RSC-SNN: Exploring the Trade-off Between Adversarial Robustness and Accuracy in Spiking Neural Networks via Randomized Smoothing Coding

Keming Wu*
Chongqing University
Chongqing, China
wukemingcqu@gmail.com

Man Yao*
Institute of Automation, Chinese Academy of Sciences
Beijing, China
man.yao@ia.ac.cn

Yuhong Chou
Xi'an Jiaotong University
Xi'an, China
yuhong_chou@outlook.com

Xuerui Qiu
Institute of Automation, Chinese Academy of Sciences
Beijing, China
qiuxuerui2024@ia.ac.cn

Rui Yang
Software Security Technology Company Ltd
Beijing, China
Yangrui@softsafe-tech.com

Bo Xu
Institute of Automation, Chinese Academy of Sciences
Beijing, China
xubo@ia.ac.cn

Guoqi Li†
Institute of Automation, Chinese Academy of Sciences
Beijing, China
guoqi.li@ia.ac.cn

Abstract

Spiking Neural Networks (SNNs) have received widespread attention due to their unique neuronal dynamics and low-power nature. Previous research empirically shows that SNNs with Poisson coding are more robust than Artificial Neural Networks (ANNs) on small-scale datasets. However, it is still unclear in theory how the adversarial robustness of SNNs is derived, and whether SNNs can still maintain its adversarial robustness advantage on large-scale dataset tasks. This work theoretically demonstrates that SNN's inherent adversarial robustness stems from its Poisson coding. We reveal the conceptual equivalence of Poisson coding and randomized smoothing in defense strategies, and analyze in depth the trade-off between accuracy and adversarial robustness in SNNs via the proposed Randomized Smoothing Coding (RSC) method. Experiments demonstrate that the proposed RSC-SNNs show remarkable adversarial robustness, surpassing ANNs and achieving state-of-the-art robustness results on large-scale dataset ImageNet. Our open-source implementation code is available at *https://github.com/KemingWu/RSC-SNN*.

CCS Concepts

• **Computing methodologies → Computer vision; Bio-inspired approaches**.

Keywords

Spiking Neural Networks, Adversarial Learning, Randomized Smoothing

ACM Reference Format:
Keming Wu, Man Yao, Yuhong Chou, Xuerui Qiu, Rui Yang, Bo Xu, and Guoqi Li. 2024. RSC-SNN: Exploring the Trade-off Between Adversarial Robustness and Accuracy in Spiking Neural Networks via Randomized Smoothing Coding. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24), October 28-November 1, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3664647.3680639

1 Introduction

Owing to the distinctive event-driven nature [3] and remarkable biological plausibility [11], SNNs have gained recognition as the third generation of artificial neural networks [18, 25]. Compared with ANNs, SNNs employ discrete binary signals for information transfer among spiking neurons, where spikes are generated solely when the membrane potential surpasses the firing threshold. After deployment to neuromorphic chips [6, 20, 22, 37], SNNs have demonstrated their effectiveness and efficacy in a variety of scenarios, including static visual tasks[35, 36], dynamic visual processing [10, 38], speech classification [24, 39].

Direct [33] and Poisson coding [31] are two popular coding strategies for SNNs, which define how information is represented via spike patterns [7]. For SNNs, a static input $x \in \mathbb{R}^d$ needs to be converted into a time sequence input using coding strategies. Direct coding will repeatedly input x for T times. Poisson coding

[*] Equal contribution.
[†] Corresponding author.

MM '24, October 28-November 1, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0686-8/24/10
https://doi.org/10.1145/3664647.3680639

uses frequency approximation to generate T binary spikes $\{p_i\}_{i=1}^{T}$ so that the average number of spikes approximates the intensity of the pixel $\frac{1}{T}\sum_{i=1}^{T} p_i \approx x$. Coding methods play a crucial role in determining the network's computational efficiency and resilience to perturbations. Existing empirical studies have shown that the adversarial robustness of SNNs using Poisson coding is higher than that of ANNs, and the robustness decreases as the time step increases [28]. In contrast, SNNs using direct coding have poorer adversarial robustness than ANNs [15].

The impact of direct coding and Poisson coding on the adversarial robustness has not been systematically analyzed, which undermines the potential advantages of SNNs over ANNs in terms of adversarial robustness. Moreover, it is still unknown whether SNN can still maintain the adversarial robustness advantage on large-scale tasks, because previous work has only been verified on small datasets such as CIFAR-10/100 [4, 17, 21, 27]. We are interested in why the adversarial robustness of SNNs employing Poisson coding is stronger. We note that randomized smoothing and Poisson coding have similar features in enhancing adversarial robustness, although they may seem like two different approaches. Randomized smoothing builds a base classifier by introducing noise [5], in contrast, Poisson coding converts the input into a binary probability.

Inspired by this, we theoretically establish the connection between randomized smoothing and Poisson coding via analyzing the statistical characteristics of them. We found that Poisson coding shares fundamental statistical properties with randomized smoothing, such as expectation and variance, introducing similar noise smoothing. Since existing research has shown that randomized smoothing can bring certified adversarial robustness, based on this observation, we can understand why SNNs using Poisson coding have adversarial robustness. However, our theoretical analysis shows that SNNs with Poisson coding are greatly affected by perturbation while bringing about the problem of reduced clean accuracy. Therefore, it is urgent to establish a guiding principle for the trade-off between accuracy and robustness in the design of defense methods against adversarial examples in SNNs. We analyze in depth the trade-off between accuracy and adversarial robustness in SNNs via a novel Randomized Smoothing Coding (RSC) method, which significantly improves the adversarial robustness of SNNs. To further exploit the potential of this approach, we propose a new training method designed for RSC-SNN. Experimental results on extensive datasets show that randomized smoothing coding greatly enhances the adversarial robustness of SNNs. Simultaneously, The final results indicate a trade-off between accuracy and adversarial robustness, which is consistent with the conclusions of ANNs [29, 30, 41]. Furthermore, We also propose an empirical estimation method to quantify the trade-off called Quantification Trade-off Estimation (QTE) to help design defense methods with better trade-offs. The main contributions of our work are summarized as follows:

- We establish the connection between Poisson coding and randomized smoothing for the first time, which is a novel insight contributing to the field. Furthermore, we prove the conceptual equivalence of randomized smoothing and Poisson coding, which provides a theoretical foundation for the robustness of Poisson-encoded SNNs. We proposed a novel coding method called **randomized smoothing coding**.
- Observing the inherent clean accuracy drop caused by randomized smoothing coding, we propose a new training method called Efficient Randomized Smoothing Coding Training (**E-RSCT**) specifically for randomized smoothing coding.
- Experimental results show that RSC-SNNs show remarkable adversarial robustness in image recognition and achieves state-of-the-art results on datasets including large datasets Tiny-ImageNet, ImageNet, while achieving a better trade-off under the metric of Quantification Trade-off Estimation.

2 Background and Related Work

2.1 Spiking Neural Network

Spiking neurons are the basic units of SNNs, which are abstracted from the dynamics of biological neurons. The leaky-integrate-and-fire (LIF) neuron model is widely acknowledged as the simplest model among all popular neuron models while maintaining biological interpretability, in contrast to the many-variable and complex H-H model [14]. It also has a significantly lower computational demand [23, 26]. We adopt the LIF neuron model and translate it to an iterative expression with the Euler method [32]. Mathematically, the LIF-SNN layer can be described as an iterable version for better computational traceability:

$$\begin{cases} u_i^{(l)}[t+1] = h_i^{(l)}[t] + f(w^{(l)}, x_i^{(l-1)}[t]) \\ s_i^{(l)}[t] = \Theta(u_i^{(l)}[t+1] - \vartheta) \\ h_i^{(l)}[t+1] = \tau u_i^{(l)}[t+1](1 - s_i^{(l)}[t]), \end{cases} \tag{1}$$

where τ is the time constant, t and i respectively represent the indices of the time step and the l-th layer, w denotes synaptic weight matrix between two adjacent layers, $f(\cdot)$ is the function operation stands for convolution (Conv) or fully connected (FC), x is the input, and $\Theta(\cdot)$ denotes the Heaviside step function. When the membrane potential u exceeds the firing threshold ϑ, the LIF neuron will trigger a spike S. Moreover, h represents the membrane potential after the trigger event which equals τu.

2.2 Adversarial Attacks

Adversarial attacks are designed to fool a model into incorrect predictions or outputs through carefully crafted inputs [12]. Given a classifier $f : \mathbb{R}^d \to \mathcal{Y}$, where $\mathcal{Y}$ is the set of class labels, the purpose of an adversarial perturbation δ is to make $f(x+\delta) \neq f(x)$, which can be formulated as an optimization problem:

$$\max_{\|\delta\|_p \leq \epsilon} \mathcal{L}\left(f\left(x+\delta\right), y\right), \tag{2}$$

where f is the network under attack, $\mathcal{L}$ is the loss function, x, y are the input and target output of the given network, respectively. ϵ is a parameter that limits the intensity of the perturbation so that it is not easily observed by the human eye. δ is the parameter we want to optimize. In this paper we mainly use two widely adopted gradient-based adversarial attacks: Fast Gradient Sign Method (FGSM) and Projected Gradient Descent method (PGD).

FGSM. As a simple but effective attack method [12], adversarial examples are generated based on the symbolic information of the gradient to maximize the loss of the perturbed $x+\delta$, which can be formulated as

$$x_{adv} = x + \epsilon \times \operatorname{sign}\left(\nabla_x L\left(f\left(x, y\right)\right)\right), \tag{3}$$

where ϵ denotes the strength of the attack.

PGD. As an iterative version of FGSM, it generates adversarial samples by adding small perturbations in the gradient direction multiple iterations and limiting the results to a certain range after each iteration [19], which can be formulated as

$$\mathbf{x}_{adv}^{(k)} = \Pi_\epsilon \left\{ \mathbf{x}_{adv}^{(k-1)} + \alpha \times \operatorname{sign}\left(\nabla_{\mathbf{x}} L \left(f \left(\mathbf{x}_{adv}^{(k-1)}, y \right) \right) \right) \right\}, \quad (4)$$

where k denotes the number of the iteration step and α is the step size of each iteration. Π_ϵ is used to ensure that the perturbation does not exceed a predefined range ϵ.

For FGSM and PGD, we explore two scenarios: white-box and black-box attacks. In the white-box scenario, the attacker possesses full access to the model's topology, parameters, and gradients. Conversely, in the black-box scenario, the attacker is limited to basic information about the model. Without specific guidelines, we fix ϵ at 8/255 across all methods for testing. For iterative techniques PGD, the attack step is set at $\alpha = 0.01$, with a total of 7 steps.

2.3 Randomized Smoothing

As a strategy aimed at enhancing model adversarial robustness, randomized smoothing fortifies the model's defense against attacks through the inclusion of random noise [5]. In addition, there are many works that further explore randomized smoothing [16, 34, 40].

In a classification scenario mapping from $\mathbb{R}^d \rightarrow \mathcal{Y}$, randomized smoothing constitutes a method to formulate a refined classifier g from any base classifier f. When evaluated at x, the smoothed classifier g identifies the class that the base classifier f is most inclined to predict when x undergoes perturbation by isotropic Gaussian noise:

$$g(x) = \underset{c \in \mathcal{Y}}{\arg\max}\, \mathbb{P}\left(f(x+\epsilon) = c \right), \quad (5)$$

where $\epsilon \sim \mathcal{N}\left(0, \sigma^2 I\right)$. The noise level σ serves as a hyperparameter for the smoothed classifier g, dictating a trade-off between adversarial robustness and accuracy.

3 Method

As aforementioned, we suggest achieving a better trade-off between adversarial robustness and accuracy for designing more practical SNN models. In this section, we first provide an empirical metric to quantify the trade-off between adversarial robustness and accuracy. We then introduce a new coding method called RSC to improve the adversarial robustness, which enhances the quantitative trade-off between adversarial robustness and accuracy. Furthermore, theoretical analysis is given to illustrate the conceptual equivalence of RSC and Poisson coding. After observing the inherent limitations of RSC, we further propose a specified training method designed for RSC to improve its clean accuracy and adversarial robustness.

3.1 Quantification Trade-off Estimation

Quantification trade-off is important for designing methods to better trade-off. Therefore, before giving full details of our methods, we first try to formulate a Quantification Trade-off Estimation.

Definition 3.1. *Quantification Trade-off Estimation (QTE).* From Figure 1, we can get that the process of achieving better trade-offs is also essentially making the area larger. For a trained model, assume that its accuracy under attack intensity η is $A(\eta)$, Quantification Trade-off Estimation between two attack intensities η_a and η_b can be formulated as

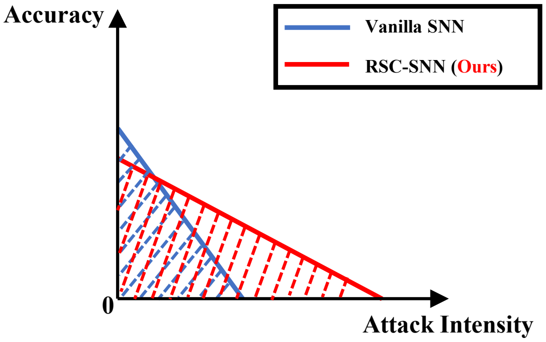


Figure 1: An illustration of the trade-off between adversarial robustness and accuracy can be represented by the absolute value of the slope, indicating the SNN model's adversarial robustness. The area of a triangle can quantitatively estimate this trade-off. It is important to note that the slope illustrates the correlation between accuracy and attack strength, rather than implying a specific linear relationship.

$$\text{QTE}=\left| \frac{(\eta_b - \eta_a)\left(A(\eta_b) + A(\eta_a)\right)}{2} \right|. \quad (6)$$

Obviously, a larger Quantification Trade-off Estimation implies higher overall accuracy within the attack interval, signifying an improved trade-off in the model implementation. Simultaneously, it's noticeable that a smaller difference in $|\eta_b - \eta_a|$ corresponds to more accurate estimations of the model's quantification trade-offs.

3.2 Randomized Smoothing Coding (RSC)

Motivated by randomized smoothing, we first introduced random smoothing of Gaussian noise into the input of SNN. For each sample input $x \in \mathbb{R}^{3\times H\times W}$, there is a given time step T. There are actually two ways to add noise to the input sample of the SNN model. One is to directly add a fixed noise a to the input and no longer changes at each time step, which can be formulated as

$$\widetilde{x} = x + \epsilon, \epsilon \sim \mathcal{N}\left(0, \sigma^2 I\right), \quad (7)$$

it can also be expressed as

$$\widetilde{x} \sim \mathcal{N}\left(x, \sigma^2 I\right). \quad (8)$$

Another way is to add different Gaussian noises at each time step, which can be formulated as

$$\widetilde{x_i} = x_i + \epsilon_i, \epsilon_i \sim \mathcal{N}\left(0, \sigma^2 I\right), i = 1, \cdots, |T|, \quad (9)$$

where the noise level σ serves as a hyperparameter for the model, dictating a trade-off between adversarial robustness and accuracy.

After adding noise to x, we further limit the range of $\widetilde{x}$ to $[0, 1]$, which can be formulated as

$$\widetilde{x}_{\text{clamp}} = \operatorname{clamp}\left(\widetilde{x}\right) = \operatorname{clamp}\left(x+\epsilon\right), \widetilde{x}_{\text{clamp}} \in [0, 1]. \quad (10)$$

For the first method, we named it RSC-I, and the second method RSC-II. In the experiment, we found that RSC-I can bring about a substantial improvement in adversarial robustness compared to RSC-II, and the degree of clean accuracy decrease will be higher than that of RSC-II; RSC-II can also bring about a small improvement in adversarial robustness, and the degree of clean accuracy decrease will be lower than that of RSC-I. Due to the need for higher adversarial robustness, we choose RSC-I as our main method.

3.3 Theoretical Analysis for Randomized Smoothing Coding

For the convenience of analysis, we first explain the meaning of the symbols. The input image is denoted as $\boldsymbol{x}$. The expectation of $\boldsymbol{X}$ is denoted as $\boldsymbol{E}[\boldsymbol{X}]$ and the covariance matrix is denoted as $\Sigma_{\boldsymbol{X}}$.

For Poisson coding, the input is a random vector $\boldsymbol{X_P}$, where the vector follows a Bernoulli binomial distribution with probability vector $\boldsymbol{p}$. For randomized smoothing, the input is a random vector $\boldsymbol{X_{RS}}$, which follows a normal distribution centered at $\boldsymbol{x}$ with covariance $\boldsymbol{\sigma}^2$, denoted as $\boldsymbol{X_{RS}} \sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{\sigma}^2)$. The following formula can be satisfied:

$$\begin{aligned} &\boldsymbol{E}[\boldsymbol{X_P}] = \boldsymbol{E}[\boldsymbol{X_{RS}}] = \boldsymbol{x} \\ &\Sigma_{\boldsymbol{X_P}} = \mathrm{diag}(\boldsymbol{x}(1-\boldsymbol{x})) = \mathrm{diag}(x_1(1-x_1), \ldots, x_d(1-x_d)). \\ &\Sigma_{\boldsymbol{X_{RS}}} = \mathrm{diag}(\boldsymbol{\sigma}^2) = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2). \end{aligned} \tag{11}$$

We can observe that Poisson coding and randomized smoothing coding can be considered as an equivalence in theory.

To better explain why randomized smoothing coding is better than Poisson coding, the results of the two different perturbed coding methods in a linear layer are explored through the following theorem. For the convenience of expression, we express the adversarial sample as $\boldsymbol{x}+\boldsymbol{\epsilon}$ and denote the linear layer as a deterministic weight matrix $\boldsymbol{W}$, the original output as a random variable $\boldsymbol{Y_{P/RS_{original}}}$ and the attacked output as a random variable $\boldsymbol{Y_{P/RS_{attack}}}$. For both coding methods, the expectation of original $\boldsymbol{Y_{original}}$ and attacked $\boldsymbol{Y_{attack}}$ are given by:

$$\begin{aligned} \boldsymbol{E}[\boldsymbol{Y_{original}}] &= \boldsymbol{E}[\boldsymbol{WX}] = \boldsymbol{W}\boldsymbol{E}[\boldsymbol{X}] = \boldsymbol{Wx}. \\ \boldsymbol{E}[\boldsymbol{Y_{attack}}] &= \boldsymbol{W}(\boldsymbol{x}+\boldsymbol{\epsilon}). \end{aligned} \tag{12}$$

Although the expectations of both exhibit the same characteristics, their covariances behave differently. In Theorems 3.2 and 3.3, we obtain two covariance results with Poisson coding and randomized smoothing coding.

Theorem 3.2. *The covariance matrix of Poisson coding before and after the attack satisfies:*

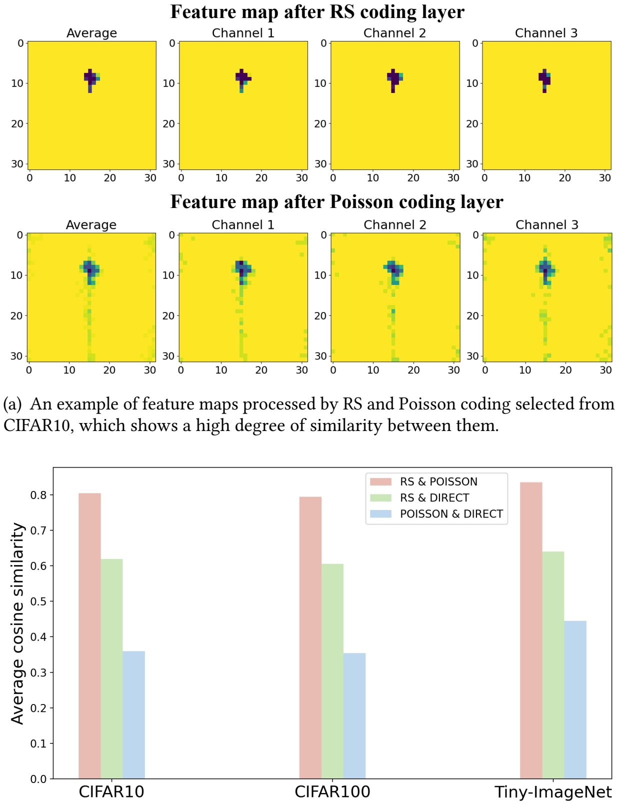
$$\begin{aligned} \Sigma_{\boldsymbol{Y_{P_{original}}}} &= \boldsymbol{W} diag(\boldsymbol{x}(1-\boldsymbol{x}))\boldsymbol{W}^T. \\ \Sigma_{\boldsymbol{Y_{P_{attack}}}} &= \boldsymbol{W} diag(\boldsymbol{x}(1-\boldsymbol{x}) + \boldsymbol{\epsilon}(1-2\boldsymbol{x}) - \boldsymbol{\epsilon}^2)\boldsymbol{W}^T. \end{aligned} \tag{13}$$

Theorem 3.3. *The covariance matrix of randomized smoothing coding before and after the attack satisfies:*

$$\Sigma_{\boldsymbol{Y_{RS_{original/attack}}}} = \boldsymbol{W} diag(\boldsymbol{\sigma}^2)\boldsymbol{W}^T. \tag{14}$$

All the above theorems are proved in the supplementary material.

Proposition 3.4. ***(Covariance invariance of RSC.)*** *Poisson coding and randomized smoothing coding demonstrate distinct characteristics regarding covariance. For randomized smoothing coding, the noise variance introduced to the input remains constant and does not vary with the input itself.*



(a) An example of feature maps processed by RS and Poisson coding selected from CIFAR10, which shows a high degree of similarity between them.

(b) The average cosine similarity among three coding methods on CIFAR10, CIFAR100, and Tiny-ImageNet datasets indicates a notably high resemblance between randomized smoothing coding and Poisson coding.

Figure 2: Visual verification of equivalence of randomized smoothing coding and Poisson coding.

While Poisson coding and randomized smoothing share the high-level objective of enhancing adversarial robustness through noise averaging, they differ significantly in the specific nature of the noise added. Randomized smoothing employs isotropic Gaussian noise, resulting in predictable and manageable outcomes due to its uniform variance in all directions. In contrast, Poisson coding introduces noise whose covariance is influenced by both the input magnitude and perturbation attacks, leading to input-dependent variance. Consequently, the constant noise variance in randomized smoothing ensures a more stable and consistent smoothing process, potentially making it more effective in certain scenarios.

Observation 3.5. ***(Equivalence between RSC and Poisson coding.)*** *The image inputs to SNNs, following processing by the randomized smoothing coding layer, exhibit similar characteristics to that of the Poisson coding layer, displaying a notably high cosine similarity.*

The phenomenon in Figure 2(a) stems from coding properties. Randomized Smoothing regularizes feature maps by smoothing out irregularities, enhancing robustness. On the contrary, Poisson coding, introduces binary spikes based on pixel intensity, potentially resulting in a noisier feature map. This observation is clearly depicted in Figure 2(a) and 2(b), where the average cosine similarity of randomized smoothing coding and Poisson coding stands at a notably elevated level, suggesting a equivalence between them. More visualization results are shown in the supplementary material.

Table 1: Checklist for characteristic behaviors caused by obfuscated and masked gradients.

Items to identify gradient obfuscation	Test
(1) One-step attacks perform better than iterative attacks	Pass
(2) Black-box attacks are better than white-box attacks.	Pass
(3) Unbounded attacks do not reach 100% success.	Pass
(4) Increasing distortion bound does not increase success.	Pass
(5) Random sampling finds adversarial examples.	Pass

Table 2: Results show that our proposed method can still defend against EOT attacks.

Dataset	Architecture	Methods	Clean	PGD	EOTPGD
CIFAR10	VGG-5	RSC-0.1	80.29	37.28	28.72
	VGG-5	RSC-0.5	78.67	58.94	48.93
CIFAR100	VGG-11	RSC-0.1	57.05	24.35	18.95
	VGG-11	RSC-0.5	56.25	33.96	27.14

3.4 Checks for RSC Gradient Obfuscation

Gradient obfuscation is the main reason why many adversarial defense methods are mistakenly considered effective. By certain methods, the neural network cannot produce accurate gradients, resulting in the inability to produce effective attacks.

To evaluate the attack effectiveness of the RSC , we employ the systematic checklist presented in [1] to scrutinize the gradient obfuscation of this novel coding scheme. This assessment is primarily grounded in the data delineated in Table 3 and Table 4 within the main body of the text, with a brief summary provided in Table 1. The detailed analysis can be found in the supplementary material.

Expectation over Transformation (EOT) computes the gradient over the expected transformation to the input as a method to attack randomized models [2]. We use EOTPGD proposed by Zimmermann [42] to evaluate the effectiveness of our method. The results in Table 2 show that our method can defend against EOT attacks.

3.5 Efficient-RSC Training (E-RSCT)

We found that while RSC improved the SNN model's adversarial robustness, it also led to a certain degree of decline in clean accuracy. To alleviate this problem, we propose E-RSCT for training SNN models using RSC. Motivated by the idea of knowledge distillation, we found that the corresponding ANN have naturally high clean accuracy, so we decided to use a pre-trained ANN as a teacher model to transfer the learned knowledge to the RSC-SNN model.

E-RSCT consists of two parts of loss functions during the training process. First, in order to transfer the learned knowledge from the teacher model [13], we define the loss function of the first part as $\mathcal{L}_{KD}$ and use KL divergence to measure the difference between the student network and the teacher network to get $\mathcal{L}_{KD}$.

$$\mathcal{L}_{KD} = KL\left(O_{stu}, O_{tea}\right), \tag{15}$$

where O_{tea} is the output of the teacher ANN model and O_{stu} is the output of the student SNN model.

For the second part of the loss function, we use the pre-synaptic loss $\mathcal{L}_{P-S}$ [8] in training.

By combining $\mathcal{L}_{KD}$ and $\mathcal{L}_{P-S}$, we get the new loss function used for the novel training as follows

$$\mathcal{L}_{E-RSC} = \lambda\mathcal{L}_{KD} + \mathcal{L}_{P-S}, \tag{16}$$

where λ is used to achieve a trade-off between $\mathcal{L}_{KD}$ and $\mathcal{L}_{P-S}$. Without special explanation, we select $\lambda = 0.1$ in the experiment. The detail of E-RSCT is shown in Algo.1.

Algorithm 1 Training process of E-RSCT for one epoch.

```
  Input: An SNN to be trained with RSC; a hyperparameter $\sigma$;
         training dataset; total training iteration: $I_{\text{train}}$.
Output: The well-trained SNN.
for all $i = 1, 2, \ldots, I_{\text{train}}$ iteration do
    Get mini-batch training data, $\boldsymbol{x}_{\text{in}}(i)$ and class label, $\boldsymbol{y}(i)$;
    Feed the $\boldsymbol{x}_{\text{in}}(i)$ into the SNN ;
    Generate new sample $\widetilde{x}$ by Eq. 7;
    Process $\widetilde{x}$ to get $\widetilde{x}_{\text{clamp}}$ by Eq. 10;
    Calculate the SNN output, $\boldsymbol{o}_{\text{out}}(i)$ by Eq. 1 ;
    Compute the loss function $\mathcal{L}_{E-RSC} = \lambda\mathcal{L}_{KD} + \mathcal{L}_{P-S}$ by
    Eq. 16;
    Backpropagation and update model parameters;
end for
```

4 Experiments

4.1 Experimental Setup

We verify the effectiveness of the proposed RSC and E-RSCT on multiple datasets and compare with ANNs simultaneously. In the following experiments, SNNs using direct coding are represented by DIRECT, SNNs using RS coding are represented by RSC and SNNs using Poisson coding are represented by POISSON. For the trade-off measurement of the model under different attacks, we use F-QTE to represent the Quantification Trade-off Estimation under the FGSM attack, and P-QTE to represent the Quantification Trade-off Estimation under the PGD attack. Detailed implementation is referred to in the supplementary material. For adversarial robustness evaluation, specific hyperparameter settings are introduced in Section 2.2. In the experiments of E-RSCT, we used the hyperparameters set in Section 3.5.

4.2 Performance under attacks.

Clean accuracy and adversarial robustness. Clean accuracy refers to the accuracy on the clean test dataset. It was represented as CLEAN in the experiment. The evaluation of adversarial robustness accuracy is denoted as FGSM and PGD respectively.

Results on different datasets. Table 3 illustrates the performance evaluation of our proposed RSC scheme. For constructing

Table 3: White-box attack results on four datasets of ANN and three different coding methods of SNN. The best result is highlighted with bold and the second with underlined. The larger the better for all metrics.

Dataset	Architecture	Coding Methods	Clean	FGSM	PGD	F-QTE	P-QTE
CIFAR10	VGG-5	ANN	90.95	10.89	0.12	4.07	3.64
	VGG-5	Direct	90.69	6.19	0.03	3.88	3.63
	VGG-5	Poisson	83.18	31.20	22.16	4.58	4.21
	VGG-5	RSC-0.1(**Ours**)	80.29	51.29	37.28	5.26	4.70
	VGG-5	RSC-0.5(**Ours**)	78.67	**66.61**	**58.94**	**5.81**	**5.50**
CIFAR100	VGG-11	ANN	72.86	4.56	0.13	3.10	2.92
	VGG-11	Direct	72.45	4.67	0.22	3.08	2.91
	VGG-11	Poisson	58.49	19.46	15.56	3.12	2.96
	VGG-11	RSC-0.1(**Ours**)	57.05	32.48	24.35	3.58	3.22
	VGG-11	RSC-0.2(**Ours**)	55.42	37.83	29.69	**3.73**	3.40
	VGG-11	RSC-0.5(**Ours**)	51.30	**40.81**	**33.96**	3.68	**3.41**
Tiny-ImageNet	VGG-16	ANN	60.77	2.08	0.00	2.51	2.43
	VGG-16	Direct	57.90	2.04	0.01	2.40	2.32
	VGG-16	Poisson	48.14	6.79	2.68	2.20	2.03
	VGG-16	RSC-0.01(**Ours**)	48.33	7.73	2.15	2.24	2.02
	VGG-16	RSC-0.1(**Ours**)	47.47	**22.63**	**13.75**	**2.80**	**2.45**
ImageNet	ResNet-19	ANN	67.00	0.66	0.00	2.71	**2.68**
	ResNet-19	Direct	56.41	2.57	0.02	2.36	2.26
	ResNet-19	Poisson	40.21	10.61	2.68	2.03	1.72
	ResNet-19	RSC-0.1(**Ours**)	44.25	17.73	**8.50**	2.48	2.11
	SEW-ResNet-18	Direct	64.40	4.56	0.00	2.76	2.58
	SEW-ResNet-18	Poisson	52.29	15.73	4.70	2.72	2.28
	SEW-ResNet-18	RSC-0.1(**Ours**)	53.79	**25.86**	7.38	**3.19**	2.45

Table 4: Black-box attack results on four datasets of ANN and three different coding methods of SNN. The best result is highlighted with bold and the second with underlined. The larger the better for all metrics.

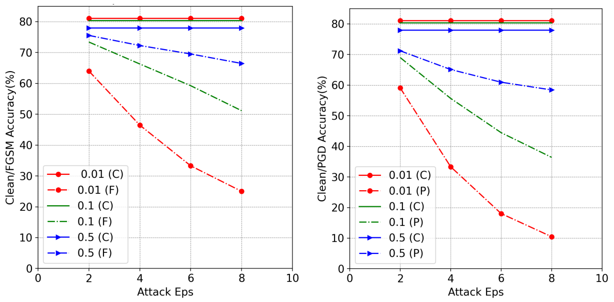
Dataset	Architecture	Coding Methods	Clean	FGSM	PGD	F-QTE	P-QTE
CIFAR10	VGG-5	Direct	90.69	20.75	3.52	4.46	3.77
	VGG-5	Poisson	83.18	43.06	36.15	5.05	4.77
	VGG-5	RSC-0.1(**Ours**)	80.29	**59.25**	**49.55**	**5.58**	**5.19**
CIFAR100	VGG-11	Direct	72.45	11.79	4.08	3.37	3.06
	VGG-11	Poisson	58.49	31.03	27.36	3.58	3.43
	VGG-11	RSC-0.1(**Ours**)	57.05	**41.74**	**35.56**	**3.95**	**3.70**
Tiny-ImageNet	VGG-16	Direct	57.90	14.82	8.15	2.91	2.64
	VGG-16	Poisson	48.14	21.22	16.73	2.77	2.59
	VGG-16	RSC-0.1(**Ours**)	47.47	**35.06**	**29.40**	**3.30**	**3.07**
ImageNet	SEW-ResNet-18	Direct	64.40	15.75	11.89	3.21	3.05
	SEW-ResNet-18	Poisson	52.29	16.87	16.65	2.77	2.76
	SEW-ResNet-18	RSC-0.1(**Ours**)	53.79	**29.52**	**27.91**	**3.33**	**3.27**

effective attacks on SNN, all gradient attacks are applied based on BPTT. The specific implementation of BPTT is in the supplementary material. The results consistently demonstrate the efficacy of our RSC in enhancing model adversarial robustness, as evidenced by notable enhancements in adversarial robustness accuracy across all attack methodologies. Particularly striking is the significant improvement in adversarial robustness against stronger white-box iterative attacks. Notably, the VGG-5 model exhibited a remarkable 58.91% increase in accuracy when attacked by PGD compared to direct coding on the CIFAR-10 dataset. Simultaneously, the improvement of F-QTE and P-QTE also shows that our method better achieves the trade-off between accuracy and adversarial robustness. The experimental results that RSC shows more adversarial robustness than Poisson coding well verify Theorem 3.2 and 3.3.

Experiments on Tiny-ImageNet and ImageNet. Previous work was limited to small datasets such as CIFAR-10 and CIFAR-100. We also conducted experiments on Tiny-ImageNet and ImageNet to provide a baseline for the evaluation of SNN adversarial robustness.

Table 5: Enhanced robustness with adversarial training. The best result is highlighted with bold. The larger the better for all metrics.

Dataset	Architecture	Methods	Clean	FGSM	PGD
CIFAR10	VGG-5	RSC-0.1	80.29	51.29	37.28
	VGG-5	RSC-0.1 + Adv	80.27	**55.58**	**43.59**
CIFAR100	VGG-11	RSC-0.1	57.05	32.48	24.35
	VGG-11	RSC-0.1 + Adv	56.25	**35.53**	**27.57**



(a) Ablation experiments of noise σ^2 - FGSM.

(b) Ablation experiments of noise σ^2 - PGD.

Figure 3: Ablation experiment for noise level σ^2.

It can be seen that the adversarial robustness of RSC-SNN has been significantly improved compared to DIRECT and POISSON.

Black-box attack results on different datasets. In this section, we assess the adversarial robustness of RSC against black-box attacks. We utilize a separately trained SNN with an identical architecture to generate white-box attack samples. The results presented in Table 4 demonstrate that RSC demonstrates significant resilience against adversarial attacks, surpassing traditional SNNs in both FGSM and PGD scenarios. This robustness positions RSC as notably more resilient to adversarial intrusions in black-box scenarios. Compared with traditional SNNs, RSC-SNN's F-QTE and P-QTE have also improved in black-box attack scenarios.

Enhanced Robustness with Adversarial Training. Adversarial training, has been the most widely accepted defense method. To evaluate the combined robustness of RSC and adversarial training, we explored their transferability and scalability. We trained SNN-Direct, SNN-Poisson and RSC using low-intensity FGSM samples and then exposed them to more complex and larger ϵ white-box attacks. Results in Table 5 (rows 1-2 for CIFAR10 and rows 3-4 for CIFAR100) show a significant boost in robustness for RSC when combined with adversarial training. On CIFAR-10, this combination increased robustness for FGSM (to 55.58% from 51.29%) and PGD (to 43.59% from 37.28%). For CIFAR-100, resilience improved against FGSM (to 35.53% from 32.48%) and PGD (to 27.57% from 24.35%). In conclusion, the integration of RSC with adversarial training gives it the versatility to withstand a wider range of more powerful adversarial attacks.

4.3 The effectiveness of E-RSCT.

We can see from Table 3 that while RSC brings significant adversarial robustness improvement to the model, it also leads to clean accuracy decline. To alleviate this problem, we proposed E-RSCT for RSC-SNN. Table 6 summarizes the clean and robust accuracy of SNN models trained with and without E-RSCT. It can be seen from Table 6 that the clean accuracy and adversarial robustness of the model trained using E-RSCT on all datasets have been improved to a certain extent. On the CIFAR10 dataset, while the clean accuracy was improved by 1.74%, its robust accuracy for FGSM and PGD was also improved by 3.23% and 2.70% respectively. On other datasets, an average accuracy improvement of more than 1% has been achieved. The experimental results concretely verify the effectiveness of our proposed training algorithm. The improvement of F-QTE and P-QTE also shows that E-RSCT achieves better trade-offs.

4.4 Ablation Studies

Effect of different noise levels σ^2 on RSC. Investigating the pivotal role of the noise level parameter σ^2 within the novel introduced RSC framework holds significant importance in regulating the model's adversarial robustness. We conducted an exhaustive ablation study to discern the effect of parameter variations on the model's adversarial robustness. Specifically, we selected three distinct values: $\sigma^2 = 0.01$, $\sigma^2 = 0.1$, and $\sigma^2 = 0.5$ to meticulously evaluate the model's adversarial robustness under varying attack intensities FGSM and PGD across multiple values ($\epsilon = 2, 4, 6, 8$). The comprehensive experimental outcomes are visually presented in the Figure 3. The solid line shows clean accuracy, and the dotted line shows post-attack accuracy.

From Figure 3, a noticeable trend emerges: as the noise level (σ^2) escalates, there's a concurrent decline in the model's clean accuracy, a phenomenon congruent with our empirical analyses. Simultaneously, with an increase in σ^2, the model's adversarial robustness exhibits a consistent uptrend. Remarkably, the robust accuracy under various attack intensities showcases a discernible hierarchy, wherein higher σ^2 values correspond to augmented adversarial robustness. Specifically, the variations in robust accuracy across different attack intensities become more pronounced with increasing σ^2. Evidently, this underscores the significance of striking a balance between clean accuracy and adversarial robustness in real-world RSC applications. Achieving this balance necessitates meticulous exploration of σ^2 values to pinpoint the optimal choice aligned with specific application requisites.

4.5 Comparison with State-of-the-art Work on Adversarial Robustness of SNN

To evaluate the effectiveness of our proposed RSC, we compare it with the results of existing state-of-the-art work. We conduct experimental comparisons in the case of FGSM and PGD white-box attacks on the CIFAR-10 and CIFAR-100 datasets. The hyperparameters of the attack are set to $\epsilon = 8/255$ for FGSM and $\alpha = 0.01$ for PGD with a total of 7 steps. The comparison results are shown in Table 7.

CIFAR-10. On the CIFAR-10 dataset, employing the RSC VGG-5 model with a noise level of $\sigma^2 = 0.1$, our approach showcased

Table 6: Comparison of results with E-RSCT and without E-RSCT. The best result is highlighted with bold. The larger the better for all metrics.

Dataset	Architecture	Methods	Clean	FGSM	PGD	F-QTE	P-QTE
CIFAR-10	VGG-5	Baseline	80.29	51.29	37.28	5.26	4.70
	VGG-5	+E-RSCT	**82.03**	**54.52**	**39.98**	**5.46**	**4.88**
CIFAR-100	VGG-11	Baseline	57.05	32.48	24.35	3.58	3.26
	VGG-11	+E-RSCT	**58.04**	**34.89**	**26.67**	**3.72**	**3.39**
Tiny-ImageNet	VGG-16	Baseline	47.47	22.63	13.75	2.80	2.45
	VGG-16	+E-RSCT	**48.29**	**24.01**	**15.46**	**2.89**	**2.55**
ImageNet	SEW-ResNet-18	Baseline	53.79	25.86	7.38	3.19	2.45
	SEW-ResNet-18	+E-RSCT	**54.77**	**27.21**	**8.84**	**3.28**	**2.54**

Table 7: Comparison with others works. The best result is highlighted with bold and the second with underlined. The larger the better for all metrics.

Dataset	Architecture	Methods	FGSM	PGD	F-QTE	P-QTE	Clean
CIFAR10	VGG-5	Baseline	6.19	0.03	3.88	3.63	90.69
	VGG-5	Sharmin et al. [28]^{ECCV}	15.00	3.80	4.17	3.72	89.30
	VGG-5	Kundu et al. [15]^{ICCV}	38.00	9.10	5.02	3.86	87.50
	VGG-5	Ding et al. [9]^{NeurIPS}	45.23	21.16	5.44	4.48	90.74
	VGG-5	**Our work**	**54.52**	**39.98**	**5.46**	**4.88**	82.03
CIFAR100	VGG-11	Baseline	5.30	0.02	3.15	2.93	73.33
	VGG-11	Sharmin et al. [28]^{ECCV}	15.50	6.30	3.20	2.83	64.40
	VGG-11	Kundu et al. [15]^{ICCV}	22.00	7.50	3.48	2.90	65.10
	VGG-11	Ding et al. [9]^{NeurIPS}	25.86	10.38	**3.87**	3.25	70.89
	VGG-11	**Our work**	**34.89**	**26.67**	3.72	**3.39**	58.04
Tiny-ImageNet	VGG-16	Baseline	2.04	0.03	2.40	2.32	57.90
	VGG-16	**Our work**	**24.01**	**15.46**	**2.89**	**2.55**	48.29
ImageNet	SEW-RESNET-18	Baseline	4.56	0.00	2.76	**2.58**	64.40
	SEW-RESNET-18	**Our work**	**27.21**	**8.84**	**3.28**	2.54	54.77

promising advancements. As indicated in the table, our method notably elevated the model accuracy against FGSM and PGD attacks by 48.33% and 39.95%, respectively, in contrast to the vanilla model. Furthermore, in comparison with the best-performing outcomes [9], our approach achieved a substantial enhancement in accuracy by 9.29% for FGSM and 18.82% for PGD attacks.

CIFAR-100. On the CIFAR-100, our utilization of the RSC VGG-11 model with a noise level of $\sigma^2 = 0.1$ yielded notable advancements. The provided table demonstrates that our method elevated accuracy against FGSM and PGD attacks by 29.59% and 26.65%, respectively, surpassing the performance of the vanilla model. In comparison to the top-performing outcomes, our approach showcased compelling enhancements, achieving a remarkable 9.03% accuracy improvement for FGSM and 16.29% for PGD attacks.

Upon comparing our experimental outcomes with the State-of-the-Art (SOTA) approaches on the CIFAR-10 and CIFAR-100 datasets, our proposed method showcased significant enhancements in the model's adversarial robustness. F-QTE and P-QTE also achieve comparable results to SOTA. Notably, while augmenting adversarial robustness, we observed a decline in clean accuracy compared to the vanilla model. Hence, to achieve a balance between clean accuracy and adversarial robustness, we meticulously fine-tuned various hyperparameters involved in the training process. Parameters like the noise level σ, the proportion of the loss function λ, etc., were judiciously adjusted in the E-RSCT framework for refinement. However, achieving superior results still necessitates further extensive research and exploration.

5 Conclusion

Our present work offers a theoretical foundation for the observed empirical robustness of Poisson-encoded classifiers against adversarial attacks. Observing that randomized smoothing and Poisson coding exhibit similar characteristics, we first demonstrate the equivalence between the two, explaining why Poisson coding SNNs have adversarial robustness. Furthermore, our theoretical analysis shows that randomized smoothing is more stable than Poisson coding. On this basis we propose a novel randomized smoothing coding, which enhances the adversarial robustness of SNNs. Experimental results show that our method achieves state-of-the-art adversarial robustness. However, there is still room for further improvement in clean accuracy. Therefore, valuable future work directions include further improving the trade-off between adversarial robustness and clean accuracy. We believe our work will pave the way for further research on more applications of safety-critical SNNs.

Acknowledgments

This work was partially supported by the National Distinguished Young Scholars (62325603), the National Natural Science Foundation of China (62236009,U22A20103,62441606), the Beijing Natural Science Foundation for Distinguished Young Scholars (JQ21015), the China Postdoctoral Science Foundation (GZB20240824, 2024M753497), and the CAAI-MindSpore Open Fund, which was developed on the OpenI Community.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*. PMLR, 274–283.

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2018. Synthesizing robust adversarial examples. In *International Conference on Machine Learning*. PMLR, 284–293.

[3] Sander M Bohte, Joost N Kok, and Johannes A La Poutré. 2000. SpikeProp: backpropagation for networks of spiking neurons.. In *ESANN*, Vol. 48. Bruges, 419–424.

[4] Tong Bu, Jianhao Ding, Zecheng Hao, and Zhaofei Yu. 2023. Rate Gradient Approximation Attack Threats Deep Spiking Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7896–7906.

[5] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*. PMLR, 1310–1320.

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

[7] Lei Deng, Yujie Wu, Xing Hu, Ling Liang, Yufei Ding, Guoqi Li, Guangshe Zhao, Peng Li, and Yuan Xie. 2020. Rethinking the performance comparison between SNNS and ANNS. *Neural Networks* 121 (2020), 294–307.

[8] Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. 2022. Temporal Efficient Training of Spiking Neural Network via Gradient Re-weighting. In *International Conference on Learning Representations*. https://openreview.net/forum?id=_XNtisL32jv

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

[10] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. 2022. Event-Based Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 1 (2022), 154–180.

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

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*.

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[14] Alan L Hodgkin and Andrew F Huxley. 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* 117, 4 (1952), 500.

[15] Souvik Kundu, Massoud Pedram, and Peter A Beerel. 2021. 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*. 5209–5218.

[16] Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi Jaakkola. 2019. Tight certificates of adversarial robustness for randomly smoothed classifiers. *Advances in Neural Information Processing Systems* 32 (2019).

[17] Mikhail Leontev, Dmitry Antonov, and Sergey Sukhov. 2021. Robustness of spiking neural networks against adversarial attacks. In *2021 International Conference on Information Technology and Nanotechnology (ITNT)*. IEEE, 1–6.

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

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[20] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 6197 (2014), 668–673.

[21] Osamu Nomura, Yusuke Sakemi, Takeo Hosomi, and Takashi Morie. 2022. Robustness of spiking neural networks based on time-to-first-spike encoding against adversarial attacks. *IEEE Transactions on Circuits and Systems II: Express Briefs* 69, 9 (2022), 3640–3644.

[22] Jing Pei, Lei Deng, et al. 2019. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* 572, 7767 (2019), 106–111.

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

[24] Arjun Rao, Philipp Plank, Andreas Wild, and Wolfgang Maass. 2022. A long short-term memory for AI applications in spike-based neuromorphic hardware. *Nature Machine Intelligence* 4, 5 (2022), 467–479.

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

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

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

[28] Saima Sharmin, Nitin Rathi, Priyadarshini Panda, and Kaushik Roy. 2020. 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*. Springer, 399–414.

[29] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. 2018. Is Robustness the Cost of Accuracy?–A Comprehensive Study on the Robustness of 18 Deep Image Classification Models. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 631–648.

[30] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2018. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152* (2018).

[31] Rufin Van Rullen and Simon J Thorpe. 2001. Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural Computation* 13, 6 (2001), 1255–1283.

[32] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. 2018. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience* 12 (2018), 331.

[33] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. 2019. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 1311–1318.

[34] Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. 2020. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*. PMLR, 10693–10705.

[35] Man Yao, JiaKui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo XU, and Guoqi Li. 2024. 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*.

[36] Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. 2024. Spike-driven transformer. *Advances in Neural Information Processing Systems* 36 (2024).

[37] Man Yao, Ole Richter, Guangshe Zhao, Ning Qiao, Yannan Xing, Dingheng Wang, Tianxiang Hu, Wei Fang, Tugba Demirci, Michele De Marchi, Lei Deng, Tianyi Yan, Carsten Nielsen, Sadique Sheik, Chenxi Wu, Yonghong Tian, Bo Xu, and Guoqi Li. 2024. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature Communications* 15, 1 (25 May 2024), 4464.

[38] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. 2023. Attention Spiking Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 8 (2023), 9393–9410.

[39] Bojian Yin, Federico Corradi, and Sander M Bohté. 2021. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence* 3, 10 (2021), 905–913.

[40] Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. 2020. MACER: Attack-free and Scalable Robust Training via Maximizing Certified Radius. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJx1Na4Fwr

[41] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*. PMLR, 7472–7482.

[42] Roland S Zimmermann. 2019. Comment on" Adv-BNN: Improved Adversarial Defense through Robust Bayesian Neural Network". *arXiv preprint arXiv:1907.00895* (2019).