This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Towards Effective and Sparse Adversarial Attack on Spiking Neural Networks via Breaking Invisible Surrogate Gradients

Li Lun[1] Kunyu Feng[2] Qinglong Ni[3] Ling Liang[1] Yuan Wang[1]
Ying Li[3*] Dunshan Yu[1] Xiaoxin Cui[1*]
[1]School of Integrated Circuits, Peking University, Beijing, China
[2]School of Software and Microelectronics, Peking University, Beijing, China
[3]Institute of Microelectronics, Chinese Academy of Sciences, Beijing, China
{lunli, lingliang, wangyuan, yuds, cuixx}@pku.edu.cn,
feng_ky21@stu.pku.edu.cn, {niqinglong, liying1}@ime.ac.cn

Abstract

Spiking neural networks (SNNs) have shown their competence in handling spatial-temporal event-based data with low energy consumption. Similar to conventional artificial neural networks (ANNs), SNNs are also vulnerable to gradient-based adversarial attacks, wherein gradients are calculated by spatial-temporal back-propagation (STBP) and surrogate gradients (SGs). However, the SGs may be invisible for an inference-only model as they do not influence the inference results, and current gradient-based attacks are ineffective for binary dynamic images captured by the dynamic vision sensor (DVS). While some approaches addressed the issue of invisible SGs through universal SGs, their SGs lack a correlation with the victim model, resulting in sub-optimal performance. Moreover, the imperceptibility of existing SNN-based binary attacks is still insufficient. In this paper, we introduce an innovative potential-dependent surrogate gradient (PDSG) method to establish a robust connection between the SG and the model, thereby enhancing the adaptability of adversarial attacks across various models with invisible SGs. Additionally, we propose the sparse dynamic attack (SDA) to effectively attack binary dynamic images. Utilizing a generation-reduction paradigm, SDA can fully optimize the sparsity of adversarial perturbations. Experimental results demonstrate that our PDSG and SDA outperform state-of-the-art SNN-based attacks across various models and datasets. Specifically, our PDSG achieves 100% attack success rate on ImageNet, and our SDA obtains 82% attack success rate by modifying only 0.24% of the pixels on CIFAR10DVS. The code is available at https://github.com/ryime/PDSG-SDA.

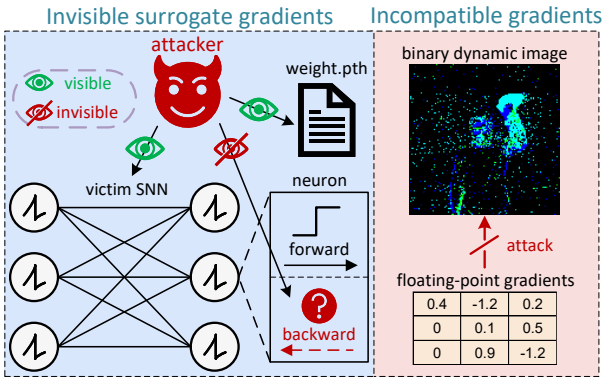


Figure 1. Illustration of the challenges of attacking SNNs. The invisible SGs hinder the attacker to perform gradient-based attacks. The incompatible gradients describe that the floating-point gradients are difficult to be converted to binary perturbations.

1. Introduction

Brain-inspired spiking neural networks (SNNs), as the third generation of neural network models, have attracted extensive attention in machine intelligence and neuromorphic computing [48, 62]. In contrast to floating-point activation in conventional artificial neural networks (ANNs), neurons in SNNs utilize binary spike sequences for communication. The sparse and event-driven natures of spikes endow SNNs with the abilities of asynchronous processing and low energy consumption [35, 60]. Nowadays, SNN-based neuromorphic hardware has emerged as low-power and high-performance edge computing devices in resource-constrained scenarios, such as TrueNorth [1], Loihi [10, 57], Tianjic [59], and Darwin [46, 47].

In practical applications, the security and robustness of neural networks are required to be concerned [25, 84]. Previous studies [22, 49, 65] have highlighted the vulnerability of ANNs to gradient-based adversarial attacks, which

* Corresponding Authors.

craft adversarial examples with imperceptible perturbations to fool the model. In SNNs, gradients can be calculated using spatial-temporal back-propagation (STBP) [68], which adopts the surrogate gradient (SG) to circumvent the non-differentiability of the Heaviside function [55]. However, since the inference only requires the structure and trained parameters of the model, the attacker might not obtain the SG of an inference-only model. As the shape and hyper-parameters of the SG significantly influence the attack success rate [70], choosing an appropriate SG is crucial for enhancing the effectiveness of the attack. Moreover, SNNs with inherent spatial-temporal characteristics excel at handling dynamic images, represented by events captured by the dynamic vision sensor (DVS) [40]. Typically, these events are aggregated into binary frames to maintain compatibility with neuromorphic hardware [74, 78, 81], making gradient-based attacks inapplicable to binary dynamic images due to different input formats [76]. These challenges are depicted in Fig. 1.

To address the issue of invisible SGs, methods such as RGA [5] and HART [26] adopted customized SGs. Nonetheless, their SGs are universal and lack a direct correlation with the victim model. Even when the attacker fortunately obtains and utilizes the SG used during the training phase [64], the performance of the attack remains unstable, as the training-phase SG is dedicated to optimizing the model's parameters rather than the gradient of the input. Consequently, establishing a connection between the attacking-phase SG and the trained model would significantly advance adversarial attacks. Additionally, in the context of binary dynamic images, adversarial perturbations should be sufficiently sparse to evade detection, since the aggregated frames are spatially sparse [66]. Although existing attack paradigms have successfully converted floating-point gradients to binary perturbations [6, 42, 76], they often suffer from gradient vanishing, and the imperceptibility of perturbations remains unsatisfactory. Therefore, an effective sparse attack method is required to thoroughly investigate the robustness of SNNs.

In this paper, we propose the potential-dependent surrogate gradient (PDSG) in adversarial attacks on SNNs. The PDSG can adapt to various models due to its shape dependent on the distribution of the membrane potential, thereby enhancing the threats of adversarial attacks across static and dynamic datasets. We also devise a paradigm for attacking binary dynamic images, named sparse dynamic attack (SDA). SDA iteratively generates significant perturbations and removes redundant perturbations, fully optimizing the imperceptibility of adversarial perturbations. The main contributions of our paper are summarized as follows:

- We propose the potential-dependent surrogate gradient to achieve more representative gradients in adversarial attacks. The PDSG establishes a connection between the SG and the victim model through the run-time distribution of membrane potential. To the best of our knowledge, this is the first time that an adaptive SG is adopted in adversarial attacks on SNNs.
- We introduce the sparse dynamic attack on binary dynamic images, which combines the gradient and the finite difference to craft sparse yet powerful adversarial examples. Our SDA carefully selects the most desirable pixels to attack, effectively improving the sparsity of adversarial perturbations while maintaining high attack success rate.
- We conduct extensive experiments on various datasets and SNN models to substantiate the effectiveness of our PDSG and SDA. On ImageNet dataset, the PDSG achieves 100% attack success rate without knowing the SG of the model. For binary dynamic images on CIFAR10DVS dataset, the SDA obtains 82% attack success rate with only 0.24% pixels perturbed.

2. Related Works

2.1. High Performance Spiking Neural Networks

There are two main types of learning algorithms for achieving high-performance SNNs: converting ANNs to SNNs (ANN2SNN) [4, 29, 37] and directly training SNNs [24, 82]. As ANN2SNN methods have been demonstrated to be more vulnerable to adversarial attacks [63, 64], we only consider directly-trained SNNs in this paper. The widely used algorithm for directly training deep SNNs is STBP [68]. STBP clarifies the propagation process across the spatial and temporal dimension, and the SG is adopted to mitigate the non-differentiable problem of the Heaviside function. Various shapes and parameters of SG are also investigated [38, 41, 67, 77]. To effectively train deep SNNs, threshold-dependent batch normalization (tdBN) [80] is proposed to balance the distribution of the pre-activations and the neuronal threshold, thereby stabilizing the gradient flow. Implementing optimization algorithms, current directly-trained SNNs exhibit low inference latency and comparable accuracy to ANNs on both static and event-based datasets [12, 17, 19, 72, 73, 75].

2.2. Adversarial Attacks and Defenses on Directly-trained SNNs

Directly-trained SNNs are vulnerable to adversarial examples crafted by STBP. *Sharmin et al.* proposed a gradient propagation algorithm for Poisson encoding on static images and demonstrated the power of adversarial attack based on STBP [64]. To optimize the gradient flow, RGA [5] and HART [26] investigated the rate and temporal information of SNNs, and generated more potent adversarial examples through changing the calculation process of STBP. For binary dynamic inputs of SNNs, DVS-attacks [50] first developed various search methods on DVS data to fool the

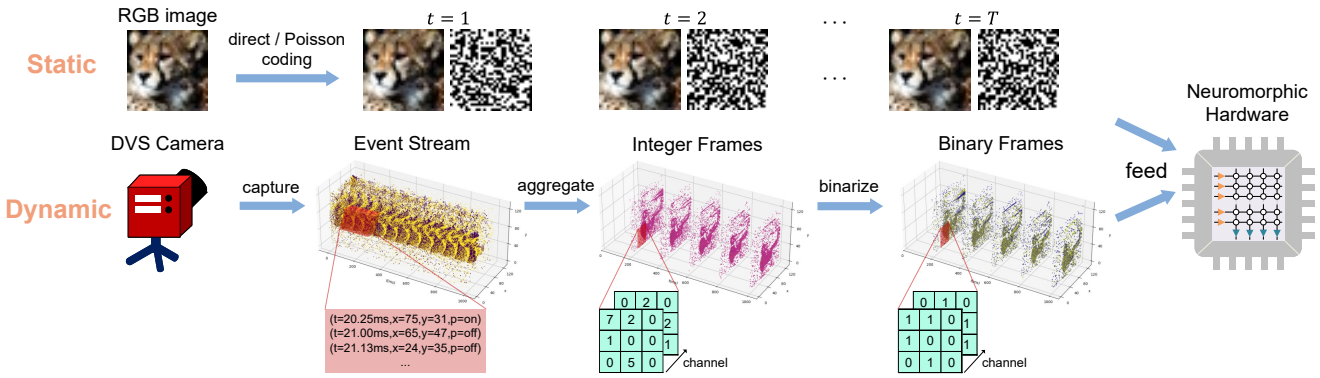


Figure 2. Illustration of the pre-processing procedure for both static and dynamic input of SNNs. For static inputs, the RGB image is encoded by direct or Poisson encoding to extend the temporal dimension. For dynamic inputs, the event stream is captured by DVS, where t denotes the time of the event, x, y are the coordinates, and p is the polarity. Then the event stream is aggregated into integer frames, and each polarity corresponds to a channel. The integer frames will be further binarized to binary frames for hardware compatibility.

SNNs. To generate sparse perturbations, spike-compatible gradient (SCG) [42] converted floating-point gradients to binary spikes through probabilistic sampling. SpikeFool [6] adapted the SparseFool [52] from ANNs to SNNs by rounding the sparse floating-point perturbations to binary values. In addition, Gumbel-softmax attack (GSAttack) [76] directly perturbed the raw event data through the Gumbel-softmax technique.

To mitigate the impact of adversarial attacks, several defense strategies inspired by ANN-based methods are proposed [58]. Certified training is extended to SNNs by investigating the interval bound propagation [43] and randomized smoothing [54]. The weight regularization [13] and gradient regularization [45] are adopted in adversarial training on SNNs. To reach bio-plausible robustness, the dynamics of neurons is reformed through stochastic gating [15] and lateral inhibition [79]. The effects of noise filters on DVS inputs were also discussed [50, 51]. Moreover, various works investigated the inherent robustness of SNNs, such as the leakage factor [14, 18], coding schemes [34, 39], and the firing threshold [18].

3. Preliminaries

3.1. Leaky Integrate-and-Fire Neuron Model

In this paper, we adopt the widely used leaky integrate-and-fire (LIF) neuron model [21] in SNNs. Considering the iterative expression of the LIF model [69], the membrane potential of the l-th layer and t-th timestep is updated by:

$$\boldsymbol{u}^l[t] = \tau \boldsymbol{u}^l[t-1](1-\boldsymbol{s}^l[t-1]) + \boldsymbol{W}^l \boldsymbol{s}^{l-1}[t] + \boldsymbol{b}^l. \quad (1)$$

$\boldsymbol{u}$ represents the membrane potential of the neuron, τ denotes the leakage factor, and $\boldsymbol{W}^l$ and $\boldsymbol{b}^l$ are weight and bias. When the membrane potential reaches the threshold V_{th}, the neuron will fire a spike $\boldsymbol{s}^l$ through the Heaviside function and trigger the reset mechanism. Here we consider the hard reset mechanism, which directly resets the membrane potential to 0. The firing function is described as:

$$\boldsymbol{s}^l[t] = h(\boldsymbol{u}^l[t] - V_{th}) = \begin{cases} 1, & \boldsymbol{u}^l[t] \geq V_{th} \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

The LIF model describes the spatial-temporal characteristic of the neuron. The firing and reset mechanisms introduce the non-linearity to the neuron, enabling SNNs to perform complex tasks.

3.2. Spatial-Temporal Backpropagation

Based on the neurodynamics of the LIF neuron model, STBP algorithm [68] demonstrates that the gradients of the spikes contain both spatial and temporal terms. The gradient of the loss function $\mathcal{L}$ with respect to the spikes $\boldsymbol{s}^l[t]$ in the l-th layer is calculated by:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{s}^l[t]} = & \frac{\partial \mathcal{L}}{\partial \boldsymbol{s}^{l+1}[t]} \frac{\partial \boldsymbol{s}^{l+1}[t]}{\partial \boldsymbol{u}^{l+1}[t]} \frac{\partial \boldsymbol{u}^{l+1}[t]}{\partial \boldsymbol{s}^l[t]} + \\ & \frac{\partial \mathcal{L}}{\partial \boldsymbol{s}^l[t+1]} \frac{\partial \boldsymbol{s}^l[t+1]}{\partial \boldsymbol{u}^l[t+1]} \frac{\partial \boldsymbol{u}^l[t+1]}{\partial \boldsymbol{s}^l[t]}. \end{aligned} \quad (3)$$

The SG is adopted to address the non-differentiable problem of the firing function in Eq. (2), where the derivative $\partial \boldsymbol{s}^l[t] / \partial \boldsymbol{u}^l[t]$ can not be directly calculated. The shape of the SG plays a crucial role in the representation of the gradient. A sharp SG can induce gradient vanishing, wherein most gradients become zero; conversely, a flat SG can cause gradient mismatch, indicating that the gradient fails to accurately reflect the trend of loss changes [23, 41]. This phenomenon implies that the effectiveness of gradient-based adversarial attacks is highly dependent on the SG.

3.3. Input Pre-processing

As SNNs contain calculations across the temporal dimension, the inputs of SNNs must be transformed to align with their spatial-temporal characteristics. For static images, common coding schemes include direct coding and Poisson

coding [30]. In direct coding, the image at every timestep is identical to the input image. Poisson coding generates a binary image at each timestep, following a Bernoulli distribution where the probability is determined by the normalized pixel values. These coding schemes are denoted as:

$$x[t] = \begin{cases} x, & \text{direct coding} \\ x > rand(0,1), & \text{Poisson coding} \end{cases}. \quad (4)$$

For dynamic data (*e.g.* , in DVS-Gesture [2] dataset), the event stream is required to be aggregated into integer frames, with the pixel value representing the count of events within the time interval [71]. As some neuromorphic processors are capable of processing multi-bit inputs, like Tianjic [59] and Loihi2 [57], they can handle both static images and integer dynamic images. However, most neuromorphic processors only accept binary spike inputs [3, 8, 33, 78], which aligns with the fundamental concept of SNNs. In these cases, binary frames for dynamic images are suitable for these processors. The pre-processing procedure is illustrated in Fig. 2. In this paper, we consider adversarial attacks on all these input scenarios for SNNs.

3.4. Adversarial Attack

The objective of an adversarial attack is to generate imperceptible perturbations that can fool the classifier. Given a SNN classifier f, we consider a benign image $\boldsymbol{x}$ with its corresponding label y. In this paper, we focus on the untargeted attack, where the attacker aims to change the classification result to any other label. The optimization problem of the untargeted attack is denoted as:

$$\text{argmax}_{\boldsymbol{\delta}} \mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta}), y), \quad \text{subject to } ||\boldsymbol{\delta}||_p \leq \epsilon. \quad (5)$$

Here, $\boldsymbol{\delta}$ is the perturbation, and $||\boldsymbol{\delta}||_p$ is the ℓ_p-norm of the perturbation. We denote $\boldsymbol{x}_{adv} = \boldsymbol{x}+\boldsymbol{\delta}$ as the adversarial example. For static images and integer dynamic frames, consistent with common adversarial attacks on ANNs, we adopt the ℓ_∞-norm to limit the maximum absolute value of perturbations. For binary dynamic frames, we use the ℓ_0-norm to limit the number of modified pixels, which also represents the sparsity of adversarial perturbations.

In adversarial attacks on ANNs, there are two simple yet effective ℓ_∞-norm attack algorithms: fast gradient sign method (FGSM) [22] and projected gradient descent (PGD) [49]. FGSM leverages the sign of the gradient of the input to craft adversarial examples, which is denoted as:

$$\boldsymbol{x}_{adv} = \boldsymbol{x} + \epsilon \cdot \text{sign}(\frac{\partial \mathcal{L}(f(\boldsymbol{x}), y)}{\partial \boldsymbol{x}}). \quad (6)$$

PGD iteratively performs the FGSM with a small step size α. In the k-th iteration, the input is projected onto the space of the ϵ-ℓ_∞ neighborhood by $\prod_\epsilon$, denoted as:

$$\boldsymbol{x}^{k+1} = \prod_{\epsilon} \{\boldsymbol{x}^k + \alpha \cdot \text{sign}(\frac{\partial \mathcal{L}(f(\boldsymbol{x}^k), y)}{\partial \boldsymbol{x}^k})\}. \quad (7)$$

4. Methods

To address the challenge of invisible SGs when attacking SNNs, we first introduce the potential-dependent surrogate gradient (**PDSG**). Our PDSG incorporates the variance of membrane potential into the shape of SG, thereby adapting to diverse distribution of models. Furthermore, we describe the sparse dynamic attack (**SDA**) to generate sparse perturbations in attacking binary dynamic images. The generation-reduction paradigm of our SDA effectively improves the imperceptibility of adversarial perturbations.

4.1. Potential-Dependent Surrogate Gradient

Due to the non-differential problem of the firing function in SNNs, the attacker is difficult to perform the attack without knowledge of the SG. Selecting an SG blindly or randomly cannot guarantee the performance of the attack. We first explore the relationship between the distribution of membrane potential and the SG to establish our PDSG. Subsequently, we identify a distribution shift in the PDSG and continue to calibrate our PDSG through right-shifting.

Derivation of PDSG. The zeroth-order method is instrumental in estimating the gradient within a specific distribution [44, 53]. Therefore, we adopt the two-point zeroth-order method to the firing function as the foundation of our PDSG, which is expressed as:

$$G^2(u; z, \delta) = \frac{h(u + z\delta - V_{th}) - h(u - z\delta - V_{th})}{2\delta} z. \quad (8)$$

Here, z is sampled from a distribution λ, and δ is a constant smooth parameter. Then, the surrogate gradient can be calculated by the expectation of the two-point zeroth-order:

$$\frac{\partial s}{\partial u} = \mathbb{E}_{z\sim\lambda}[G^2(u; z, \delta)] = \int_{\frac{|u-V_{th}|}{\delta}}^{\infty} \frac{|z|}{\delta} \lambda(z) dz. \quad (9)$$

Consequently, we first investigate the distribution λ and the selection of δ. Since $\Delta u = z\delta$ can be viewed as a change to the origin membrane potential u, we treat $u' = u + z\delta$ as the perturbed potential. According to Eq. (1), the value of potential depends solely on the spikes from the previous layer when the weight and bias are fixed. The firing rate should remain stable after perturbed, as a severe change of the firing rate can lead to attacks being detected in terms of power consumption [31]. Therefore, u' and u can be approximately considered to be in the same distribution. Assuming that the membrane potential follows a normal distribution $\mathcal{N}(\mu, \sigma^2)$, then $z \sim \mathcal{N}(\frac{\mu-u}{\delta}, \frac{\sigma^2}{\delta^2})$. Given that the two-point zeroth-order estimation requires $\mathbb{E}(z) = 0$ and $\mathbb{E}(z^2) = 1$, we set $\delta = \sigma$, ensuring that $z \sim \mathcal{N}(0, 1)$ holds approximately when $u \approx \mu$. Substituting $\lambda(z)$ into Eq. (9), our PDSG is formulated as (See Sec. S1 for derivation):

$$\frac{\partial \boldsymbol{s}^l[t]}{\partial \boldsymbol{u}^l[t]} = \frac{1}{\sqrt{2\pi}\boldsymbol{\sigma}} \exp(-\frac{(\boldsymbol{u}^l[t] - V_{th})^2}{2\boldsymbol{\sigma}^2}). \quad (10)$$

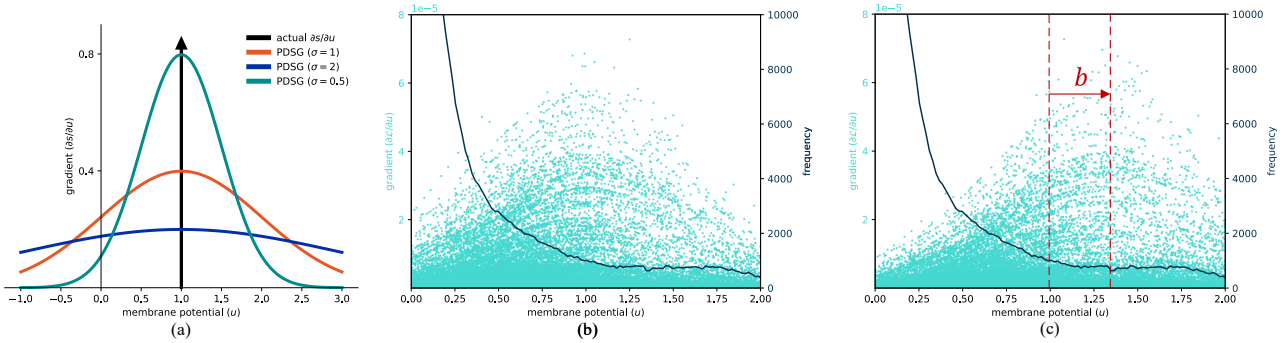


Figure 3. (a) Illustration of our PDSG under different distributions of membrane potential. (b) The scatter diagram of the gradients and the frequency curve of membrane potential in the penultimate layer of ResNet18. The gradients cluster on the left side of the threshold, causing imbalanced gradients. (c) The scatter diagram of the gradients after calibration. The distribution of the gradients is balanced around the threshold, urging the attack to pay equal attention to the gradients on the both sides of the threshold.

We depict the PDSG under diverse distributions of membrane potential in Fig. 3(a). A large deviation implies that the membrane potential is more dispersed, and the possible changes are larger, necessitating a flatter surrogate gradient to encompass a wider range of membrane potential, and vice versa. Consequently, our PDSG can adapt to various distributions of membrane potential while disregarding the training-phase SG of the models.

Calibration. Eq. (10) approximately holds when $u \approx \mu$. However, our interest lies in the surrogate function near the threshold where u significantly deviates from μ, as the firing rate of neurons is much lower than 50% [19, 67]. We depict the distribution of the gradients in the penultimate layer of ResNet18 in Fig. 3(b). The distribution of membrane potential is not symmetric about V_{th}, inducing dominant gradients clustering on the left side of the threshold. In this case, the attack will disproportionately focus the gradients at $u < V_{th}$, potentially causing an increasing firing rate and neglecting gradients on the other side of the threshold. Since an increasing firing rate can lead to saturated gradients during attacks [42], the gradients around the threshold are required to be balanced.

To address this issue, we calibrate our PDSG through right-shifting the SG by bias b in Eq. (11). As shown in Fig. 3(c), the distribution of gradients is balanced, ensuring that the gradients on both sides of the threshold receive equal attention. In this paper, we set $b = 0.5\boldsymbol{\sigma}$ in all experiments, which will be discussed in Sec. 5.4.

$$\frac{\partial \boldsymbol{s}^l[t]}{\partial \boldsymbol{u}^l[t]} = \frac{1}{\sqrt{2\pi}\boldsymbol{\sigma}} \exp\left(-\frac{(\boldsymbol{u}^l[t] - V_{th} - b)^2}{2\boldsymbol{\sigma}^2}\right) \tag{11}$$

4.2. Sparse Dynamic Attack

Gradient-based adversarial attacks are not applicable to binary dynamic images, since these attacks produce floating-point gradients. To effectively generate sparse perturbations, we propose a generation-reduction paradigm for our SDA. The generation process combines the gradients and the finite differences (FDs) to select the most desirable pixels to attack. The reduction process leverages recorded FDs to remove redundant perturbations.

4.2.1. Generation

①Contributing Gradients. Since the gradients are useful for rapid estimation, we first calculate the gradients of the input through STBP and our aforementioned PDSG as below:

$$\boldsymbol{g} = \frac{\partial \mathcal{L}(f(\boldsymbol{x}), y)}{\partial \boldsymbol{x}}. \tag{12}$$

To prevent gradient vanishing, we adopt the loss function from C&W [7]:

$$\mathcal{L}(f(\boldsymbol{x}), y) = \max\{f(\boldsymbol{x})_y - \max_{i \neq y} f(\boldsymbol{x})_i, 0\}. \tag{13}$$

When the loss decreases to zero, the attack will be successful. According to Eq. (12), the change of the input $\Delta\boldsymbol{x}$ contributes to the decline of the loss only when $\Delta\boldsymbol{x}$ and $\boldsymbol{g}$ have opposite signs. As the change of the input is unique for binary inputs ($\Delta x_i = 1$ for $x_i = 0$ and $\Delta x_i = -1$ for $x_i = 1$), we select the pixels which have contributing gradients and are not in the perturbation mask m:

$$\boldsymbol{g}^c = \boldsymbol{g} \cdot ((1 - 2\boldsymbol{x}) \cdot \boldsymbol{g} <= 0) \cdot (1 - \boldsymbol{m}). \tag{14}$$

②Top-k gradients. To optimize the sparsity of perturbations, we select only a portion of pixels for further calculations. As the pixels with larger gradient values are considered to contribute more adversary [9, 16], we select k pixels with the largest absolute gradient values, where k is incremented by k_{init} for each iteration. For all experiments, we set $k_{init} = 10$. In the n-th iteration, the top-k gradients are obtained as:

$$\boldsymbol{g}^k = \{g_i^c | i \in \text{argtop}k(|\boldsymbol{g}_c|), k = (n+1)k_{init}\}. \tag{15}$$

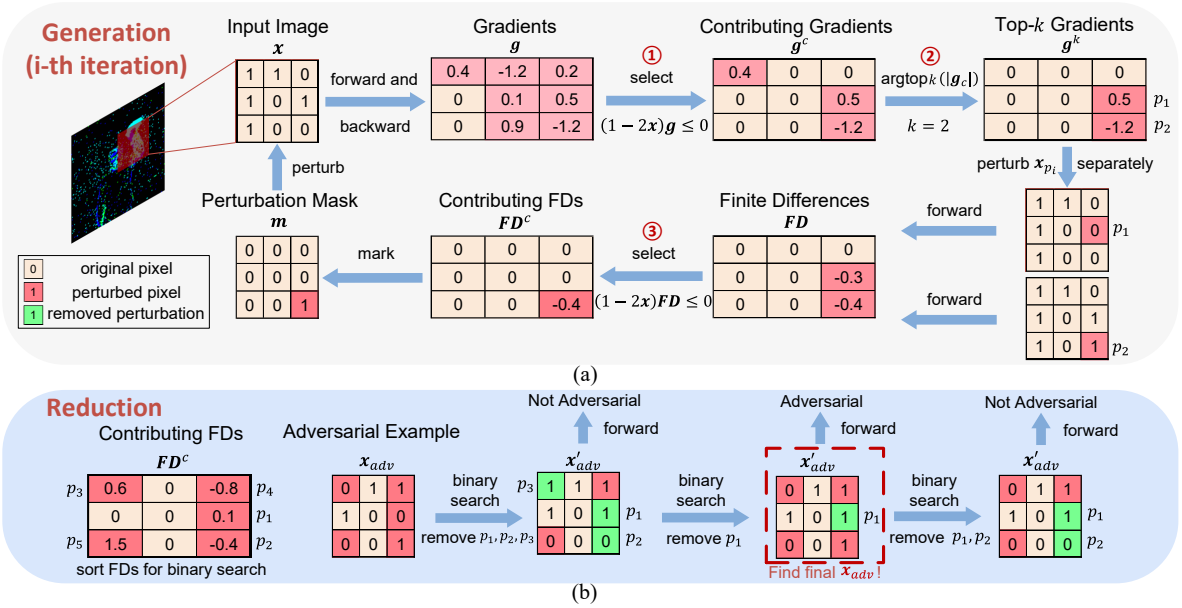


Figure 4. Illustration of our sparse dynamic attack. (a) In the generation process, we select contributing gradients through their signs, achieve top-k significant gradients and calculate their FDs to add perturbations. (b) In the reduction process, we sort the contributing FDs calculated by the generation process, then adopt binary search to find p_2 which makes the example cease to be adversarial after removed.

③Contributing FDs. The process of selecting top-k gradients is coarse. The gradients are inaccurate because the change of the input ($\Delta \boldsymbol{x} = \pm 1$) is large compared to the infinitesimal in Eq. (12). To perform a fine-grained selection, we leverage the finite difference, which reflects the actual trend of the loss when a binarized pixel is perturbed:

$$FD_i(\boldsymbol{x}) = \frac{\mathcal{L}(f(\boldsymbol{x} + \Delta x_i \boldsymbol{e}_i), y) - \mathcal{L}(f(\boldsymbol{x}), y)}{\Delta x_i} \boldsymbol{e}_i. \quad (16)$$

In Eq. (16), Δx_i denotes the change in the i-th index of the input, and $\boldsymbol{e}_i$ is the standard basis vector with 1 at i-th index. $FD_i(\boldsymbol{x})$ represents the change of the loss when the i-th index of the input is perturbed, which approximates to the gradient when $\Delta x_i \to 0$. We only calculate the FDs for the pixels selected by top-k gradients, and the forward process can be parallelized for acceleration. Similar to contributing gradients, we select contributing FDs for pixels whose change and FD have opposite signs:

$$\boldsymbol{FD}^c = \boldsymbol{FD} \cdot ((1 - 2\boldsymbol{x}) \cdot \boldsymbol{FD} <= 0). \quad (17)$$

Finally, we add the pixels with valid FD^c to the perturbation mask $\boldsymbol{m}$, perturb the input through the mask in the current iteration, and iteratively perform the generation process until the input becomes adversarial. The entire generation process is illustrated in Fig. 4(a).

4.2.2. Reduction

In the generation process, we have already obtained an adversarial example $\boldsymbol{x}_{adv}$ with sparse perturbations. Nonetheless, the selection of the top-k gradients leads to a local optimum. Therefore, we devise a perturbation reduction method to eliminate perturbations with minor impact.

Intuitively, we believe the pixel with the smallest absolute value of the FD has the least influence on the loss, and removing the pixel with the smallest FD is unlikely to affect the classification result. Consequently, we attempt to eliminate perturbations sequentially in ascending order of their absolute FDs. As shown in Fig. 4(b), we construct a sorted set of perturbed pixels, where n is the number of perturbed pixels:

$$\mathcal{S} = \{p_i \mid |FD^c_{p_1}| < \cdots < |FD^c_{p_n}|, \ i = 1, \cdots, n\}. \quad (18)$$

Then we sequentially remove the perturbations p_i from p_1 to p_n. If $\boldsymbol{x}_{adv}$ ceases to be adversarial upon removing p_j, we consider $\mathcal{S}_1 = \{p_1, \cdots, p_{j-1}\}$ as redundant perturbations which have no effect on the classification result, and $\mathcal{S}_2 = \{p_j, \cdots, p_n\}$ as necessary perturbations. Therefore, the objective of the reduction process is to find j in a sorted set. To improve the efficiency, we adopt the binary search to reduce the complexity to $O(\log\ n)$. An example with $j = 2$ and $n = 5$ is illustrated in Fig. 4(b). In general, this reduction process removes dispensable perturbations and fully optimizes the sparsity of perturbations.

5. Experiments

5.1. Experimental Setup

We validate the effectiveness of our PDSG on both static and dynamic datasets, and our SDA on dynamic datasets.

Dataset	Architecture	Input	Acc. (%)	Attack	ASR. (%) ($\epsilon = 2/255$)				ASR. (%) ($\epsilon = 8/255$)			
					STBP	RGA	HART	**PDSG (Ours)**	STBP	RGA	HART	**PDSG (Ours)**
CIFAR10	ResNet18	Direct	94.72	FGSM	38.21	31.14	37.30	**43.98**	52.36	45.80	46.72	**79.56**
				PGD	66.74	61.97	66.77	**69.62**	99.81	92.47	98.64	**100.00**
	ResNet18 (Adv. trained)	Direct	90.65	FGSM	4.10	6.66	8.66	**9.29**	22.31	34.64	47.18	**47.42**
				PGD	4.21	7.56	9.95	**10.68**	28.23	46.50	59.56	**62.16**
	ResNet18	Poisson	76.98	FGSM	5.40	5.85	6.64	**7.05**	28.92	25.58	31.57	**32.59**
				PGD	8.39	7.63	**8.50**	8.22	39.66	35.78	**42.30**	39.98
	VGG11	Direct	94.08	FGSM	26.93	21.39	27.29	**30.45**	45.37	37.02	38.89	**82.71**
				PGD	41.07	39.07	**48.12**	39.20	98.35	85.67	97.35	**99.71**
CIFAR100	ResNet18	Direct	75.94	FGSM	56.16	52.45	59.01	**59.05**	71.50	64.43	70.32	**83.29**
				PGD	81.92	78.36	**86.95**	78.50	99.60	98.00	99.62	**99.83**
ImageNet	HST-10-768	Direct	84.28	FGSM	47.52	0.47	**57.12**	54.89	56.29	4.97	76.61	**79.71**
				PGD	90.01	0.03	**91.41**	87.43	99.89	2.84	99.98	**100.00**
CIFAR10DVS	VGGSNN	Integer	78.80	FGSM	**9.90**	8.25	9.52	9.64	41.37	35.28	42.26	**42.39**
				PGD	9.39	7.74	10.15	**10.15**	**45.94**	37.31	45.81	44.54

Table 1. Comparison with state-of-the-art approaches on attacking static images and integer dynamic frames. ASR. denotes the attack success rate. ϵ is the attack intensity. STBP denotes attacking using training-phase SG. The best results are in bold.

Dataset	Architecture	Input	Acc. (%)	Attack	ASR. (%) ($\ell_0 < 200$)				ASR. (%) ($\ell_0 < 800$)				Dynamic Evaluation		
					STBP	RGA	HART	**PDSG (Ours)**	STBP	RGA	HART	**PDSG (Ours)**	ASR.	Mean ℓ_0	Median ℓ_0
N-MNIST	PLIFNet	Binary	99.57	SCG	0.0	0.0	0.0	0.0	18.0	15.0	0.0	34.0	91.0	1144.90	1162.00
				SpikeFool	15.0	0.0	2.0	1.0	93.0	20.0	46.0	27.0	97.0	444.44	356.00
				GSAttack	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	91.0	2828.25	2905.00
				SDA(Ours)	23.0	9.0	22.0	**63.0**	93.0	86.0	92.0	**99.0**	**100.0**	**207.34**	**171.00**
DVS-Gesture	VGGSNN	Binary	95.14	SCG	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	100.0	8377.84	7586.50
				SpikeFool	3.0	2.0	1.0	2.0	16.0	12.0	6.0	14.0	69.0	2762.41	1908.00
				GSAttack	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	71.0	9820.14	8521.50
				SDA(Ours)	19.0	10.0	16.0	**21.0**	38.0	39.0	40.0	**52.0**	**100.0**	**1731.63**	**769.50**
CIFAR10-DVS	ResNet18	Binary	78.20	SCG	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	100.0	2346.10	2191.00
				SpikeFool	19.0	2.0	13.0	0.0	70.0	13.0	33.0	8.0	100.0	674.89	491.00
				GSAttack	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	65.0	8511.60	8741.00
				SDA(Ours)	34.0	12.0	21.0	**38.0**	78.0	34.0	57.0	**82.0**	**100.0**	**458.02**	**303.00**

Table 2. Comparison with state-of-the-art approaches on SNN-based binary attack. $\ell_0 < 200$ means the number of modified pixels is less than 200. We incorporate the PDSG into our SDA in the dynamic evaluation. The best results are in bold.

CIFAR10/100 [32] and ImageNet [11] are adopted as static datasets, while N-MNIST [56], DVS-Gesture [2] and CIFAR10-DVS [36] are utilized as dynamic datasets. As the dynamic datasets are all in event-stream forms, we utilize SpikingJelly [20] framework to aggregate the events into 10 frames, and binarize the frames by capping the maximum value of each pixel to 1 [6, 27]. The input size for N-MNIST is 34×34, and for DVS-Gesture and CIFAR10-DVS it is 128×128, aligning with their original input sizes. The models contain spiking ResNet-18 [28], spiking VGG-11 [5], VGGSNN [12], and hierarchical spiking transformer (HST) [83]. The timestep for static datasets is 4 and for dynamic datasets is 10. Details of the datasets and models are provided in Sec. S4.

The evaluation metrics for the experiments include the attack success rate (ASR) and the ℓ_0-norm of perturbations. For attacking binary inputs, we randomly select 100 correctly classified inputs in the test set to perform attacks, and the attack fails when iterations exceed 500.

5.2. Comparison with State-of-the-art Works

In this section, we demonstrate the effectiveness of our PDSG and SDA by comparing them with state-of-the-art (SOTA) adversarial attacks on SNNs. We compare our PDSG with RGA [5] and HART [26], which focus on optimizing the gradient flow and adopt universal SGs, and compare our SDA with SCG [42], SpikeFool [6], and GSAttack [76], which perform attacks on binary dynamic images.

The results for attacking static images and integer dynamic frames in various attack intensities are shown in Tab. 1. On CIFAR10, we use ResNet18 with standard training and adversarial training. The adversarial training is conducted by PGD attack with $\epsilon = 2/255$ [61]. For ResNet18 in standard training, our PDSG significantly surpasses SOTA methods and STBP (attacking using the training-phase SG). The results also indicate that the training-phase SG is not always the most effective during attack. Although adversarial training effectively reduces the ASR, our PDSG is the least affected by the defense. For other models and

Training SG	Acc. (%)	Attack SG					
		Rect.(1)	Rect.(2)	Rect.(0.5)	ATan	Triangle	PDSG
Rect.(1)	94.72	52.36	64.13	27.20	68.56	41.29	**79.56**
Rect.(2)	93.25	48.33	73.03	20.99	59.41	37.06	**84.68**
Rect.(0.5)	94.58	44.50	50.73	29.25	65.35	38.41	**79.85**
ATan	94.67	47.10	60.98	25.72	66.91	37.79	**82.07**
Triangle	95.01	44.90	55.69	26.34	68.09	36.31	**79.77**
PDSG	90.69	60.57	65.99	26.76	64.73	51.44	**82.02**

Table 3. Attack success rate of FGSM ($\epsilon = 8/255$) on ResNet18 in cases of various training SGs and attack SGs. *Rect.*(1) means using the Rectangular SG with $w = 1$. The best results are in bold.

datasets, our PDSG performs the most stably and has the ability to maintain high ASR in various scenarios. Notably on ImageNet, the PDSG achieves 100% ASR.

For evaluating our SDA on attacking binary dynamic images, in Tab. 2, we measure the ASR under ℓ_0-norm bounded attack and unbounded attack. As SCG and GSAttack do not specifically optimize the sparsity of perturbations, their ℓ_0 are insufficient. Our SDA completely outperforms SpikeFool in terms of attack success rate and the sparsity. When combined with the PDSG, our SDA achieves 82% ASR under a constraint of $\ell_0 < 800$ on CIFAR10DVS, which is only 0.24% of the input pixels. In dynamic evaluation, the SDA perturbs a median of 303 pixels on CIFAR10DVS, which is only 62% of the SpikeFool.

5.3. Effects of the PDSG

Our PDSG focuses on addressing the issue of invisible SGs during attacks on SNNs. To demonstrate the adaptability of our PDSG, we first train ResNet18 on CIFAR10 using various shapes and parameters of SGs, with further details shown in the Appendix. Then we adopt FGSM to attack these models with various attack-phase SGs. The results are presented in Tab. 3. We observe that the ASR is highly dependent on the attack-phase SG. For fixed SGs, although the ATan SG performs well in most experiments, it is defeated by the Rect.(2) SG when the model is also trained using the Rect.(2) SG. Therefore, it is challenging to identify a fixed SG that consistently exhibits stable attack performance. Our PDSG demonstrates the best performance across all experiments, achieving approximate 80% ASR.

5.4. Ablation study

We first evaluate the effectiveness of the offset b in our PDSG. As illustrated in Fig. 5(a), the performance of the attack is sub-optimal before calibration ($b = 0$) due to the imbalanced distribution of membrane potential. After calibration, the ASR significantly improves as b increases within an appropriate range. However, the ASR will also decrease when b is excessively large. Therefore, to achieve stable performance, we adopt $b = 0.5\sigma$ in all experiments. Moreover, we discuss the impact of diverse timesteps of the model. As shown in Fig. 5(b), the accuracy increases

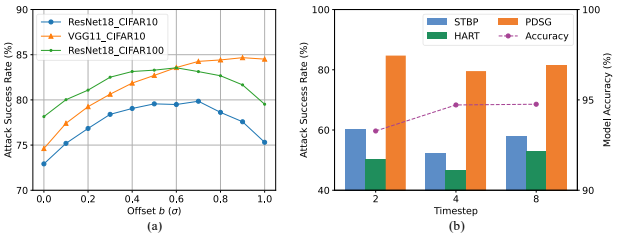


Figure 5. (a) Effectiveness of various offsets in the calibration of our PDSG. We adopt FGSM ($\epsilon = 8/255$) to perform attacks. (b) Impact of diverse timesteps in attacking ResNet18 on CIFAR10.

Top$k$ gradients	C&W Loss	FDs	Reduction	PDSG	ASR. (%)	Mean ℓ_0	Median ℓ_0
✓					69.0	655.70	450.0
✓	✓				72.0	678.90	450.0
✓	✓	✓			75.0	567.82	389.5
✓	✓	✓	✓		78.0	521.26	351.5
✓	✓	✓	✓	✓	**82.0**	**458.02**	**303.0**

Table 4. The effect of each component in our SDA method on CIFAR10DVS dataset. ✓denotes utilizing the component. The best results are in bold.

with the timestep; however, the ASR is independent from the timestep, indicating that the performance of the attack is related to the distribution of the actual model. Our PDSG performs the best at all timesteps, demonstrating that the PDSG can adapt to various timesteps of the model.

Tab. 4 shows that every optimization component plays an important role in our SDA. We set the top-k gradients in Eq. (15) as the base component because it provides a basic sparse selection of perturbations. When the C&W loss in Eq. (13) is adopted, although the mean ℓ_0 slightly increased, the ASR is improved due to resolved gradient vanishing. After introducing the FDs in Eq. (16) and the reduction process in Eq. (18), redundant perturbations are effectively removed, jointly improving the sparsity and the ASR. Cooperating with the PDSG, the representation of the gradients is fully optimized, thereby demonstrating superior attack performance of our SDA.

6. Conclusion

In this paper, we introduce the potential-dependent surrogate gradient to adaptively address invisible SGs in attacking SNNs. Moreover, a novel sparse dynamic attack method is proposed to effectively attack binary dynamic images on SNNs with sparse perturbations. The experiments demonstrate that our PDSG and SDA achieve superior performance with 100% attack success rate on ImageNet and 82% attack success rate by modifying only 0.24% of the pixels on CIFAR10DVS. Looking ahead, we hope that our proposed methods will inspire further research to comprehensively rethink the robustness of SNNs in various scenarios and to redesign corresponding defense mechanisms.

Acknowledgment

This paper is supported by the National Natural Science Foundation of China (NSF) under Grant No. 62406008.

References

[1] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(10):1537–1557, 2015. 1

[2] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7243–7252, 2017. 4, 7

[3] Arindam Basu, Lei Deng, Charlotte Frenkel, and Xueyong Zhang. Spiking neural network integrated circuits: A review of trends and future directions. In *2022 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–8. IEEE, 2022. 4

[4] Tong Bu, Wei Fang, Jianhao Ding, PENGLIN DAI, Zhaofei Yu, and Tiejun Huang. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International Conference on Learning Representations*, 2022. 2

[5] Tong Bu, Jianhao Ding, Zecheng Hao, and Zhaofei Yu. Rate gradient approximation attack threats deep spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7896–7906, 2023. 2, 7

[6] Julian Büchel, Gregor Lenz, Yalun Hu, Sadique Sheik, and Martino Sorbaro. Adversarial attacks on spiking convolutional neural networks for event-based vision. *Frontiers in Neuroscience*, 16:1068193, 2022. 2, 3, 7

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017. 5

[8] Sung-Gun Cho, Edith Beigné, and Zhengya Zhang. A 2048-neuron spiking neural network accelerator with neuro-inspired pruning and asynchronous network on chip in 40nm cmos. In *2019 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–4. IEEE, 2019. 4

[9] Francesco Croce, Maksym Andriushchenko, Naman D Singh, Nicolas Flammarion, and Matthias Hein. Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6437–6445, 2022. 5

[10] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018. 1

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 7

[12] Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. In *International Conference on Learning Representations*, 2022. 2, 7

[13] Jianhao Ding, Tong Bu, Zhaofei Yu, Tiejun Huang, and Jian Liu. Snn-rat: Robustness-enhanced spiking neural network through regularized adversarial training. *Advances in Neural Information Processing Systems*, 35:24780–24793, 2022. 3

[14] Jianhao Ding, Zhiyu Pan, Yujia Liu, Zhaofei Yu, and Tiejun Huang. Robust stable spiking neural networks. In *Forty-first International Conference on Machine Learning*, 2024. 3

[15] Jianhao Ding, Zhaofei Yu, Tiejun Huang, and Jian K Liu. Enhancing the robustness of spiking neural networks with stochastic gating mechanisms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 492–502, 2024. 3

[16] Xiaoyi Dong, Dongdong Chen, Jianmin Bao, Chuan Qin, Lu Yuan, Weiming Zhang, Nenghai Yu, and Dong Chen. Greedyfool: Distortion-aware sparse adversarial attack. *Advances in Neural Information Processing Systems*, 33: 11226–11236, 2020. 5

[17] Chaoteng Duan, Jianhao Ding, Shiyan Chen, Zhaofei Yu, and Tiejun Huang. Temporal effective batch normalization in spiking neural networks. *Advances in Neural Information Processing Systems*, 35:34377–34390, 2022. 2

[18] Rida El-Allami, Alberto Marchisio, Muhammad Shafique, and Ihsen Alouani. Securing deep spiking neural networks against adversarial attacks through inherent structural parameters. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 774–779. IEEE, 2021. 3

[19] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021. 2, 5

[20] Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, Liwei Huang, Huihui Zhou, Guoqi Li, and Yonghong Tian. Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40):eadi1480, 2023. 7

[21] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014. 3

[22] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 4

[23] Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. Recdis-snn: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 326–335, 2022. 3

[24] Yufei Guo, Xuhui Huang, and Zhe Ma. Direct learning-based deep spiking neural networks: a review. *Frontiers in Neuroscience*, 17:1209795, 2023. 2

[25] Sicong Han, Chenhao Lin, Chao Shen, Qian Wang, and Xiaohong Guan. Interpreting adversarial examples in deep learning: A review. *ACM Computing Surveys*, 55(14s):1–38, 2023. 1

[26] Zecheng Hao, Tong Bu, Xinyu Shi, Zihan Huang, Zhaofei Yu, and Tiejun Huang. Threaten spiking neural networks through combining rate and temporal information. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 7

[27] Weihua He, YuJie Wu, Lei Deng, Guoqi Li, Haoyu Wang, Yang Tian, Wei Ding, Wenhui Wang, and Yuan Xie. Comparing snns and rnns on neuromorphic vision datasets: Similarities and differences. *Neural Networks*, 132:108–120, 2020. 7

[28] Yulong Huang, Xiaopeng LIN, Hongwei Ren, Haotian FU, Yue Zhou, Zunchang LIU, biao pan, and Bojun Cheng. CLIF: Complementary leaky integrate-and-fire neuron for spiking neural networks. In *Forty-first International Conference on Machine Learning*, 2024. 7

[29] kang you, Zekai Xu, Chen Nie, Zhijie Deng, Qinghai Guo, Xiang Wang, and Zhezhi He. SpikeZIP-TF: Conversion is all you need for transformer-based SNN. In *Forty-first International Conference on Machine Learning*, 2024. 2

[30] Youngeun Kim, Hyoungseob Park, Abhishek Moitra, Abhiroop Bhattacharjee, Yeshwanth Venkatesha, and Priyadarshini Panda. Rate coding or direct coding: Which one is better for accurate, robust, and energy-efficient spiking neural networks? In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 71–75. IEEE, 2022. 4

[31] Sarada Krithivasan, Sanchari Sen, Nitin Rathi, Kaushik Roy, and Anand Raghunathan. Efficiency attacks on spiking neural networks. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pages 373–378, 2022. 4

[32] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report*, 2009. 7

[33] Yisong Kuang, Xiaoxin Cui, Yi Zhong, Kefei Liu, Chenglong Zou, Zhenhui Dai, Yuan Wang, Dunshan Yu, and Ru Huang. A 64k-neuron 64m-1b-synapse 2.64 pj/sop neuromorphic chip with all memory on chip for spike-based models in 65nm cmos. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(7):2655–2659, 2021. 4

[34] Souvik Kundu, Massoud Pedram, and Peter A Beerel. Hire-snn: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5209–5218, 2021. 3

[35] Guoqi Li, Lei Deng, Huajin Tang, Gang Pan, Yonghong Tian, Kaushik Roy, and Wolfgang Maass. Brain-inspired computing: A systematic survey and future trends. *Proceedings of the IEEE*, 2024. 1

[36] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:309, 2017. 7

[37] Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In *International conference on machine learning*, pages 6316–6325. PMLR, 2021. 2

[38] Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in Neural Information Processing Systems*, 34: 23426–23439, 2021. 2

[39] Yanjie Li, Xiaoxin Cui, Yihao Zhou, and Ying Li. A comparative study on the performance and security evaluation of spiking neural networks. *IEEE Access*, 10:117572–117581, 2022. 3

[40] Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic data augmentation for training spiking neural networks. In *European Conference on Computer Vision*, pages 631–649. Springer, 2022. 2

[41] Shuang Lian, Jiangrong Shen, Qianhui Liu, Ziming Wang, Rui Yan, and Huajin Tang. Learnable surrogate gradient for direct training spiking neural networks. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 3002–3010, 2023. 2, 3

[42] Ling Liang, Xing Hu, Lei Deng, Yujie Wu, Guoqi Li, Yufei Ding, Peng Li, and Yuan Xie. Exploring adversarial attack in spiking neural networks with spike-compatible gradient. *IEEE Transactions on Neural Networks and Learning Systems*, 34(5):2569–2583, 2021. 2, 3, 5, 7

[43] Ling Liang, Kaidi Xu, Xing Hu, Lei Deng, and Yuan Xie. Toward robust spiking neural network against adversarial perturbation. *Advances in Neural Information Processing Systems*, 35:10244–10256, 2022. 3

[44] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020. 4

[45] Yujia Liu, Tong Bu, Jianhao Ding, Zecheng Hao, Tiejun Huang, and Zhaofei Yu. Enhancing adversarial robustness in SNNs with sparse gradients. In *Forty-first International Conference on Machine Learning*, 2024. 3

[46] De Ma, Juncheng Shen, Zonghua Gu, Ming Zhang, Xiaolei Zhu, Xiaoqiang Xu, Qi Xu, Yangjing Shen, and Gang Pan. Darwin: A neuromorphic hardware co-processor based on spiking neural networks. *Journal of Systems Architecture*, 77:43–51, 2017. 1

[47] De Ma, Xiaofei Jin, Shichun Sun, Yitao Li, Xundong Wu, Youneng Hu, Fangchao Yang, Huajin Tang, Xiaolei Zhu, Peng Lin, et al. Darwin3: a large-scale neuromorphic chip with a novel isa and on-chip learning. *National Science Review*, 11(5):nwae102, 2024. 1

[48] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10 (9):1659–1671, 1997. 1

[49] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 4

[50] Alberto Marchisio, Giacomo Pira, Maurizio Martina, Guido Masera, and Muhammad Shafique. Dvs-attacks: Adversarial attacks on dynamic vision sensors for spiking neural networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2021. 2, 3

[51] Alberto Marchisio, Giacomo Pira, Maurizio Martina, Guido Masera, and Muhammad Shafique. R-snn: An analysis and design methodology for robustifying spiking neural networks against adversarial attacks through noise filters for dynamic vision sensors. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6315–6321. IEEE, 2021. 3

[52] Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: a few pixels make a big difference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9087–9096, 2019. 3

[53] Bhaskar Mukhoty, Velibor Bojkovic, William de Vazelhes, Xiaohan Zhao, Giulia De Masi, Huan Xiong, and Bin Gu. Direct training of snn using local zeroth order method. *Advances in Neural Information Processing Systems*, 36: 18994–19014, 2023. 4

[54] Bhaskar Mukhoty, Hilal AlQuabeh, Giulia De Masi, Huan Xiong, and Bin Gu. Certified adversarial robustness for rate encoded spiking neural networks. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[55] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36 (6):51–63, 2019. 2

[56] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015. 7

[57] Garrick Orchard, E Paxon Frady, Daniel Ben Dayan Rubin, Sophia Sanborn, Sumit Bam Shrestha, Friedrich T Sommer, and Mike Davies. Efficient neuromorphic signal processing with loihi 2. In *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 254–259. IEEE, 2021. 1, 4

[58] Ozan Ozdenizci and Robert Legenstein. Adversarially robust spiking neural networks through conversion. *Transactions on Machine Learning Research*, 2024. 3

[59] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019. 1, 4

[60] Nitin Rathi, Indranil Chakraborty, Adarsh Kosta, Abhronil Sengupta, Aayush Ankit, Priyadarshini Panda, and Kaushik Roy. Exploring neuromorphic computing based on spiking neural networks: Algorithms to hardware. *ACM Computing Surveys*, 55(12):1–49, 2023. 1

[61] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020. 7

[62] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019. 1

[63] Saima Sharmin, Priyadarshini Panda, Syed Shakib Sarwar, Chankyu Lee, Wachirawit Ponghiran, and Kaushik Roy. A comprehensive analysis on adversarial robustness of spiking neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 2

[64] Saima Sharmin, Nitin Rathi, Priyadarshini Panda, and Kaushik Roy. Inherent adversarial robustness of deep spiking neural networks: Effects of discrete input encoding and nonlinear activations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 399–414. Springer, 2020. 2

[65] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[66] Xiao Wang, Zongzhen Wu, Bo Jiang, Zhimin Bao, Lin Zhu, Guoqi Li, Yaowei Wang, and Yonghong Tian. Hardvs: Revisiting human activity recognition with dynamic vision sensors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(6):5615–5623, 2024. 2

[67] Ziming Wang, Runhao Jiang, Shuang Lian, Rui Yan, and Huajin Tang. Adaptive smoothing gradient learning for spiking neural networks. In *International Conference on Machine Learning*, pages 35798–35816. PMLR, 2023. 2, 5

[68] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018. 2, 3

[69] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence*, pages 1311–1318, 2019. 3

[70] Nuo Xu, Kaleel Mahmood, Haowen Fang, Ethan Rathbun, Caiwen Ding, and Wujie Wen. Securing the spike: On the transferability and security of spiking neural networks to adversarial examples. *arXiv e-prints*, pages arXiv–2209, 2022. 2

[71] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10221–10230, 2021. 4

[72] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 2023. 2

[73] Man Yao, JiaKui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo XU, and Guoqi Li. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. In

The Twelfth International Conference on Learning Representations, 2024. 2

[74] Man Yao, Ole Richter, Guangshe Zhao, Ning Qiao, Yannan Xing, Dingheng Wang, Tianxiang Hu, Wei Fang, Tugba Demirci, Michele De Marchi, et al. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature Communications*, 15(1):4464, 2024. 2

[75] Xingting Yao, Fanrong Li, Zitao Mo, and Jian Cheng. Glif: A unified gated leaky integrate-and-fire neuron for spiking neural networks. *Advances in Neural Information Processing Systems*, 35:32160–32171, 2022. 2

[76] Yanmeng Yao, Xiaohan Zhao, and Bin Gu. Exploring vulnerabilities in spiking neural networks: Direct adversarial attacks on raw event data. In *ECCV*, 2024. 2, 3, 7

[77] Friedemann Zenke and Tim P Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural computation*, 33 (4):899–925, 2021. 2

[78] Jilin Zhang, Dexuan Huo, Jian Zhang, Chunqi Qian, Qi Liu, Liyang Pan, Zhihua Wang, Ning Qiao, Kea-Tiong Tang, and Hong Chen. Anp-i: A 28-nm 1.5-pj/sop asynchronous spiking neural network processor enabling sub-0.1-uj/sample on-chip learning for edge-ai applications. *IEEE Journal of Solid-State Circuits*, 2024. 2, 4

[79] Yan Zhang, Cheng Chen, Dian Shen, Meng Wang, and Beilun Wang. Take care: Improving inherent robustness of spiking neural networks with channel-wise activation recalibration module. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 828–837. IEEE, 2023. 3

[80] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11062–11070, 2021. 2

[81] Yi Zhong, Yisong Kuang, Kefei Liu, Zilin Wang, Shuo Feng, Guang Chen, Youming Yang, Xiuping Cui, Qiankun Wang, Jian Cao, Song Jia, Yun Liang, Guangyu Sun, Xiaoxin Cui, Ru Huang, and Yuan Wang. Paicore: A 1.9-million-neuron 5.181-tsops/w digital neuromorphic processor with unified snn-ann and on-chip learning paradigm. *IEEE Journal of Solid-State Circuits*, pages 1–21, 2024. 2

[82] Chenlin Zhou, Han Zhang, Liutao Yu, Yumin Ye, Zhaokun Zhou, Liwei Huang, Zhengyu Ma, Xiaopeng Fan, Huihui Zhou, and Yonghong Tian. Direct training high-performance deep spiking neural networks: a review of theories and methods. *Frontiers in Neuroscience*, 18:1383844, 2024. 2

[83] Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, Li Yuan, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. QKFormer: Hierarchical spiking transformer using q-k attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 7

[84] Shuai Zhou, Chi Liu, Dayong Ye, Tianqing Zhu, Wanlei Zhou, and Philip S Yu. Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. *ACM Computing Surveys*, 55(8):1–39, 2022. 1