ROBUST SPIKING NEURAL NETWORKS AGAINST ADVERSARIAL ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking Neural Networks (SNNs) represent a promising paradigm for energy-efficient neuromorphic computing due to their bio-plausible and spike-driven characteristics. However, the robustness of SNNs in complex adversarial environments remains significantly constrained. In this study, we theoretically demonstrate that those threshold-neighboring spiking neurons are the key factors limiting the robustness of directly trained SNNs. We find that these neurons set the upper limits for the maximum potential strength of adversarial attacks and are prone to state-flipping under minor disturbances. To address this challenge, we propose a Threshold Guarding Optimization (TGO) method, which comprises two key aspects. First, we incorporate additional constraints into the loss function to move neurons' membrane potentials away from their thresholds. It increases SNNs' gradient sparsity, thereby reducing the theoretical upper bound of adversarial attacks. Second, we introduce noisy spiking neurons to transition the neuronal firing mechanism from deterministic to probabilistic, decreasing their state-flipping probability due to minor disturbances. Extensive experiments conducted in standard adversarial scenarios prove that our method significantly enhances the robustness of directly trained SNNs. These findings pave the way for advancing more reliable and secure neuromorphic computing in real-world applications.

1 INTRODUCTION

Spiking Neural Networks (SNNs) (Maass, 1997; Gerstner & Kistler, 2002; Izhikevich, 2003; Masquelier et al., 2008) mimics biological information transmission mechanisms using discrete spikes as the medium for information exchange, representing the cutting edge of neural computation (Cao et al., 2020; Varghese et al., 2016). Spiking neurons fire spikes only upon activation and remain silent otherwise. This event-driven mechanism (Liu & Yue, 2018) promotes sparse synapse operations and avoids multiply-accumulate (MAC) operations, significantly enhancing energy efficiency on neuromorphic platforms (Pei et al., 2019; DeBole et al., 2019; Ma et al., 2024; Pei et al., 2019). Recently, directly training SNNs with surrogate gradient methods (Wu et al., 2018; 2019; Deng et al., 2022; Li et al., 2021) has significantly reduced their performance gap with ANNs in classification tasks (Yao et al., 2024a; Shi et al., 2024; Zhou et al., 2024). However, these directly trained SNNs rely on Backpropagation Through Time (BPTT) (Werbos, 1990), thereby inheriting significant robustness issues associated with ANNs.

Directly trained SNNs (Fang et al., 2021b; Zhou et al., 2023; Bu et al., 2022; Duan et al., 2022) using surrogate gradient methods often exhibit a strong dependency on specific patterns or features (Ding et al., 2022; Mukhoty et al., 2024), rendering them particularly sensitive to minor disturbances. This characteristic reduces robustness in complex environments, especially against finely crafted adversarial disturbances (Laskov & Lippmann, 2010). To enhance the robustness of SNNs against adversarial attacks, researchers adapt strategies from ANNs, such as adversarial training (Ho et al., 2022; Ding et al., 2022) and certified training (Zhang et al., 2019; Liang et al., 2022). Furthermore, researchers develop optimization methods tailored to spike-driven mechanisms, , integrating with adversarial training to enhance robustness. Some researchers (Sharmin et al., 2020; Ding et al., 2023; El-Allami et al., 2021) utilize the temporal characteristics of SNNs to counteract environmental white noise attacks. Additionally, Evolutionary Leak Factor (Xu et al., 2024) and gradient sparsity regularization (SR) (Liu et al., 2024) significantly enhance the robustness of SNNs against gradient-

based attacks. However, a comprehensive and unified analysis of the robustness bottlenecks in directly trained SNNs remains lacking.

In this study, we theoretically demonstrate that threshold-neighboring spiking neurons are a key factor influencing the robustness of directly trained SNNs under adversarial attacks. We find that these neurons provide maximum potential pathways for adversarial attacks and are more prone to state-flipping under minor disturbances. To address this, we propose a Threshold Guarding Optimization (TGO) method. The TGO method aims to: (1) maximize the distance between neurons' membrane potentials and their thresholds to enhance gradient sparsity; (2) minimize the probability of state-flipping in neurons under minor disturbances. A series of experiments in standard adversarial scenarios demonstrates that our TGO method significantly enhances the robustness of directly trained SNNs. The contributions of this work are summarized as follows:

- We theoretically demonstrate that those threshold-neighboring spiking neurons are critical in limiting the robustness of directly trained SNNs under adversarial attacks. These neurons set the upper limits for the maximum potential strength of adversarial attacks and are prone to state-flipping under minor disturbances.
- We propose a Threshold Guarding Optimization (TGO) method, aiming to minimize threshold-neighboring neurons' sensitivity to adversarial attacks. First, we integrate additional constraints into the loss function, distancing the membrane potential from the threshold. Second, we introduce noisy spiking neurons to transit neuronal firing from deterministic to probabilistic, reducing the probability of state flips due to minor disturbances.
- We validate the effectiveness of the TGO method across various adversarial attack scenarios using different training strategies. Extensive experiments demonstrate TGO method achieves state-of-the-art (SOTA) performance in multiple adversarial attacks, significantly enhancing the robustness of directly trained SNNs. Notably, under RFGSM adversarial attacks, TGO combined with vanilla SNNs surpasses those adversarial training strategies for the first time.

2 Related Work

Spiking Neural Networks: SNNs offer a promising solution for resource-constrained edge computing (Zhang et al., 2023). To enhance the performance of SNNs, Wu et al. (2018) introduces the spatial-temporal backpropagation (STBP) algorithm, an adaptation of BPTT from Recurrent Neural Networks (RNNs) (Graves & Graves, 2012; Lipton, 2015). This method uses surrogate functions to approximate the non-differentiable Heaviside step function in spiking neurons. Additionally, researchers explore parallel training strategies (Fang et al., 2024) within the ResNet framework (He et al., 2016), shortcut residual connections (Zheng et al., 2021; Hu et al., 2021; Lee et al., 2020; Fang et al., 2021a), and Spike transformer (Li et al., 2022). Despite surrogate gradient methods (Deng et al., 2023; Yang & Chen, 2023) significantly improve training efficiency, SNNs remain susceptible to adversarial attacks as ANNs (Finlayson et al., 2019; Xu et al., 2020), limiting their applicability in adversarial environments.

Robustness of SNNs in Adversarial Attacks While biologically event-driven mechanisms (Marchisio et al., 2020; Hao et al., 2020) enhance SNNs' adaptability in complex environments, empirical studies (Liang et al., 2021; El-Allami et al., 2021) reveal that directly trained SNNs remain vulnerable to adversarial attacks. Initial efforts to mitigate this vulnerability start with adapting Adversarial Training (AT) (Goodfellow et al., 2014; Kundu et al., 2021) and subsequently advance to Regularized Adversarial Training (RAT) (Ding et al., 2022) with Lipschitz analysis. However, these approaches are constrained by additional training overhead and limited portability (Shafahi et al., 2019). Recently, researchers have developed optimization methods tailored to spike-driven mechanisms of SNNs. Such as Hao et al. (2023) enhances intrinsic robustness through rate-temporal information integration. (Xu et al., 2024) introduces FEEL-SNN with random membrane potential decay and innovative encoding mechanisms, and Ding et al. (2024b) develops gradient SR to strengthen defenses against Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) and Projected Gradient Descent (PGD) (Madry, 2017). Despite these advances, these strategies achieve significant enhancements only through synergistic integration with AT and RAT strategies. Moreover, a theoretical analysis of SNNs' inherent vulnerabilities in adversarial environments is still lacking. Thus, devising more effective robustness optimization strategies for SNNs remains a focused research.

3 PRELIMINARIES

3.1 SURROGATE GRADIENT FOR DIRECT TRAINED SNNS

SNNs effectively model the complex dynamics of biological neurons. Within the Leaky Integrate-and-Fire (LIF) framework, the membrane potential transitions through three key stages: integration, leakage, and firing. During integration, the membrane potential $V[t]$ accumulates over time in response to incoming spikes. When $V[t]$ exceeds a predefined threshold V_{th}, it triggers a spike that may influence downstream neurons. Following the spike, the membrane potential is reset to a specified baseline V_{reset}, preparing the neuron for subsequent inputs, which can be described as:

$$V[t] = \tau U[t-1] + WS[t], \tag{1}$$

$$S[t] = \Theta\left(V[t] - V_{\text{th}}\right), \tag{2}$$

$$U[t] = V[t]\left(1 - S[t]\right) + V_{\text{reset}} S[t], \tag{3}$$

where τ is the membrane time constant, W represents the synaptic weights, $S[t]$ denotes the spike at time t, and $\Theta(\cdot)$ is the Heaviside step function, indicating firing when $V[t]$ exceeds V_{th}. In the directly trained SNNs, the total loss L with respect to the weights W can be described as:

$$\frac{\partial L}{\partial W} = \sum_t \frac{\partial L}{\partial S[t]} \frac{\partial S[t]}{\partial V[t]} \frac{\partial V[t]}{\partial W}. \tag{4}$$

where $\frac{\partial S[t]}{\partial V[t]}$ represents the gradient of a non-differentiable step function involving the derivative of the Dirac δ-function, which is typically replaced by surrogate gradients with derivable curves. Various forms of surrogate gradients have been utilized, such as rectangular (Wu et al., 2018; 2019), triangular (Esser et al., 2016; Rathi & Roy, 2020), and exponential (Shrestha & Orchard, 2018) curves. Surrogate gradients provide a differentiable approximation to non-differentiable functions.

3.2 ADVERSARIAL ATTACKS

Adversarial attacks, including the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014)and Projected Gradient Descent (PGD) (Madry, 2017), rely on the model's gradient information to craft adversarial examples. FGSM generates such examples by applying a single-step perturbation designed to maximize the model's prediction error. The adversarial input is calculated as:

$$\mathbf{x}_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x, y_{\text{true}})), \tag{5}$$

where $\mathbf{x}_{\text{adv}}$ is the adversarial example, x is the original input, ϵ is the perturbation magnitude, $\mathcal{L}(x, y_{\text{true}})$ is the loss function, and $\text{sign}(\nabla_x \mathcal{L}(x, y_{\text{true}}))$ gives the sign of the gradient concerning the input. This process leverages the model's loss landscape to introduce minimal disturbances that significantly increase the classification error. Building on this, PGD iteratively refines adversarial examples by applying gradient updates and projecting them back into a bounded ϵ-ball centered on the original input. The attack rule of PGD can be expressed as:

$$\mathbf{x}_{\text{adv}}^{t+1} = \text{Clip}_{x,\epsilon}\left(\mathbf{x}_{\text{adv}}^{t} + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(\mathbf{x}_{\text{adv}}^{t}, y_{\text{true}}))\right), \tag{6}$$

where $\mathbf{x}_{\text{adv}}^{t}$ is the adversarial example at iteration t, α is the step size, and $\text{Clip}_{\mathbf{x},\epsilon}$ ensures that the perturbation remains within the prescribed ϵ-ball. PGD employs a multi-step approach to more precisely explore the disturbance space, producing adversarial examples closer to the optimal solution. At the same time, it strictly constrains the magnitude of disturbances, ensuring the perturbed input remains nearly indistinguishable from the original data to human observers. These two methods serve as standard benchmarks for evaluating the adversarial defense capabilities of neural networks.

4 METHODS

4.1 ROBUSTNESS ANALYSIS OF DIRECTLY TRAINED SNNS

To explore the key factors affecting the adversarial robustness of SNNs, we conduct a detailed analysis of the SNNs' dynamic properties under adversarial attacks. Our findings highlight two critical vulnerabilities associated with those threshold-neighboring spiking neurons. First, they establish a theoretical upper bound for the maximum potential strength of adversarial attacks. Second, they exhibit a higher probability of state-flipping under minor disturbances.

Upper Bound of Adversarial Attack Potency: Adversarial attacks maximize a model's expected loss by introducing carefully crafted perturbations to the original input. These perturbations are typically derived from the input's gradient information and applied as subtle yet strategically critical modifications along the gradient direction, thereby significantly degrading model performance while remaining imperceptible. The metric $\mathcal{R}_{\text{adv}}(f, x, \epsilon)$ quantifies the maximum potential strength of an adversarial attack on the neural network f at a input x, where the disturbances are constrained within a unit ℓ_p-norm ball and scaled by the factor ϵ. We prove the metric $\mathcal{R}_{\text{adv}}(f, x, \epsilon)$ of SNNs from two perspectives: surrogate gradient and spike pattern activation transitions. Details are described in Appendix.C and D.

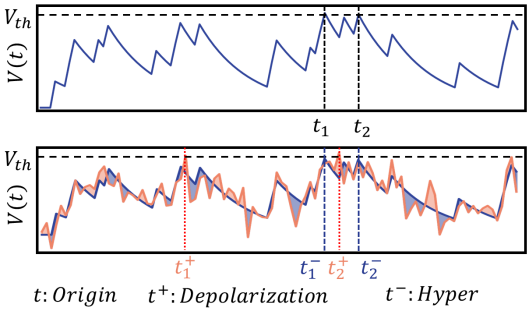

Figure 1: Red traces represent membrane potential dynamics of spiking neurons under adversarial attack. Only membrane potentials near thresholds undergo spike pattern transitions, while others remain unchanged.

Theorem 1 ***Gradient-based Adversarial Robustness Bound****: For a differentiable mapping $f : \mathbb{R}^d \to \mathbb{R}^m$, the robustness measure $\mathcal{R}_{adv}(f, x, \epsilon)$ under ℓ_2-bounded perturbations is upper-bounded as:*

$$\mathcal{R}adv(f, x, \epsilon) \leq \epsilon^2 |J_f(x)|2^2 + O(\epsilon^2),$$

where $J_f(x) \in \mathbb{R}^{m\times d}$ is the Jacobian matrix at x, with $\|J_f(x)\|_2^2 = \lambda_{\max}(G(x))$ for the gradient Gram matrix $G(x) = \sum_{i=1}^m \nabla f_i(x) \nabla f_i(x)^\top$. The maximum eigenvalue $\lambda_{\max}(G(x))$ captures the worst-case directional sensitivity, bounding adversarial vulnerability.

According to Theorem 1, the sensitivity of SNNs to adversarial attacks is correlated with the ℓ_2 norm of their Jacobian matrix, where higher gradient ℓ_2 norms indicate greater susceptibility to adversarial attacks. Notably, directly trained SNNs rely on surrogate gradients, which exhibit peak values near the V_{th}. As the number of threshold-neighboring spiking neurons increases, the ℓ_2 norm of the gradients in SNNs also rises, thereby enlarging $\mathcal{R}_{\text{adv}}(f, x, \epsilon)$. Beyond the surrogate gradient-based approximation analysis, we further derive adversarial robustness bounds for SNNs from the perspective of activation pattern transitions.

Theorem 2 ***Activation-based Adversarial Robustness Bound****: For a discrete spike pattern mapping $f : \mathbb{R}^n \to \mathbb{R}^m$, small perturbations $\varepsilon\delta$ around input x induce a finite set of activation pattern transitions. The adversarial robustness upper bound can be approximated as:*

$$R_{adv}(f, x, \varepsilon) \leq \varepsilon^2 \max_{1\leq k\leq K} \|A_{\mathcal{A}_k}\|_{p\to 2}^2,$$

where K denotes the number of activation regions intersecting the perturbation ball $B_\varepsilon(x)$, and $A_{\mathcal{A}_k} \in \mathbb{R}^{m\times n}$ is the affine transformation matrix for activation pattern $\mathcal{A}_k = \{(l, i) : u_i(x) \geq \theta_i\}$.

The transformation matrix $A_{\mathcal{A}_k}$ in Theorem.2 is structurally defined as: $A_{\mathcal{A}_k} = W^{(L)} \cdot \text{diag}(s_k^{(L-1)}) \cdot W^{(L-1)} \cdots \text{diag}(s_k^{(1)}) \cdot W^{(1)}$. $s_k^{(l)} = (s_{k,1}^{(l)}, s_{k,2}^{(l)}, \ldots)$ denotes the binary activation vector for activation pattern $\mathcal{A}_k$ at layer l, $\text{diag}(s_k^{(l)})$ operator constructs a diagonal matrix from the activation vector $s_k^{(l)} \in 0, 1^{n_l}$, and $s_{k,i}^{(l)} \in \{0, 1\}$ indicating the activation state of neuron i in layer l. As shown in Fig. 1, when all neurons' membrane potential are sufficiently distant from their thresholds, small perturbations $\varepsilon\delta$ fail to induce activation state transitions. Only a few membrane potentials near the threshold undergo state changes. In conclusion, these neurons significantly increase the theoretical upper limit of adversarial perturbation strength. How to effectively reduce the influence of these neurons will be crucial in enhancing the robustness of SNNs.

Strong State-flipping Probability: Adversarial attacks introduce carefully crafted small disturbances into the input data, achieving their disruptive effects. In multi-layer SNNs, these disturbances propagate through the networks, causing state-flipping in spiking neurons and ultimately altering the final output. Due to the spike-driven nature of SNNs, changes occur only when spiking neurons'

membrane potential crosses the threshold. Thus, the robustness of SNNs is directly linked to state-flipping in spiking neurons, which can be modeled as follows:

Theorem 3 *Let $V[t]$ be the membrane potential, V_{th} the threshold, and $\eta[t] \sim \mathcal{N}(0, \sigma^2)$ random perturbation. The probability P_{flip} of each neuron's flipping is given by:*

$$P_{\text{flip}} = \begin{cases} \Phi\left(\frac{V_{\text{th}} - V[t]}{\sigma}\right), & \text{if } V[t] \geq V_{\text{th}}, \\ 1 - \Phi\left(\frac{V_{\text{th}} - V[t]}{\sigma}\right), & \text{if } V[t] < V_{\text{th}}. \end{cases}$$

where Φ denotes the cumulative distribution function (CDF) of the standard normal distribution.

Theorem 3 defines the relationship between neuronal membrane potential and their state-flipping probability. Specifically, when $V[t] \geq V_{\text{th}}$, the neuron output switches from 1 to 0, and when $V[t] < V_{\text{th}}$, it flips from 0 to 1. Since the CDF of the standard normal distribution $\Phi(\cdot)$ is increasing monotonically, P_{flip} increases as the membrane potential $V[t]$ approaches the threshold potential V_{th}, whether $V[t]$ is above or below V_{th}. Those neurons are the primary targets of adversarial attacks.

In summary, threshold-neighboring spiking neurons play a crucial role in the adversarial robustness of SNNs. To address this, we propose an optimization strategy designed to mitigate their impact, thereby strengthening the overall resilience of SNNs in adversarial environments.

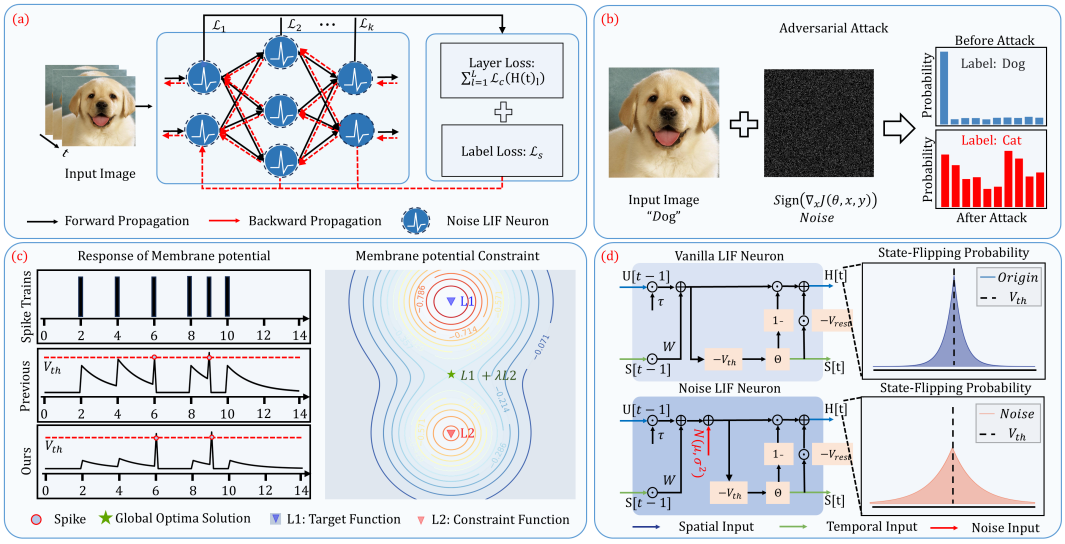

Figure 2: Mechanism and working principle of the TGO method. (a) The TGO method combines membrane potential constraints with noisy LIF neuron models for adversarial defense. (b) Gradient-based adversarial attacks illustrate how disturbances affect input images. (c) The joint optimization of the objective and constraint functions drives neuron membrane potentials away from the firing threshold. (d) The noisy LIF model effectively reduces the probability of state flips caused by small input disturbances, enhancing model stability.

4.2 Threshold Guarding Optimization Method

4.2.1 Membrane Potential Constraints

The surrogate gradients of threshold-neighboring spiking neuron, significantly influence the $\|J_f(x)\|_2^2$ of SNNs. To mitigate this effect, we propose additional constraints at each spiking neuron layer to optimize the membrane potential distribution, ensuring it remains as distant as possible from the threshold. The membrane potential constraint function can be described as:

$$\mathcal{C}(V(t)_l) = \frac{1}{TN} \sum_{i=1}^{n} \max(0, \delta - |V(t)_i - V_{\text{th}}|)^2, \tag{7}$$

$\mathcal{C}(V(t)_l)$ computes the average quadratic penalty when membrane potentials $V(t)_i$ of neurons in layer l approach the firing threshold V_{th}. N and T represent the number of time steps and the total number of layers in the SNNs, respectively. The hyperparameter δ establishes a margin around Vth where proximate potentials incur proportional penalties. Subsequently, we integrate this constraint with the target loss function, defining the overall loss within the framework of Lagrangian constraints (Kim & Jeong, 2021; Yoo & Jeong, 2023), which can be expressed as:

$$\mathcal{L}(\mathbf{x}, \lambda) = \mathcal{L}oss(\mathbf{x}) + \lambda \sum_{l} \mathcal{C}(V(t)_l). \tag{8}$$

Here, $\mathcal{L}oss(\mathbf{x})$ is the original loss function, $\mathcal{C}(V(t)_l)$ represents the penalty term for the membrane potentials across all layers, and λ is a dynamically adjusted parameter that controls the significance of the constraint. We reveal that using a fixed magnitude for λ hinders network convergence and constraint satisfaction. Specifically, a larger λ leads to significant performance degradation and poor convergence during the initial training phase, while a smaller λ fails to enforce the constraint effectively. Therefore, to achieve an optimal balance between gradients sparsity and performance, we propose dynamic λ, which can be described as:

$$\lambda = 0.5 \times \lambda_{\text{max}} \times \left(1 - \cos\left(\frac{\pi \times \text{epoch}}{\text{epoch}_{max}}\right)\right). \tag{9}$$

In the dynamic adjustment strategy, λ is initially set low to allow extensive exploration of parameter space and prevent premature constraints on $V[t]$ distributions. As λ increases, constraints intensify, pushing $V[t]$ further from V_{th} and ensuring strict adherence to the operational threshold. Membrane potential constraints effectively reduce the number of threshold-neighboring neurons, thereby decreasing $|J_f(x)|_2^2$ of SNNs. However, during training, some neurons inevitably remain near the threshold due to their significant impact on the loss function. Therefore, additional strategies are required to enhance the robustness of these critical neurons.

4.2.2 Noisy Spike Neurons for Mitigating State-flip Probability

As previously mentioned, these neurons exhibit high sensitivity to minor disturbances, readily undergoing state flipping by crossing the firing threshold. Such flipping not only increases output instability but can also severely disrupt the network's overall output through cascade effects. Consequently, we introduce the Noisy-LIF neuron model (Gerstner et al., 2014) as a complementary mechanism to membrane potential constraints, significantly reducing the probability of state flipping in critical neurons and thereby enhancing the robustness of directly trained SNNs against adversarial attacks. The dynamics of Noisy-LIF can be described as:

$$V[t] = \tau U[t-1] + WS[t] + \xi[t], \tag{10}$$

where $\xi[t]$ denotes Gaussian white noise, $V[t] = \tau U[t-1] + WS[t]$ is the membrane potential of the LIF model before the addition of noise. In the traditional LIF model, spike generation is deterministic: neurons emit a spike whenever the membrane potential $V[t]$ surpasses the firing threshold V_{th}. Details are described in Appidex.F. As a result, even small noise perturbations can cause significant output flips if they drive the membrane potential above the threshold. By introducing $\xi[t]$, the output transitions to an expected value rather than a binary outcome. For small changes $\Delta V[t]$ around V_{th}, the change in spike probability, interpreted as the flipping probability, can be approximated using a Taylor expansion with the first order:

$$\Delta P(S_l = 1 \mid V[t], \xi[t]) \approx \frac{1}{\sigma} \phi(z)\, \Delta V[t], \tag{11}$$

where $z = \frac{V_{\text{th}} - V[t] - \mu}{\sigma}$ and $\phi(\cdot)$ represents the PDF of the standard normal distribution. Then the derivative of the approximated flipping probability $\Delta P(S_l = 1 \mid V[t], \xi[t])$ with respect to σ, denoted as $\frac{\partial(\Delta P)}{\partial \sigma}$ can be expressed as:

$$\frac{\partial(\Delta P)}{\partial \sigma} = \Delta V[t] \frac{\phi(z)}{\sigma^2} \left(z^2 - 1\right). \tag{12}$$

For values of $V[t]$ close to V_{th}, an appropriate choice of σ can ensure that $z^2 < 1$. Under these conditions, $\frac{\partial(\Delta P)}{\partial \sigma}$ is negative, implying that the flipping probability decreases monotonically with

increasing σ. This observation indicates that increasing the noise level σ reduces the sensitivity of the flipping probability to small perturbations in the membrane potential $V[t]$, thereby enhancing the buffering effect of the Noisy-LIF neuron against such disturbances. Overall, the membrane potential constraints and noisy-LIF neurons in the TGO method work synergistically. The constraints ensure that most neuronal potentials are distanced from the threshold, while the noisy-LIF neurons further reduce the probabilities of state flipping in those threshold-neighboring spiking neurons.

5 EXPERIMENTS

5.1 EXPERIMENTS SETTING

In this study, the TGO method is evaluated in image classification tasks using the CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009) datasets. The employed network architectures include VGG-11 and WideResNet-16 with a width of 4 and depth of 16 (Xu et al., 2024; Ding et al., 2024a;b). We employ multiple adversarial attack methods, including FGSM (Goodfellow et al., 2014) and RFGSM (Wong et al., 2020) and PGD (Madry, 2017). They are all assessed using a fixed attack intensity of 8/255, with the number of iterations specified in the attack name (e.g., PGD 10). Notably, RFGSM refers to introducing a small random disturbance samples before applying FGSM. Moreover, three training strategies are implemented: first, a vanilla training strategy (BPTT) (Wu et al., 2018) using original images, which incurs no additional training costs; second, AT (Kundu et al., 2021) strategy involves training with white-box PGD attacks (attack intensity of 2/255 and step size $k = 2$); third, RAT incorporates Lipschitz penalties (Ding et al., 2022) into the adversarial training.

Table 1: Comparative results of model robustness across different methods and conditions in CIFAR-100. SOTA performances are marked in gray.

Model	Train	Method	Adversarial Attacks						
			Clean	FGSM	RFGSM	PGD7	PGD10	PGD20	PGD40
VGG-11	BPTT	Vanilla (Deng et al., 2022)	71.42	5.92	24.95	0.18	0.07	0.04	0.02
		DLIF (Ding et al., 2024a)	70.79	6.95	-	0.08	0.05	0.00	0.00
		StoG (Ding et al., 2024b)	70.44	8.27	-	0.49	-	-	-
		TGO(Ours)	69.47	17.20	38.23	2.30	1.33	0.69	0.42
	AT	AT (Kundu et al., 2021)	66.27	17.20	45.13	8.30	9.62	8.16	7.52
		StoG (Ding et al., 2024b)	66.37	23.45	-	14.42	-	-	-
		TGO(Ours)	65.93	24.16	47.90	12.83	10.12	8.72	6.59
	RAT	RAT (Ding et al., 2022)	67.76	20.87	46.21	11.14	9.34	7.66	6.90
		StoG (Ding et al., 2024b)	62.26	33.40	-	23.15	-	-	-
		TGO(Ours)	65.64	33.84	51.44	18.84	14.59	10.57	8.90
WRN-16	BPTT	Vanilla (Deng et al., 2022)	73.46	6.36	11.15	0.01	0.00	0.00	0.00
		DLIF (Ding et al., 2024a)	73.85	8.08	-	0.00	0.00	0.00	0.00
		SR (Liu et al., 2024)	67.67	11.15	18.18	-	-	-	-
		TGO(Ours)	69.04	23.90	41.17	3.26	1.84	0.67	0.35
	AT	AT (Kundu et al., 2021)	68.57	21.18	46.60	11.14	9.22	7.50	6.72
		DLIF (Ding et al., 2024a)	65.86	25.90	-	15.20	14.03	13.37	13.30
		SR (Liu et al., 2024)	60.37	25.76	36.93	-	19.76	-	-
		TGO(Ours)	64.49	41.99	55.01	27.78	22.75	15.99	12.91
	RAT	RAT (Ding et al., 2022)	67.59	25.07	48.92	13.60	11.41	9.07	8.35
		SR (Liu et al., 2024)	60.37	25.76	36.93	-	-	-	
		TGO(Ours)	64.22	40.35	54.53	25.79	20.93	14.02	10.77

5.2 COMPARE WITH THE SOTA METHODS

To evaluate the effectiveness of the proposed TGO method, we implement three training strategies (BPTT, AT, and RAT) and compare them with other SOTA robustness methods. Specifically, we replicate the AT and RAT configurations. Table.1 reports the classification accuracy of various network architectures under different attack scenarios on the CIFAR-100 datasets. Our results show

that the TGO method consistently achieves SOTA performance across nearly all tested architectures. Significantly, under the BPTT training strategy, our TGO method enhances performance by 10-20% in FGSM and RFGSM attack scenarios. It outperforms other robustness methods and even surpasses those adversarial training methods. Additionally, the TGO strategy is highly compatible with both AT and SAT approaches, achieving approximately 10% performance improvement under PGD10, PGD20, and PGD40 attacks in the WRN-16 model. Similar to other constraint-based approaches, our method incurs minimal performance loss on clean data while achieving a 3%-5% increase in performance under adversarial attacks.

Table 2: Comparative results under APGD and MTPGD Attacks with WideResNet-16 on CIFAR-100.

Model	Attack	Iteration Steps						
		10	20	30	40	50	80	100
AT (Kundu et al., 2021)	$APGD_{DLR}$	9.74	7.69	7.42	6.92	6.53	6.21	6.09
	$APGD_{CE}$	5.73	4.34	3.85	3.60	3.38	3.18	3.02
	MTPGD	11.61	9.59	9.27	8.97	8.57	8.49	8.49
SR(Liu et al., 2024)+AT	$APGD_{DLR}$	18.87	16.73	16.02	15.03	15.08	14.89	14.71
	$APGD_{CE}$	15.40	13.48	12.78	12.48	12.04	11.55	11.17
	MTPGD	23.17	21.79	21.54	21.39	21.45	21.28	20.81
TGO+AT (Ours)	$APGD_{DLR}$	**35.60**	**31.53**	**29.84**	**28.72**	**28.15**	**25.80**	**24.83**
	$APGD_{CE}$	**29.19**	**27.85**	**24.24**	**22.32**	**21.32**	**21.28**	**21.30**
	MTPGD	**43.49**	**36.88**	**34.04**	**31.45**	**30.70**	**30.10**	**29.76**

Furthermore, we conducted experiments across various perturbation magnitudes ϵ (2, 4, 6, 8/255). As shown in Fig. 3, TGO outperforms SR across all perturbation levels. Moreover, we evaluated the TGO

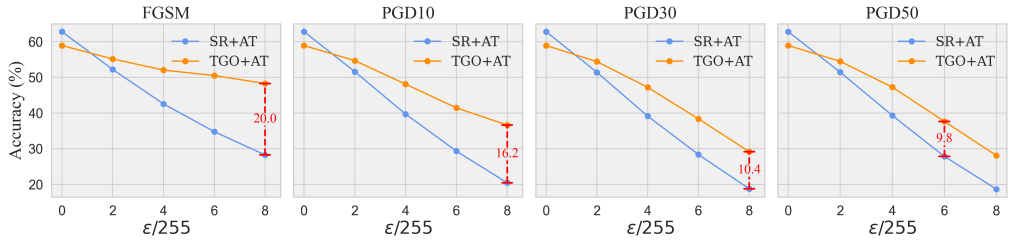

Figure 3: Performance comparison of TGO (ours) and SR with AT across different perturbation budgets ϵ. Experiments are conducted on CIFAR-100 dataset using WRN-16 architecture.

method against more advanced attack strategies: Auto-PGD (APGD) (Croce & Hein, 2020) and Multi-Targeted PGD (MTPGD) (Gowal et al., 2019). For APGD, we employed a dual-loss configuration using both Difference of Logits Ratio (DLR) and cross-entropy losses, where DLR directly targets the margin between correct and runner-up classes, providing a more challenging evaluation. MTPGD presents unique challenges due to its multi-targeted nature, simultaneously considering multiple misclassification objectives. We implemented MTPGD with random initialization to avoid gradient masking and gradient clipping for stable optimization. Table 2 presents our comparative analysis of TGO against standard Adversarial Training (AT) and SR+AT (Liu et al., 2024), maintaining consistent hyperparameters across all methods (momentum=0.9, ϵ=8/255) to ensure fair comparison. Finally, we also conduct extensive evaluations on neuromorphic datasets (Li et al., 2017). The details these part are presented in Appendix A. Results across all attack scenarios conclusively demonstrate TGO's significant robustness improvements against diverse adversarial threats.

5.3 Ablation Study

In this study, we evaluate the two core components of the TGO method: membrane potential constraint (MC) and the noisy-LIF (NLIF) model through a series of ablation experiments. The experiments

are performed using two training strategies: vanilla BPTT and RAT, with the CIFAR-100 dataset and VGG-11 architecture ensuring result reliability. The results show that our method significantly outperforms pure AT and SR method against APGD and MTPGD attacks. Although it may not yet match the latest ANN methods, it represents a meaningful effort to explore the robustness of spiking neurons. As shown in Table.3, our experimental results reveal several key findings. First, the

Table 3: Ablation study of our TGO method on CIFAR-100 with VGG-11.

		BPTT				RAT			
MC	**NLIF**	**Clean**	**FGSM**	**RFGSM**	**PGD40**	**Clean**	**FGSM**	**RFGSM**	**PGD40**
✗	✗	71.4	5.9	25.0	0.0	67.8	20.9	46.2	6.9
✓	✗	64.3 (-7.2)	17.1 (+11.2)	25.9 (+0.9)	0.5 (+0.5)	61.4 (-6.4)	26.2 (+5.3)	42.7 (-3.5)	6.2 (-0.7)
✗	✓	70.6 (-0.8)	8.1 (+2.1)	31.7 (+6.6)	0.1 (+0.1)	68.1 (+0.4)	25.2 (+4.3)	50.1 (+3.9)	9.1 (+2.2)
✓	✓	66.9 (-4.6)	21.5 (+15.5)	39.1 (+14.1)	0.5 (+0.5)	63.3 (-3.8)	33.8 (+13.0)	50.8 (+4.6)	9.3 (+2.4)

vanilla SNN exhibits significant vulnerability to FGSM attacks, achieving only 5.92% classification accuracy, highlighting the urgent need to improve its adversarial robustness. Second, each individual component of TGO significantly improves network performance in adversarial attacks, confirming their efficacy in strengthening the robustness of SNNs. Notably, MC and NLIF enhance each other synergistically rather than functioning independently, which further confirms that TGO is a holistic protection strategy specifically targeting neurons near the threshold.

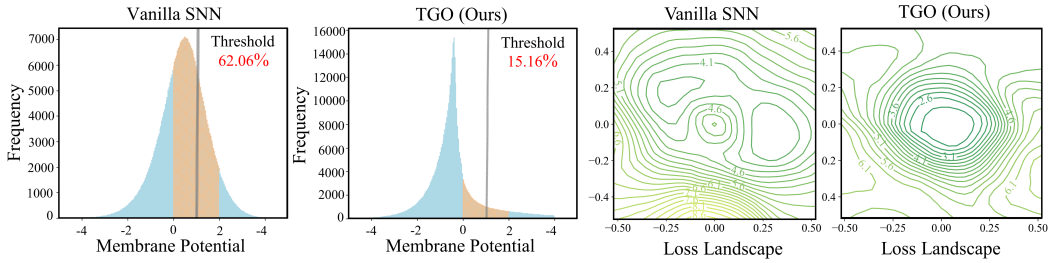

Figure 4: Comparison of membrane potential distributions and loss landscapes: The TGO-optimized SNN decreases membrane potentials near the threshold by approximately 40% and effectively circumvents adversarial traps during RFGSM attacks.

To better understand TGO's regulatory mechanism on the dynamics of spiking neurons, we analyze the membrane potential distributions of both vanilla and TGO-optimized SNNs. As shown in Fig.4, TGO reduces the number of threshold-neighboring neurons by approximately 40%, strongly supporting our hypothesis that threshold-neighboring neurons are key to adversarial robustness in SNNs.

Additionally, we compare the loss landscapes of vanilla and TGO-optimized SNNs under RFGSM attacks with the BPTT training strategy. The loss landscape of our TGO method exhibits smoother gradient trajectories, while the vanilla SNNs show multiple local optima and peaks, demonstrating the effectiveness of TGO in defending against gradient-based adversarial attacks. Moreover, we employ expectation over transformations (EoT) (Athalye et al., 2018) to validate the random introduced by Noisy-LIF, details are described in the Appendix. B. The results indicate that while it introduces some randomness, it still significantly enhances the adversarial robustness of the SNNs.

6 Conclusion

This study thoroughly analyzes the vulnerabilities of directly trained SNNs under adversarial attack conditions and theoretically confirms that threshold-neighboring spiking neurons define the upper limits of adversarial attack effectiveness. To address this issue, we propose a TGO method, which consists of two aspects. First, membrane potential constraints distance neurons from their thresholds, thereby reducing the upper limits of adversarial attacks. Second, noisy-LIF model transitions the neuronal firing mechanism from deterministic to probabilistic, effectively reducing the probability of state flips caused by minor disturbances. Extensive experiments prove that our TGO method significantly enhances the robustness of directly trained SNNs against various adversarial attacks.

REFERENCES

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pp. 284–293. PMLR, 2018.

Tong Bu, Jianhao Ding, Zhaofei Yu, and Tiejun Huang. Optimized potential initialization for low-latency spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 11–20, 2022.

Keyan Cao, Yefan Liu, Gongjie Meng, and Qimeng Sun. An overview on edge computing research. *IEEE access*, 8:85714–85728, 2020.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Michael V DeBole, Brian Taba, Arnon Amir, Filipp Akopyan, Alexander Andreopoulos, William P Risk, Jeff Kusnitz, Carlos Ortega Otero, Tapan K Nayak, Rathinakumar Appuswamy, et al. Truenorth: Accelerating from zero to 64 million neurons in 10 years. *Computer*, 52(5):20–29, 2019.

Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*, 2022.

Shikuang Deng, Hao Lin, Yuhang Li, and Shi Gu. Surrogate module learning: Reduce the gradient error accumulation in training spiking neural networks. In *International Conference on Machine Learning*, pp. 7645–7657. PMLR, 2023.

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.

Jianhao Ding, Zhaofei Yu, Tiejun Huang, and Jian K Liu. Spike timing reshapes robustness against attacks in spiking neural networks. *arXiv preprint arXiv:2306.05654*, 2023.

Jianhao Ding, Zhiyu Pan, Yujia Liu, Zhaofei Yu, and Tiejun Huang. Robust stable spiking neural networks. *arXiv preprint arXiv:2405.20694*, 2024a.

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*, volume 38, pp. 492–502, 2024b.

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.

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)*, pp. 774–779. IEEE, 2021.

Steven K Esser, Paul A Merolla, John V Arthur, Andrew S Cassidy, Rathinakumar Appuswamy, Alexander Andreopoulos, David J Berg, Jeffrey L McKinstry, Timothy Melano, Davis R Barch, et al. From the cover: Convolutional networks for fast, energy-efficient neuromorphic computing. *Proceedings of the National Academy of Sciences of the United States of America*, 113(41):11441, 2016.

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, 2021a.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2661–2671, 2021b.

Wei Fang, Zhaofei Yu, Zhaokun Zhou, Ding Chen, Yanqi Chen, Zhengyu Ma, Timothée Masquelier, and Yonghong Tian. Parallel spiking neurons with high efficiency and ability to learn long-term dependencies. *Advances in Neural Information Processing Systems*, 36, 2024.

Samuel G Finlayson, John D Bowers, Joichi Ito, Jonathan L Zittrain, Andrew L Beam, and Isaac S Kohane. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289, 2019.

Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

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.

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

Sven Gowal, Jonathan Uesato, Chongli Qin, Po-Sen Huang, Timothy Mann, and Pushmeet Kohli. An alternative surrogate loss for pgd-based adversarial testing. *arXiv preprint arXiv:1910.09338*, 2019.

Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

Yunzhe Hao, Xuhui Huang, Meng Dong, and Bo Xu. A biologically plausible supervised learning method for spiking neural networks using the symmetric stdp rule. *Neural Networks*, 121:387–395, 2020.

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*, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jiacang Ho, Byung-Gook Lee, and Dae-Ki Kang. Attack-less adversarial training for a robust adversarial defense. *Applied Intelligence*, 52(4):4364–4381, 2022.

Yangfan Hu, Huajin Tang, and Gang Pan. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):5200–5205, 2021.

Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14 (6):1569–1572, 2003.

Guhyun Kim and Doo Seok Jeong. Cbp: backpropagation with constraint on weight precision using a pseudo-lagrange multiplier method. *Advances in Neural Information Processing Systems*, 34: 28274–28285, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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*, pp. 5209–5218, 2021.

Pavel Laskov and Richard Lippmann. Machine learning in adversarial environments, 2010.

Chankyu Lee, Syed Shakib Sarwar, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy. Enabling spike-based backpropagation for training deep neural network architectures. *Frontiers in neuroscience*, 14:497482, 2020.

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.

Yudong Li, Yunlin Lei, and Xu Yang. Spikeformer: a novel architecture for training high-performance low-latency spiking neural network. *arXiv preprint arXiv:2211.10686*, 2022.

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.

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.

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.

Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. *arXiv Preprint, CoRR, abs/1506.00019*, 2015.

Daqi Liu and Shigang Yue. Event-driven continuous stdp learning with deep structure for visual pattern recognition. *IEEE transactions on cybernetics*, 49(4):1377–1390, 2018.

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.

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.

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

Aleksander Madry. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Alberto Marchisio, Giorgio Nanfa, Faiq Khalid, Muhammad Abdullah Hanif, Maurizio Martina, and Muhammad Shafique. Is spiking secure? a comparative study on the security vulnerabilities of spiking and deep neural networks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Timothée Masquelier, Rudy Guyonneau, and Simon J Thorpe. Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. *PloS one*, 3(1):e1377, 2008.

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.

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.

Nitin Rathi and Kaushik Roy. Diet-snn: Direct input encoding with leakage and threshold optimization in deep spiking neural networks. *arXiv preprint arXiv:2008.03658*, 2020.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in neural information processing systems*, 32, 2019.

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

Xinyu Shi, Zecheng Hao, and Zhaofei Yu. Spikingresformer: Bridging resnet and vision transformer in spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5610–5619, 2024.

Sumit B Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31, 2018.

Blesson Varghese, Nan Wang, Sakil Barbhuiya, Peter Kilpatrick, and Dimitrios S Nikolopoulos. Challenges and opportunities in edge computing. In *2016 IEEE international conference on smart cloud (SmartCloud)*, pp. 20–26. IEEE, 2016.

Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

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.

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*, volume 33, pp. 1311–1318, 2019.

Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International journal of automation and computing*, 17:151–178, 2020.

Mengting Xu, De Ma, Huajin Tang, Qian Zheng, and Gang Pan. Feel-snn: Robust spiking neural networks with frequency encoding and evolutionary leak factor. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Shuangming Yang and Badong Chen. Effective surrogate gradient learning with high-order information bottleneck for spike-based machine intelligence. *IEEE transactions on neural networks and learning systems*, 2023.

Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *Advances in neural information processing systems*, 36, 2024a.

Yanmeng Yao, Xiaohan Zhao, and Bin Gu. Exploring vulnerabilities in spiking neural networks: Direct adversarial attacks on raw event data. In *European Conference on Computer Vision*, pp. 412–428. Springer, 2024b.

Donghyung Yoo and Doo Seok Jeong. Cbp-qsnn: Spiking neural networks quantized using constrained backpropagation. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2023.

Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019.

Jilin Zhang, Dexuan Huo, Jian Zhang, Chunqi Qian, Qi Liu, Liyang Pan, Zhihua Wang, Ning Qiao, Kea-Tiong Tang, and Hong Chen. 22.6 anp-i: A 28nm 1.5 pj/sop asynchronous spiking neural network processor enabling sub-o. 1 μj/sample on-chip learning for edge-ai applications. In *2023 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 21–23. IEEE, 2023.

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*, volume 35, pp. 11062–11070, 2021.

Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Zhengyu Ma, Han Zhang, Huihui Zhou, and Yonghong Tian. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv preprint arXiv:2304.11954*, 2023.

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 qk attention. *arXiv preprint arXiv:2403.16552*, 2024.

A Robustness of Our TGO method for Neuromorphic Datasets

Table 4 illustrates the performance comparison between our proposed TGO+AT method and the standard Adversarial Training (AT) baseline with WideResNet-16 on the DVS-CRIAF10 neuromorphic dataset. In this study, we integrated TGO with conventional adversarial training, resulting in consistent improvements across all evaluation metrics. As evidenced in the table.4, TGO+AT not only enhances clean sample accuracy by 1.9% but also significantly improves robustness against various adversarial attacks, with particularly notable gains against stronger iterative attacks such as PGD7 (+4.1%) and PGD30 (+3.6%). Furthermore, we evaluated the efficacy of our approach

Table 4: Performance Comparison on DVS-CRIAF10 Dataset

Model	Clean	FGSM	RFGSM	PGD7	PGD10	PGD30	PGD50
AT	72.2	57.4	65.2	46.9	45.6	45.2	44.2
TGO+AT	**74.1**	**59.7**	**68.0**	**51.0**	**48.1**	**48.8**	**46.2**

against event-based attacks (Yao et al., 2024b) using WideResNet-16 on DVS-CRIAF10 (Li et al., 2017). The Attack Success Rate (ASR) for standard AT was measured at 44.32%, whereas AT+TGO achieved a substantially lower ASR of 31.0%. This reduction in ASR indicates enhanced robustness, further demonstrating the effectiveness of the TGO methodology in mitigating event-based adversarial attacks.

B Expectation Over Transformations Experiment for Randomness

we implemented the expectation over transformations (EoT) test to assess the impact of the random noise component on our TGO method's robust performance. In each iteration, we compute the expected loss to eliminate randomness from single inference steps and use this expectation to compute attack gradients. Finally, we compare the performance of TGO with AT (baseline) and the SR method with with WideResNet-16 on CIFAR-100, as shown below:

Table 5: Performance Comparison of TGO with AT and SR[2]+AT under Different PGD Attacks

Model	FGSM	PGD10	PGD20	PGD30	PGD40	PGD50	PGD80
AT	21.18	9.22	7.50	7.06	6.72	6.82	6.67
SR[2]+AT	28.27	20.46	19.28	18.80	18.78	18.69	18.51
TGO+AT(EoT)	42.86	26.59	24.57	22.68	22.43	22.28	21.96

Table 6: Performance Comparison of TGO with AT and SR+AT under the APGD Attack (CELoss)

Model (CELoss)	APGD20	APGD30	APGD40	APGD50	APGD80	APGD100
AT	4.34	3.85	3.60	3.38	3.18	3.02
SR[2]+AT	13.48	12.78	12.04	11.55	11.17	10.77
TGO+AT(EoT)	22.15	21.40	20.66	19.92	19.38	19.09

The experimental results show that our method indeed introduces a random component. However, even with this, under the EoT-PGD and EoT-APGD tests, our TGO method also can improve the robustness of SNNs.

C Gradient-based Upper Bound of Adversarial Attack

Theorem: Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a continuously differentiable neural network function at point x, and let $\epsilon > 0$ be sufficiently small. The adversarial perturbation measure $\mathcal{R}_{\text{adv}}(f, x, \epsilon)$ satisfies:

$$\mathcal{R}_{\text{adv}}(f, x, \epsilon) \leq \epsilon^2 \|J_f(x)\|_2^2 + O(\epsilon^3), \tag{13}$$

where $J_f(x)$ is the Jacobian matrix of function $f(\cdot)$ at point x.

Proof: We begin by defining our objective as the maximization of the squared difference between $f(x+\epsilon\delta)$ and $f(x)$, subject to the constraint $\|\delta\|_p \leq 1$:

$$\mathcal{R}_{\text{adv}}(f, x, \epsilon) = \underset{\delta}{\text{argmax}} \left\{ \|f(x+\epsilon\delta) - f(x)\|_2^2 : |\delta|_p \leq 1 \right\}. \tag{14}$$

Next, we apply Taylor's theorem with the Lagrange remainder for the vector-valued function $f(x+\epsilon\delta)$ around point x. Specifically, for δ with $\|\delta\|_p \leq 1$, there exists $\xi \in (0,1)$ such that:

$$f(x+\epsilon\delta) = f(x) + J_f(x)(\epsilon\delta) + \frac{(\epsilon\delta)^2}{2} H_f(x+\xi\epsilon\delta), \tag{15}$$

where $J_f(x)$ is the Jacobian matrix of the function $f(\cdot)$ at the point x, and H_f is the Hessian tensor of second derivatives. Substituting this expansion into the squared difference, we obtain:

$$\|f(x+\epsilon\delta) - f(x)\|_2^2 = \left\| J_f(x)(\epsilon\delta) + \frac{(\epsilon\delta)^2}{2} H_f(x+\xi\epsilon\delta) \right\|_2^2 \tag{16}$$

$$\leq \left(\|J_f(x)(\epsilon\delta)\|_2 + \frac{1}{2} \|(\epsilon\delta)^2 H_f(x+\xi\epsilon\delta)\|_2 \right)^2. \tag{17}$$

Combining Eq. 14 and the Cauchy-Schwarz inequality, we can systematically expand each component of the Eq. 17. For the first term, it can be expand as follows:

$$\|J_f(x)(\epsilon\delta)\|_2 \leq \|J_f(x)\|_2 \|\epsilon\delta\|_2 \leq \epsilon \|J_f(x)\|_2, \tag{18}$$

For the second term of Eq. 17, we can expand it by utilizing the ℓ_2 norm characteristic of the Hessian matrix. It can be defined as follows:

$$\|H_f(x+\xi\epsilon\delta)(\epsilon\delta)^2\|_2 \leq \|\lambda_{Hmax}(\epsilon\delta)^2\|_2 \leq \lambda_{Hmax}\epsilon^2, \tag{19}$$

where λ_{Hmax} is the maximum eigenvalue of the Hessian matrix. Substituting these bounds into the expression for the squared difference, we obtain:

$$\|f(x+\epsilon\delta) - f(x)\|_2^2 \leq \epsilon^2 \|J_f(x)\|_2^2 + \epsilon^3 \lambda_{Hmax} \|J_f(x)\|_2 + \frac{\epsilon^4 \lambda_{Hmax}^2}{4}, \tag{20}$$

Thus, the adversarial perturbation measure satisfies the following upper bound:

$$\mathcal{R}_{\text{adv}}(f, x, \epsilon) \leq \epsilon^2 \|J_f(x)\|_2^2 + O(\epsilon^2), \tag{21}$$

where $O(\epsilon^2)$ represents a higher-order infinitesimal of ϵ^2. Since ϵ is a very small quantity, the term $O(\epsilon^2)$ in the formula can be neglected. Consequently, the theoretical upper bound of the adversarial perturbation is primarily dependent on the ℓ_2 norm of $J_f(x)$. Specially, $J_f(x)$ can be expressed as the collection of gradients of each component of the function:

$$J_f(x) = \begin{bmatrix} \nabla f_1(x)^T \\ \nabla f_2(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{bmatrix}. \tag{22}$$

The ℓ_2 norm of the Jacobian matrix is related to the gradients through the following expression:

$$\|J_f(x)\|_2^2 = \lambda_{Jmax}(J_f(x)^T J_f(x)) = \lambda_{Jmax} \left(\sum_{i=1}^{m} \nabla f_i(x) \nabla f_i(x)^T \right). \tag{23}$$

Where λ_{Jmax} denotes the maximum eigenvalue of the matrix, which reflects the largest possible stretching effect of the Jacobian matrix, directly linking the gradient magnitudes to the overall sensitivity of the network's output.

D ACTIVATION-BASED UPPER BOUND OF ADVERSARIAL ATTACK

Theorem: For a discrete spike pattern mapping $f : \mathbb{R}^n \to \mathbb{R}^m$, small perturbations $\varepsilon\delta$ around input x induce a finite set of activation pattern transitions. The adversarial robustness upper bound can be approximated as:

$$R_{\text{adv}}(f, x, \varepsilon) \leq \varepsilon^2 \max_{1 \leq k \leq K} \|A_{\mathcal{A}_k}\|_{p\to 2}^2,$$

where K denotes the number of activation regions intersecting the perturbation ball $B_\varepsilon(x)$, and $A_{\mathcal{A}_k} \in \mathbb{R}^{m\times n}$ is the affine transformation matrix for activation pattern $\mathcal{A}_k = \{(l, i) : u_i(x) \geq \theta_i\}$.

Proof: Consider an SNN represented by the function $f : \mathbb{R}^n \to \mathbb{R}^m$, where $\mathbb{R}^n$ denotes the input space and $\mathbb{R}^m$ denotes the output space. The adversarial perturbation measure at an input point $x \in \mathbb{R}^n$ with a perturbation radius $\varepsilon > 0$ is defined as:

$$R_{\text{adv}}(f, x, \varepsilon) = \max_{\|\delta\|_p \leq 1} \|f(x + \varepsilon\delta) - f(x)\|_2^2, \tag{24}$$

where δ represents the perturbation direction vector constrained by the p-norm unit ball. Given the piecewise-linear nature of the SNN, small perturbations $\varepsilon\delta$ around an input x lead to a finite set of possible activation pattern changes. Specifically, the activation pattern at any input x can be defined as:

$$A(x) = \{(l, i) : u_i^{(l)}(x) \geq \theta_i^{(l)}\}, \tag{25}$$

where $u_i^{(l)}(x)$ denotes the membrane potential of neuron i at layer l, and $\theta_i^{(l)}$ denotes the corresponding firing threshold. Consequently, each distinct activation pattern $\mathcal{A}$ corresponds uniquely to a convex polyhedral region in the input space defined as:

$$R_{\mathcal{A}} = \{x \in \mathbb{R}^n : A(x) = \mathcal{A}\}. \tag{26}$$

Within any such region $R_{\mathcal{A}}$, the network behaves as an affine transformation described by:

$$f(x) = A_{\mathcal{A}}x + b_{\mathcal{A}}, \tag{27}$$

where $A_{\mathcal{A}} \in \mathbb{R}^{m\times n}$ and $b_{\mathcal{A}} \in \mathbb{R}^m$ are determined entirely by the network's weights and biases under the specific activation pattern $\mathcal{A}$. For sufficiently small perturbations δ ensuring $x + \varepsilon\delta \in R_{\mathcal{A}}$, the network's output change can explicitly be expressed as:

$$f(x + \varepsilon\delta) - f(x) = \varepsilon A_{\mathcal{A}}\delta. \tag{28}$$

Utilizing the properties of operator norms, we bound the change in network output within the given region by:

$$G(x) = \|f(x + \varepsilon\delta) - f(x)\|_2^2, \quad G(x) \leq \varepsilon^2 \|A_{\mathcal{A}}\|_{p\to 2}^2, \tag{29}$$

where $\|A_{\mathcal{A}}\|_{p\to 2}$ denotes the operator norm of the matrix $A_{\mathcal{A}}$ defined with respect to the input p-norm and output 2-norm. Considering all possible regions intersecting the perturbation ball $B_\varepsilon(x) = \{x + \varepsilon\delta : \|\delta\|_p \leq 1\}$, we derive the global bound for the adversarial perturbation measure as:

$$R_{\text{adv}}(f, x, \varepsilon) \leq \varepsilon^2 \max_{1 \leq k \leq K} \|A_{\mathcal{A}_k}\|_{p\to 2}^2, \tag{30}$$

where K represents the finite number of distinct activation regions intersecting $B_\varepsilon(x)$. Further analyzing the structure of the matrices $A_{\mathcal{A}_k}$, one observes that the sensitivity of these matrices primarily depends on neurons whose membrane potentials $u_i^{(l)}(x)$ are near their firing thresholds $\theta_i^{(l)}$. Consequently, larger distances between membrane potentials and thresholds indicate greater activation stability, leading to smaller variations in the affine transformation $A_{\mathcal{A}_k}$ and ultimately reducing the operator norm $\|A_{\mathcal{A}_k}\|_{p\to 2}$. Formally stated, as the absolute difference between the membrane potential $u_i^{(l)}(x)$ and the threshold $\theta_i^{(l)}$ increases, the adversarial perturbation measure strictly decreases:

$$|u_i^{(l)}(x) - \theta_i^{(l)}| \uparrow \quad \Longrightarrow \quad R_{\text{adv}}(f, x, \varepsilon) \downarrow . \tag{31}$$

E PROOF OF THE PROBABILITY FOR SPIKING NEURONS' STATE FLIPPING

Theorem: Consider a spiking neuron with membrane potential $V[t]$ at time t, firing threshold V_{th}, and subject to Gaussian white noise perturbation $\eta[t] \sim \mathcal{N}(0, \sigma^2)$. The probability P_{flip} of the neuron's state transition (flipping) is given by:

$$P_{\text{flip}} = \begin{cases} \Phi\left(\frac{V_{\text{th}} - V[t]}{\sigma}\right), & \text{if } V[t] \geq V_{\text{th}}, \\ 1 - \Phi\left(\frac{V_{\text{th}} - V[t]}{\sigma}\right), & \text{if } V[t] < V_{\text{th}}, \end{cases}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function (CDF) of the standard normal distribution.

Proof: Let us consider the stochastic dynamics of the neuron's membrane potential under noise perturbation. Define the noise-perturbed membrane potential as $H' = V[t] + \eta[t]$, where $\eta[t] \sim \mathcal{N}(0, \sigma^2)$ represents additive Gaussian white noise. The neuron's state transition probability depends on whether this perturbed potential crosses the threshold V_{th}. We analyze this probability by considering two distinct cases.

For Case 1, when $V[t] \geq V_{\text{th}}$, the deterministic dynamics would result in the neuron firing (output state = 1). However, the presence of noise can induce a transition to the non-firing state (0). This flip occurs if and only if the perturbed potential falls below threshold:

$$H' < V_{\text{th}} \quad \Longleftrightarrow \quad V[t] + \eta[t] < V_{\text{th}} \quad \Longleftrightarrow \quad \eta[t] < V_{\text{th}} - V[t]. \tag{32}$$

Since $\eta[t]$ follows a normal distribution with mean 0 and variance σ^2, we can standardize this inequality. The probability of transition from state 1 to 0 is:

$$P_{1\to 0} \;=\; P(\eta[t] < V_{\text{th}} - V[t]) \;=\; \Phi\left(\frac{V_{\text{th}} - V[t]}{\sigma}\right), \tag{33}$$

where the last equality follows from the definition of the standard normal CDF.

For Case 2, when $V[t] < V_{\text{th}}$, the deterministic dynamics would result in no firing (output state = 0). A transition to the firing state (1) occurs when noise pushes the membrane potential above threshold:

$$H' \geq V_{\text{th}} \quad \Longleftrightarrow \quad V[t] + \eta[t] \geq V_{\text{th}} \quad \Longleftrightarrow \quad \eta[t] \geq V_{\text{th}} - V[t]. \tag{34}$$

Following the same probabilistic reasoning, and noting that $P(\eta[t] \geq x) = 1 - P(\eta[t] < x)$ for any x, the probability of transition from state 0 to 1 is:

$$P_{0\to 1} \;=\; P(\eta[t] \geq V_{\text{th}} - V[t]) \;=\; 1 - \Phi\left(\frac{V_{\text{th}} - V[t]}{\sigma}\right). \tag{35}$$

Combining these cases yields the desired expression for P_{flip}. Note that this result naturally captures the intuition that the probability of state transition decreases as the membrane potential moves further from the threshold in either direction, due to the monotonicity properties of Φ.

F PROOF OF THE PROBABILITY FOR NOISY-LIF'S STATE FLIPPING

Theorm: Consider a Noisy-LIF neuron with membrane potential $V[t]$, Gaussian noise $\xi[t] \sim \mathcal{N}(0, \sigma^2)$, and threshold V_{th}. The probability of firing can be expressed as:

$$P(S_l = 1 \mid V[t], \xi[t]) = 1 - \Phi\left(\frac{V_{\text{th}} - (V[t] + \xi[t])}{\sigma}\right),$$

where $\Phi(\cdot)$ is the CDF of the standard normal distribution. As the σ increases, the probability density function ϕ becomes broader, reducing the sensitivity to small variations in $V[t]$.

Proof: The firing condition in the Noisy-LIF neuron model is given by the inequality $V[t]+\xi[t] \geq V_{\text{th}}$. To analyze the firing probability conditioned on the membrane potential $V[t]$ and the noise $\xi[t]$, we start by rewriting the firing condition in terms of the standard normal distribution:

$$P(S_l = 1 \mid V[t], \xi[t]) = P(\xi[t] \geq V_{\text{th}} - V[t]) = 1 - \Phi\left(\frac{V_{\text{th}} - V[t]}{\sigma}\right). \tag{36}$$

Differentiating this probability with respect to $V[t]$ gives us the sensitivity of the firing probability to changes in the membrane potential:

$$\frac{\partial P(S_l = 1 \mid V[t], \xi[t])}{\partial V[t]} = \frac{1}{\sigma}\phi\left(\frac{V_{\text{th}} - V[t]}{\sigma}\right), \tag{37}$$

where ϕ is the probability density function of the standard normal distribution. This derivative indicates how a small change in $V[t]$ affects the probability of firing. To understand the impact of σ on the sensitivity, consider that as σ increases, the width of ϕ also increases, making it flatter. This change results in a decrease in the magnitude of the derivative, indicating reduced sensitivity to small changes in $V[t]$.

Table 7: Performance of the proposed TGO method on VGG11 under different λ_{Max}. The attack perturbation $\epsilon = 2/255$ for all attacks, iterative step $k = 4$, step size $\alpha = 0.01$.

Datasets	networks	λ_{Max}	clean	FGSM	RFGSM	PGD7	PGD10	PGD20	PGD40
CIFAR100	BPTT	0.2	68.47	17.20	38.23	2.30	1.33	0.69	0.42
		0.4	67.68	19.25	39.12	2.57	1.47	0.62	0.39
		0.6	67.79	19.76	38.04	2.55	1.48	0.72	0.39
		0.8	66.86	21.46	39.06	3.09	1.73	0.76	0.53
	AT	0.2	65.40	23.06	46.81	12.56	10.11	7.78	6.69
		0.4	64.93	24.14	47.90	12.83	10.12	7.72	6.59
		0.6	64.42	25.05	47.71	13.05	10.23	7.37	6.75
		0.8	63.44	26.98	46.47	14.42	10.79	7.47	6.49
	RAT	0.2	66.78	28.68	51.77	17.08	13.27	9.96	9.06
		0.4	66.41	30.54	51.61	17.59	13.86	9.61	8.28
		0.6	65.64	31.90	51.44	18.84	14.59	10.57	8.90
		0.8	63.32	33.84	50.79	20.60	16.12	11.16	9.32

G More Experiment Details

In this research, the training process for all experiments extends over a duration of 300 epochs. To address potential vanishing or exploding gradients, batch normalization techniques are integrated throughout the network architecture. The LIF neurons in the experiments are configured with a decay factor of 0.5 and a threshold of 1. The proposed noisy LIF neurons incorporate Gaussian noise with a mean of 0 and a variance of 0.4. All experiments were conducted with 300 training iterations, repeated thrice, and the reported results are the averages of these three runs. For optimization, we implement the stochastic gradient descent (SGD) algorithm, starting with a learning rate set at 0.1. The adjustment of the learning rate follows a cosine annealing strategy. All experiments are performed on a PyTorch framework, facilitated by the computational power of an NVIDIA RTX 4090 GPU.

G.1 More Experimental Results

We integrate the proposed TGO method into the BPTT, AT, and RAT training of SNNs. We observe that the TGO method effectively and consistently enhances the robustness of vanilla models against various attacks across different networks, such as VGG11 and WideResNet16, on CIFAR-10 and CIFAR 100. This further validates the effectiveness of our approach.

G.2 Effect of λ_{Max} on robustness.

To further examine the impact of different values of λ_{Max} on the performance of SNNs, we conduct a series of sensitivity experiments. As shown in the right part of Fig.5, the results indicate that as λ_{Max} increases, there is a substantial enhancement in adversarial robustness.

However, this improvement is accompanied by a notable degradation in performance in clean environments when λ_{Max} exceeds 0.5, highlighting a clear trade-off between adversarial robustness and clean performance. To further refine the evaluation of the trade-off between performance and

robustness for different settings of λ_{Max}, especially considering multiple attack scenarios, the utility function $U(\lambda)$ can be expanded into a summation form. This modified utility function more comprehensively accounts for performance under various types of perturbations. Here is the updated formulation:

$$U(\lambda) = \alpha \cdot P(\lambda) + (1 - \alpha) \cdot \sum_{i=1}^{n} R_i(\lambda). \tag{38}$$

Where $P(\lambda)$ denotes the performance on clean data, and $R_i(\lambda)$ represents robustness under the i-th adversarial attack. n is the total number of attack types, and α is a weighting coefficient that balances the importance of clean data performance relative to adversarial robustness. Adjusting α and considering all types of attack n, this formulation allows for a nuanced evaluation of λ_{Max}, ensuring optimal performance and robustness in various real-world perturbations. As shown in Table 7, we analyzed the impact of different values λ_{Max} on the performance of the TGO algorithm. Our results indicate that $\lambda_{Max} = 0.4$ achieves the best balance between performance and robustness. Although increasing λ_{Max} to 0.6 and 0.8 improves adversarial robustness, it significantly reduces performance in clean data scenarios. This suggests that higher λ_{Max} values while improving stability under perturbations compromise efficiency under normal conditions. Based on these findings, we selected $\lambda_{Max} = 0.4$ for all subsequent experiments, as it offers the optimal trade-off between adversarial resilience and performance in clean environments.

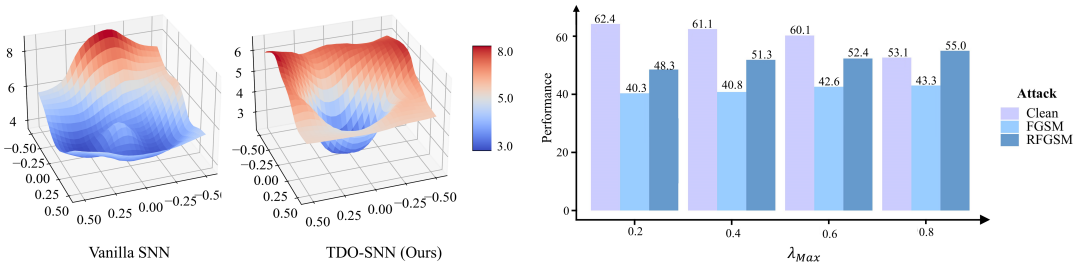

Figure 5: Loss landscape visualization and ablation experiments on $\lambda_{\max}$. The loss landscape of a vanilla SNN and our TGO-optimized SNN under the RFGSM attack, trained using the BPTT method. Notably, the vanilla SNN exhibits significant instability under the RFGSM attack, with its loss curve demonstrating reverse local peaks, leading to substantial errors during inference. In contrast, the TGO-optimized SNN maintains robust performance, with its global optimum loss remaining largely unaffected by the attack. The right part shows CIFAR-100 results on WideResNet16 for clean, FGSM, and RFGSM attacks with varying $\lambda_{\max}$.

H Visualization of Gradient Sparsity and Loss Landscape

To assess the effectiveness of TGO in adversarial environments, we visualized the Gradient Sparsity and loss landscape of the BPTT-based WR16 model under RFSGM attack, comparing TGO optimization with the vanilla SNN. The Gradient Sparsity quantifies the gradient $\nabla_x f_y$ between the label and the input image.

Gradient Sparsity: The gradient visualization begins with an input image $\mathbf{I}_i \in \mathbb{R}^{C\times H\times W}$, where C denotes the number of channels (e.g., RGB), and $H \times W$ represents the image's spatial resolution. A pre-trained model $f(\mathbf{I}; \theta)$, parameterized by θ, maps the input image to a probability distribution over K classes, denoted as $\mathbf{p} \in \mathbb{R}^K$. The prediction process is given by:

$$\mathbf{p} = f(\mathbf{I}_i; \theta), \quad \mathbf{p}[y_i] = \frac{\exp(z_{y_i})}{\sum_{k=1}^{K} \exp(z_k)}, \tag{39}$$

where z_k is the pre-activation value for class k prior to the application of the softmax function. The model is optimized using the cross-entropy loss between the predicted probability distribution and the true label y_i, formulated as:

$$\mathcal{L}(\mathbf{I}_i, y_i) = -\log \mathbf{p}[y_i] = -\log \left(\frac{\exp(z_{y_i})}{\sum_{k=1}^{K} \exp(z_k)} \right). \tag{40}$$

To assess the sensitivity of the model's predictions to the input, the gradient of the loss with respect to the input image is computed, resulting in a gradient tensor $\mathbf{g}_i \in \mathbb{R}^{C\times H\times W}$, expressed as:

$$\mathbf{g}_i = \frac{\partial \mathcal{L}(\mathbf{I}_i, y_i)}{\partial \mathbf{I}_i}, \quad \mathbf{g}_i[c,x,y] = \frac{\partial}{\partial \mathbf{I}_i[c,x,y]} \left(-\log \mathbf{p}[y_i]\right), \tag{41}$$

where c, x, and y correspond to the channel and spatial coordinates of the image. We compute the gradient $\nabla_x f_y(x)$ under two scenarios: (1) a vanilla SNN, and (2) a TGO-optimized SNN, both trained using BPTT. As shown in Fig.6, we visualize $\nabla_x f_y(x)$ for both models on CIRAF00. The experimental results clearly indicate that, compared to the vanilla SNN, the TGO-optimized SNN exhibits sparser gradients with respect to the input image, demonstrating the effectiveness of membrane potential constraints in increasing gradient sparsity in SNNs. These findings also validate that sparse gradients contribute to enhancing the robustness of SNNs.

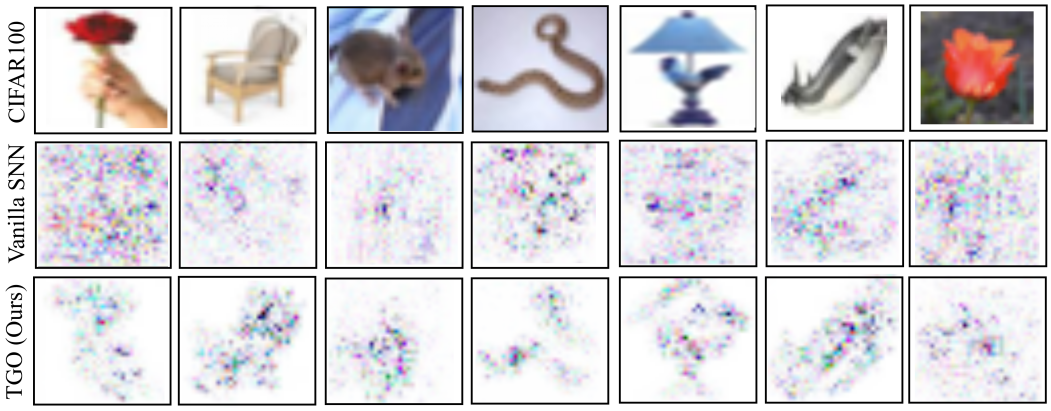

Figure 6: Heatmaps of $\nabla_x f_y$, where f denotes a vanilla SNN (top) or our TGO-optimized SNN (bottom).

Loss Landscape: The process of visualizing the loss landscape starts with a pre-trained model parameterized by $\theta \in \mathbb{R}^M$, where M represents the total number of parameters in the model. The performance of the model is quantified using a loss function $\mathcal{L}(\theta)$, defined over a dataset of N samples as:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f(\mathbf{I}_i;\theta), y_i), \quad \ell(f(\mathbf{I}_i;\theta), y_i) = -\log \mathbf{p}[y_i], \tag{42}$$

where $f(\mathbf{I}_i;\theta)$ denotes the output of the model for input $\mathbf{I}_i$, ℓ is the sample-wise loss (e.g., cross-entropy), and $(\mathbf{I}_i, y_i)$ are the input and label for the i-th sample. The predicted probability distribution $\mathbf{p} \in \mathbb{R}^K$ is computed via softmax as:

$$\mathbf{p}[y_i] = \frac{\exp(z_{y_i})}{\sum_{k=1}^{K}\exp(z_k)}, \quad z_k = f_k(\mathbf{I}_i;\theta), \tag{43}$$

where z_k is the pre-activation value for class k, and K is the total number of classes. To visualize the local geometry of $\mathcal{L}(\theta)$ around a specific parameter set θ, two directions $\mathbf{d}_1, \mathbf{d}_2 \in \mathbb{R}^M$ are defined, satisfying:

$$\|\mathbf{d}_1\| = \|\mathbf{d}_2\| = 1, \quad \mathbf{d}_1^\top \mathbf{d}_2 = 0. \tag{44}$$

Typically, $\mathbf{d}_1$ is chosen as the gradient direction:

$$\mathbf{d}_1 = \frac{\nabla_\theta \mathcal{L}(\theta)}{\|\nabla_\theta \mathcal{L}(\theta)\|}, \quad \nabla_\theta \mathcal{L}(\theta) = \frac{\partial \mathcal{L}(\theta)}{\partial \theta}, \tag{45}$$

while $\mathbf{d}_2$ is sampled randomly and orthogonalized to $\mathbf{d}_1$. The perturbed parameter vector is then expressed as:

$$\theta' = \theta + \alpha \mathbf{d}_1 + \beta \mathbf{d}_2, \tag{46}$$

where $\alpha, \beta \in \mathbb{R}$ control the magnitudes of the perturbations along $\mathbf{d}_1$ and $\mathbf{d}_2$. At each perturbation point (α, β), the loss is computed as:

$$\mathcal{L}(\theta') = \frac{1}{N} \sum_{i=1}^{N} \ell(f(\mathbf{I}_i; \theta + \alpha \mathbf{d}_1 + \beta \mathbf{d}_2), y_i). \tag{47}$$

The parameter space is discretized into a grid of perturbation values $\alpha \in [-\alpha_{\text{max}}, \alpha_{\text{max}}]$ and $\beta \in [-\beta_{\text{max}}, \beta_{\text{max}}]$, such that the loss values form a matrix:

$$\mathbf{L} = [\mathcal{L}(\theta + \alpha \mathbf{d}_1 + \beta \mathbf{d}_2)] \in \mathbb{R}^{n_\alpha \times n_\beta}, \tag{48}$$

where n_α and n_β are the number of grid points along the α and β directions, respectively. Finally, $\mathbf{L}$ is visualized as either a 3D surface or a 2D contour plot:

$$\mathcal{L}(\alpha, \beta) = \mathcal{L}(\theta + \alpha \mathbf{d}_1 + \beta \mathbf{d}_2) \in \mathbb{R}^3, \tag{49}$$

where the gradient of the loss surface can reveal the flatness, sharpness, or other geometric properties of $\mathcal{L}(\theta)$ in the vicinity of the parameter set. As shown in the left part of Fig. 5, the 3D loss landscape reveals that the TGO-optimized model maintains stable loss convergence, demonstrating robust performance. In contrast, the vanilla SNN exhibits localized reverse peaks in the loss landscape under attack. This instability is due to the FGSM gradient-based attack, which perturbs the input along the loss gradient, causing significant fluctuations in the loss. Specifically, FGSM computes the gradient of the loss with respect to the input image and perturbs the input to maximize the loss, pushing the vanilla SNN into regions of the loss landscape that are highly sensitive to small perturbations. In contrast, the TGO-optimized model exhibits smoother transitions in the loss landscape, indicating that its optimization enhances stability against adversarial attacks.

I LIMITATIONS

The limitations of this study include the performance evaluation of Our TGO method on larger model architectures, primarily because existing research predominantly utilizes these specific network structures and their corresponding datasets. For comparative consistency, we maintained these established architectures. Additionally, we have not addressed deployment challenges related to hardware transitions, nor conducted robustness testing in authentic edge-computing adversarial environments. These limitations will be addressed in future research. The experimental results presented in this paper are reproducible, with detailed explanations of model training and configuration provided in the main text and supplemented in the appendix. Our code and models will be made publicly available on GitHub upon acceptance of this paper.

J USE OF LARGE LANGUAGE MODEL

In preparing this manuscript, we utilize a large language model (LLM) solely to aid and polish the writing. The LLM is used for grammar checking, language refinement, and improving clarity of expression. It does not contribute to the formulation of research ideas, methodology, experiments, data analysis, or conclusions. All presented in this paper is entirely the work of the authors.