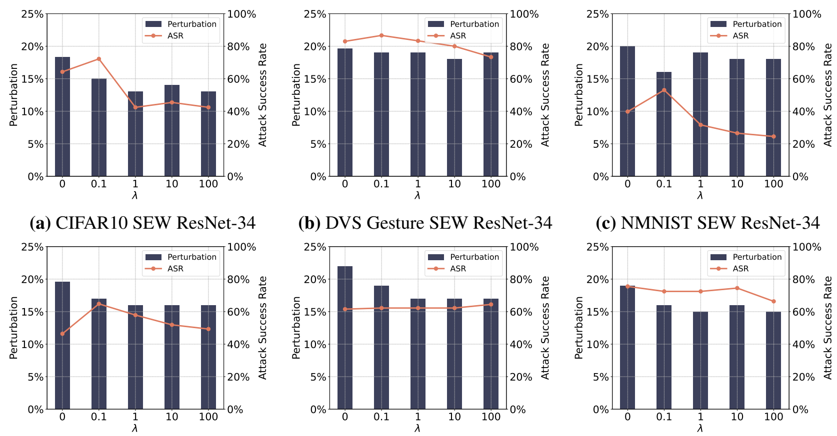
Tab. 22. The study reveals that employing a greater number of samples in each iteration leads to a reduction in the overall variance of the Gumbel distribution. This decrease in variance is associated with improved performance outcomes. The relationship between the increased sample number and the resulting enhancement in results underscores the importance of sample size in the effectiveness of the iteration process.

**Influence of $\alpha$ Range.** This section outlines an ablation study concerning the range of $\alpha$, denoted as $\alpha^*$. In the course of optimization, $\alpha$ is constrained within the range $[-\alpha^*, \alpha^*]$. Detailed results are provided in Fig. 8. Analysis of these results indicates that setting $\alpha$ to 1 or 2 yields better performance. Conversely, an $\alpha$ value larger than 4 leads to a decrease in the attack success rate, which then stabilizes.

The reason for this phenomenon is the nature of the softmax function. Regarding the softmax function, when its elements are large, exponentiating these elements can produce excessively large values. Such magnitudes pose risks of numerical instability and can cause the gradient to vanish. Therefore, it is crucial to maintain $\alpha$ within a reasonable range to prevent the disappearance of the gradient.

**Table 21:** Performance of adding indices with target label events and adding indices with non-target events on NMNIST dataset.

| | | SEW Resnet-34 | | | Spiking Resnet-18 | | |
|---|---|---|---|---|---|---|---|
| | SEED | ASR | $\Delta$ | MSE | ASR | $\Delta$ | MSE |
| $y_{target}$ | – | 53.06% | 0.16±0.05 | 0.77±0.15 | 78.57% | 0.16±0.04 | 0.77±0.19 |
| $y_{RL}$ | 0 | 23.47% | 0.15±0.07 | 0.84±0.19 | 61.22% | 0.15±0.09 | 0.79±0.12 |
| $y_{RP}$ | 0 | 20.41% | 0.13±0.01 | 0.79±0.10 | 70.41% | 0.14±0.05 | 0.82±0.13 |
| $y_{RL}$ | 42 | 20.20% | 0.14±0.10 | 0.85±0.15 | 62.00% | 0.15±0.08 | 0.81±0.13 |
| $y_{RP}$ | 42 | 19.19% | 0.13±0.00 | 0.79±0.09 | 66.00% | 0.13±0.03 | 0.79±0.09 |
| $y_{RL}$ | 1024 | 15.31% | 0.16±0.0 6 | 0.82±0.11 | 63.64% | 0.15±0.09 | 0.82±0.14 |
| $y_{RP}$ | 1024 | 16.33% | 0.13±0.03 | 0.80±0.11 | 65.66% | 0.14±0.04 | 0.83±0.13 |



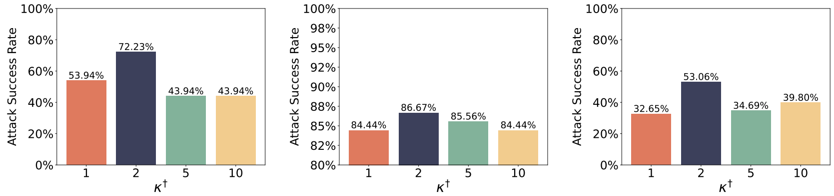

**(a)** CIFAR10 SEW ResNet-34 **(b)** DVS Gesture SEW ResNet-34 **(c)** NMNIST SEW ResNet-34

**(d)** CIFAR10 Spiking ResNet-18 **(e)** DVS Gesture Spiking ResNet-18 **(f)** NMNIST Spiking ResNet-18

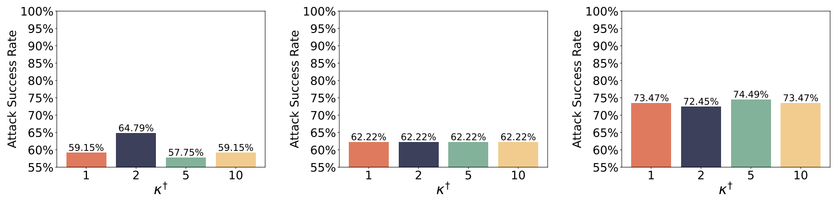
**Fig. 5:** Impact of $\lambda$ on attack success rate and perturbation ratio. ASR represents the attack success rate.

**Table 22:** Influence of varying sampling number per attack on the attack success rate.

| Model | $m$ | Dataset | | |
|---|---|---|---|---|
| | | CIFAR10-DVS | DVS gesture | NMNIST |
| SEW ResNet-34 | 10 | 49.10% | 83.33% | 26.53% |
| | 20 | 49.10% | 85.56% | 26.53% |
| | 30 | 53.94% | 84.44% | 49.59% |
| | 40 | 72.23% | 86.67% | 53.06% |
| Spiking ResNet-18 | 10 | 53.52% | 58.24% | 59.18% |
| | 20 | 54.93% | 62.64% | 65.31% |
| | 30 | 59.15% | 62.64% | 69.39% |
| | 40 | 64.79% | 62.22% | 72.45% |

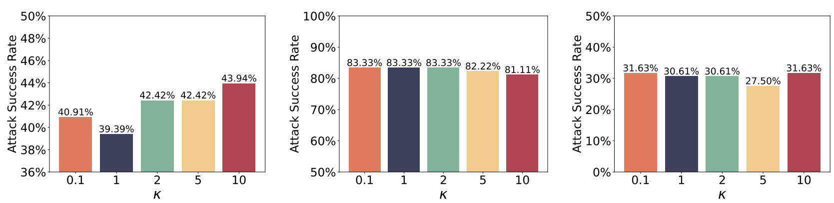



(**a**) CIFAR10 SEW ResNet-34 (**b**) DVS Gesture SEW ResNet-34 (**c**) NMNIST SEW ResNet-34

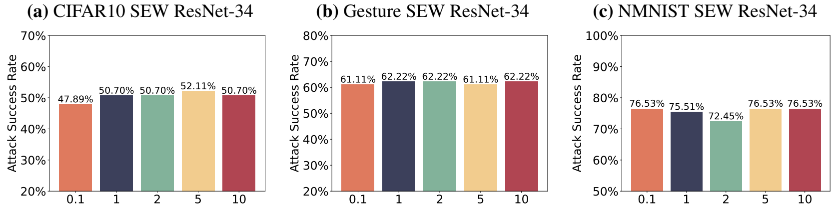



(**d**) CIFAR10 Spiking ResNet-18 (**e**) DVS Gesture Spiking ResNet-18 (**f**) NMNIST Spiking ResNet-18

**Fig. 6:** Impact of cosine annealing version $\kappa^\dagger$ on attack success rate.

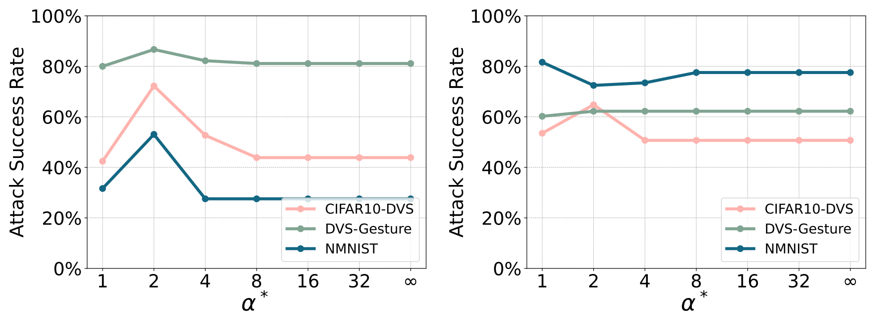



(**a**) CIFAR10 SEW ResNet-34 (**b**) Gesture SEW ResNet-34 (**c**) NMNIST SEW ResNet-34




(**d**) CIFAR10 Spiking ResNet-18 (**e**) Gesture Spiking ResNet-18 (**f**) NMNIST Spiking ResNet-18

**Fig. 7:** Impact of $\kappa$ on attack success rate.




**(a)** SEW ResNet-34 **(b)** Spiking ResNet-18

**Fig. 8:** Impact of $\alpha^*$ range on attack success rate.

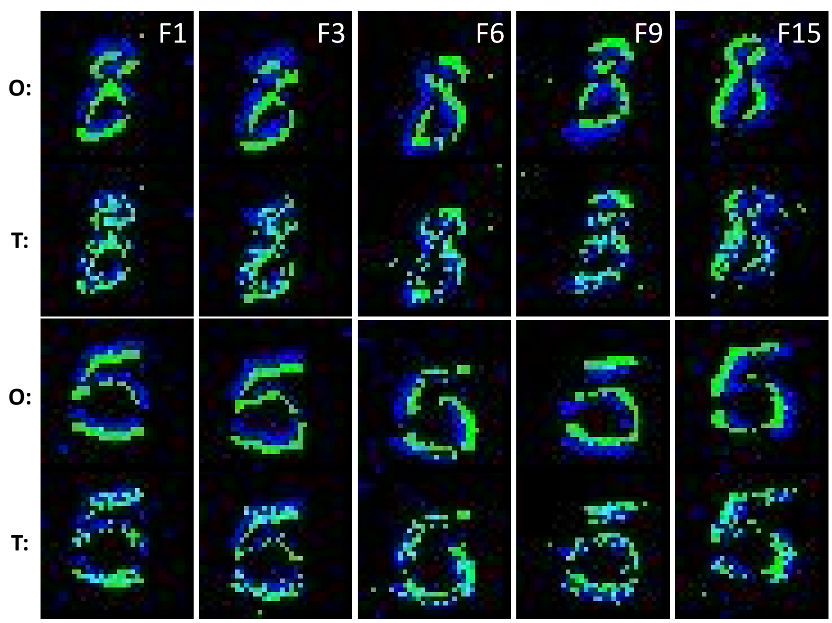



**Fig. 9:** Visualization of the raw event in NMNIST. Sequential frames 1, 3, 6, 9, and 15 are displayed from left to right. O means the original sample, and T means the target sample.

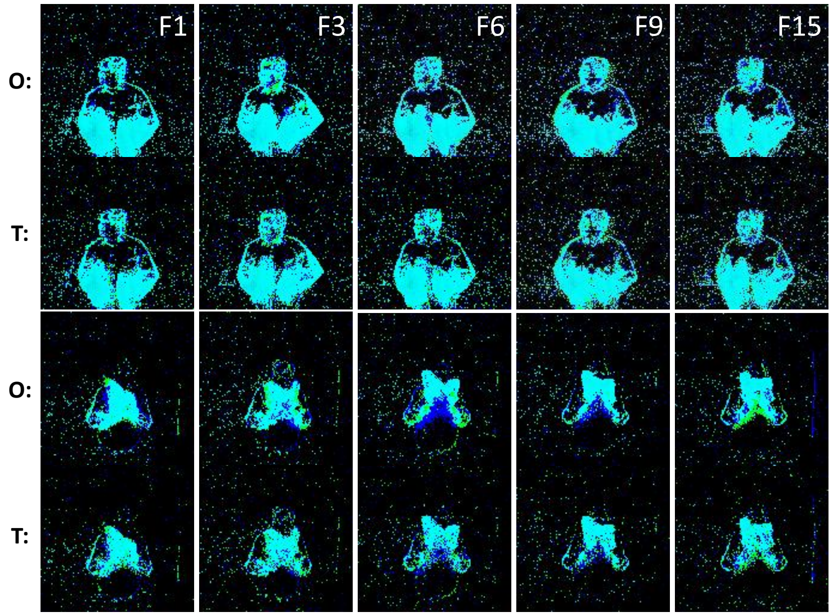



**Fig. 10:** Depiction of the raw events in DVS gesture. Sequential frames 1, 3, 6, 9, and 15 are displayed from left to right. O means the original sample, and T means the target sample.

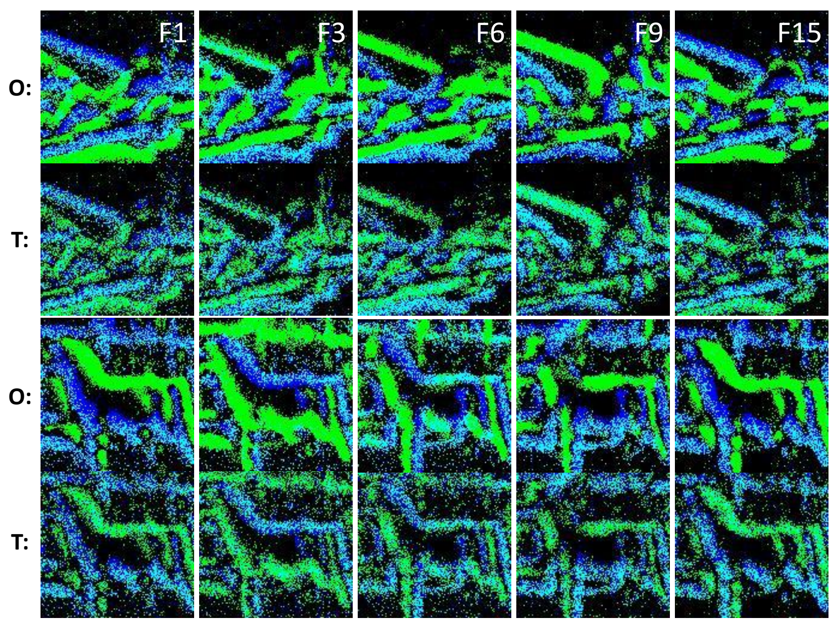



**Fig. 11:** Depiction of the raw events in CIFAR10-DVS. Sequential frames 1, 3, 6, 9, and 15 are displayed from left to right. O means the original sample, and T means the target sample.