Appendix for *SNN-RAT: Robustness-enhanced Spiking Neural Network through Regularized Adversarial Training*

Theorem 1. *Given an L-layered SNN intended to inference T time-steps with θ as threshold, suppose that there are N_l neurons in layer l for $l = 1, 2, \cdots, L$. $\boldsymbol{W}^l \in \mathbb{R}^{N_l \times N_{l-1}}$. For layer l, it satisfies:*

$$D_2(\boldsymbol{S}^l, \widetilde{\boldsymbol{S}}^l)^2 \leq \frac{1}{\theta^2}{\Lambda^l}^2 D_2(\boldsymbol{S}^{l-1}, \widetilde{\boldsymbol{S}}^{l-1})^2 + \Gamma^l, \tag{A1}$$

where Λ^l is a Lipschitz constant and Γ^l is a constant for layer l, which can be expressed as:

$$\Lambda^l = \sup_{\boldsymbol{s}\neq 0, \boldsymbol{s}\in\psi^{N_{l-1}}} \frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2}, \tag{A2}$$

$$\Gamma^l = \frac{N_l T(T+1)}{\lambda}\left[\frac{\gamma^l}{\theta} + \left(\frac{\gamma^l}{\theta}\right)^2\right], \tag{A3}$$

where $\gamma^l = \sup_{\boldsymbol{s}\neq 0, \boldsymbol{s}\in\chi^{N_{l-1}}} \|\boldsymbol{W}^l\boldsymbol{s}\|_\infty + \sup_{\boldsymbol{s}\neq 0, \boldsymbol{s}\in\chi^{N_{l-1}}} \| - \boldsymbol{W}^l\boldsymbol{s}\|_\infty$. $\chi = \{0, 1\}, \psi = \{-1, 0, 1\}$.

Proposition 1. *Given a weight matrix with real values $\boldsymbol{W}^l \in \mathbb{R}^{N_l \times N_{l-1}}$. $\psi = \{-1, 0, 1\}$. It satisfies:*

$$\Lambda^l = \sup_{\boldsymbol{s}\neq 0, \boldsymbol{s}\in\psi^{N_{l-1}}} \frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2} \leq \|\boldsymbol{W}^l\|_2 = \sigma^{\max}(\boldsymbol{W}^l), \tag{A4}$$

where $\|\boldsymbol{W}^l\|_2$ is the induced l_2 matrix norm, and $\sigma^{\max}(\boldsymbol{W}^l)$ is the largest singular value of $\boldsymbol{W}^l$.

Proposition 2. *Given a weight matrix with real values $\boldsymbol{W}^l \in \mathbb{R}^{N_l \times N_{l-1}}$. $\chi = \{0, 1\}$. It satisfies:*

$$\gamma^l = \sup_{\boldsymbol{s}\neq 0, \boldsymbol{s}\in\chi^{N_{l-1}}} \|\boldsymbol{W}^l\boldsymbol{s}\|_\infty + \sup_{\boldsymbol{s}\neq 0, \boldsymbol{s}\in\chi^{N_{l-1}}} \| - \boldsymbol{W}^l\boldsymbol{s}\|_\infty \leq 2\|\boldsymbol{W}^l\|_\infty \leq 2\sqrt{N_{l-1}}\|\boldsymbol{W}^l\|_2, \tag{A5}$$

where $\|\boldsymbol{W}^l\|_p$ is the induced l_p matrix norm.

A Proofs

Proof for Theorem 1

Proof. For neurons following dynamics in Eq. 1-3 in the main text, Eq. 3 can be rewritten as:

$$\boldsymbol{m}^l(t) = \lambda(\boldsymbol{m}^l(t^-) - \boldsymbol{s}^l(t)\boldsymbol{r}^l(t)), \tag{A6}$$

where $\boldsymbol{r}^l(t)$ represents the subtracted value by reset. When $\boldsymbol{s}^l(t) = 1$, $\boldsymbol{r}^l(t)) = \boldsymbol{m}^l(t) \geq \theta$; otherwise, $\boldsymbol{r}^l(t)) = 0$. Thus, we have:

$$\boldsymbol{r}^l(t)\boldsymbol{s}^l(t) \geq \theta\boldsymbol{s}^l(t). \tag{A7}$$

By stacking Eq. 1 and Eq. A6, we have for $t = 2, \cdots, T$:

$$\begin{aligned} \boldsymbol{m}^l(t) - \lambda\boldsymbol{m}^l(t-1) &= \lambda(\boldsymbol{W}^l\boldsymbol{s}^{l-1}(t) - \boldsymbol{s}^l(t)\boldsymbol{r}^l(t)) \\ \lambda\boldsymbol{m}^l(t-1) - \lambda^2\boldsymbol{m}^l(t-2) &= \lambda^2(\boldsymbol{W}^l\boldsymbol{s}^{l-1}(t-1) - \boldsymbol{s}^l(t-1)\boldsymbol{r}^l(t-1)) \end{aligned} \tag{A8}$$

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Notice that the left sides of Eq. A8 can iteratively eliminate $\boldsymbol{m}^l(t)$ in the T formulas. Usually the initial membrane potential $\boldsymbol{m}^l(0)$ is set to zero. Thus, we can obtain:

$$\boldsymbol{m}^l(T) = \boldsymbol{W}^l \sum_{i=1}^{T} \lambda^{T+1-i} \boldsymbol{s}^{l-1}(i) - \sum_{i=1}^{T} \lambda^{T+1-i} \boldsymbol{s}^l(i) \boldsymbol{r}^l(i) \tag{A9}$$

To simplify annotation, $\sum_{i=1}^{t} \lambda^{t+1-i} \boldsymbol{s}^l(i)$ is later denoted as $\boldsymbol{h}^l(t)$. Thus, $\boldsymbol{s}^l(t) = \frac{\boldsymbol{h}^l(t) - \boldsymbol{h}^l(t-1)}{\lambda}$. The weighted current $\boldsymbol{W}^l \boldsymbol{s}^{l-1}(t)$ is naturally bounded by norms induced by l_∞ norms. The upper bound of currents in layer l is determined by ($\chi = \{0, 1\}$):

$$\eta^l = \sup_{\boldsymbol{s} \neq 0, \boldsymbol{s} \in \chi^{N_{l-1}}} \|\boldsymbol{W}^l \boldsymbol{s}\|_\infty \geq 0. \tag{A10}$$

Similarly, the lower bound is determined by:

$$\mu^l = \inf_{\boldsymbol{s} \neq 0, \boldsymbol{s} \in \chi^{N_{l-1}}} \|\boldsymbol{W}^l \boldsymbol{s}\|_\infty. \tag{A11}$$

For common cases where $\boldsymbol{W}^l$ includes items of negative values, μ^l is negative and can be reformulated as:

$$\mu^l = -1 \cdot \sup_{\boldsymbol{s} \neq 0, \boldsymbol{s} \in \chi^{N_{l-1}}} \| - \boldsymbol{W}^l \boldsymbol{s}\|_\infty \leq 0. \tag{A12}$$

At timestep t, the reset mechanism of spike constrains $\boldsymbol{m}^l(t)$ to be less than $\eta^l t$. Consider some neuron receiving negative inputs all the time, $\boldsymbol{m}^l(t)$ should be no less than $\mu^l t$. Hence,

$$\begin{aligned} \mu^l t &\leq \boldsymbol{W}^l \boldsymbol{h}^{l-1}(t) - \sum_{i=1}^{t} \lambda^{t+1-i} \boldsymbol{s}^l(i) \boldsymbol{r}^l(i) \\ &\leq \boldsymbol{W}^l \boldsymbol{h}^{l-1}(t) - \theta \boldsymbol{h}^l(t) \leq \eta^l t. \end{aligned} \tag{A13}$$

Thus, we have:

$$\frac{\boldsymbol{W}^l \boldsymbol{h}^{l-1}(t) - \eta^l t}{\theta} \leq \boldsymbol{h}^l(t) \leq \frac{\boldsymbol{W}^l \boldsymbol{h}^{l-1}(t) - \mu^l t}{\theta}. \tag{A14}$$

So we can obtain:

$$\begin{aligned} \boldsymbol{s}^l(t) - \widetilde{\boldsymbol{s}}^l(t) &= \frac{1}{\lambda} \left[\boldsymbol{h}^l(t) - \boldsymbol{h}^l(t-1) - \widetilde{\boldsymbol{x}}^l(t) + \widetilde{\boldsymbol{x}}^l(t-1) \right] \\ &\leq 2t \frac{\eta^l - \mu^l}{\lambda \theta} + \frac{1}{\theta} \boldsymbol{W}^l (\boldsymbol{s}^{l-1}(t) - \widetilde{\boldsymbol{s}}^{l-1}(t)). \end{aligned} \tag{A15}$$

Meanwhile, we have:

$$\boldsymbol{s}^l(t) - \widetilde{\boldsymbol{s}}^l(t) \leq 1, \tag{A16}$$

and

$$\frac{1}{\theta} \boldsymbol{W}^l (\boldsymbol{s}^{l-1}(t) - \widetilde{\boldsymbol{s}}^{l-1}(t)) \leq \frac{\eta^l - \mu^l}{\theta}. \tag{A17}$$

Thus,

$$\boldsymbol{s}^l(t) - \widetilde{\boldsymbol{s}}^l(t) + \frac{1}{\theta} \boldsymbol{W}^l (\boldsymbol{s}^{l-1}(t) - \widetilde{\boldsymbol{s}}^{l-1}(t)) \leq 1 + \frac{\eta^l - \mu^l}{\theta}. \tag{A18}$$

Combining Eq. A18 and Eq. A15, we obtain:

$$|\boldsymbol{s}^l(t) - \widetilde{\boldsymbol{s}}^l(t)|^2 - \frac{1}{\theta^2} |\boldsymbol{W}^l (\boldsymbol{s}^{l-1}(t) - \widetilde{\boldsymbol{s}}^{l-1}(t))|^2 \leq \frac{2t}{\lambda} \left[\frac{\eta^l - \mu^l}{\theta} + \left(\frac{\eta^l - \mu^l}{\theta} \right)^2 \right]. \tag{A19}$$

For simplicity, we denote $\eta^l - \mu^l$ as γ^l. The calculation of $\|\boldsymbol{s}^l(t) - \widetilde{\boldsymbol{s}}^l(t)\|_2^2$ should add the N_l nodes up, which gives:

$$\begin{aligned} \|\boldsymbol{s}^l(t) - \widetilde{\boldsymbol{s}}^l(t)\|_2^2 &= \sum_{i=1}^{N_l} |s_i^l(t) - \widetilde{s}_i^l(t)|^2 \\ &\leq \frac{1}{\theta^2} \|\boldsymbol{W}^l (\boldsymbol{s}^{l-1}(t) - \widetilde{\boldsymbol{s}}^{l-1}(t))\|_2^2 + \frac{2N_l t}{\lambda} \left[\frac{\gamma^l}{\theta} + \left(\frac{\gamma^l}{\theta} \right)^2 \right]. \end{aligned} \tag{A20}$$

The l_2 norm of weighted currents from previous layer $l-1$ satisfies the following inequality:

$$\|\boldsymbol{W}^l(\boldsymbol{s}^{l-1}(t)-\widetilde{\boldsymbol{s}}^{l-1}(t))\|_2^2 \leq {\Lambda^l}^2\|\boldsymbol{s}^{l-1}(t)-\widetilde{\boldsymbol{s}}^{l-1}(t)\|_2^2. \tag{A21}$$

where Λ^l is the matrix norm induced by l_2 norm ($\psi=\{-1,0,1\}$):

$$\Lambda^l = \sup_{\boldsymbol{s}\neq 0, \boldsymbol{s}\in\psi^{N_{l-1}}} \frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2}. \tag{A22}$$

Hence, the l_2 perturbation distance of spike train satisfies:

$$\begin{aligned} D_2(\boldsymbol{S}^l,\widetilde{\boldsymbol{S}}^l)^2 &= \|\boldsymbol{S}^l-\widetilde{\boldsymbol{S}}^l\|_2^2 \\ &= \sum_{t=1}^{T}\|\boldsymbol{s}^l(t)-\widetilde{\boldsymbol{s}}^l(t)\|_2^2 \\ &\leq \Gamma^l + \frac{1}{\theta^2}{\Lambda^l}^2 D_2(\boldsymbol{S}^{l-1},\widetilde{\boldsymbol{S}}^{l-1})^2. \end{aligned} \tag{A23}$$

where

$$\Gamma^l = \frac{N_l T(T+1)}{\lambda}\left[\frac{\eta^l-\mu^l}{\theta}+\left(\frac{\eta^l-\mu^l}{\theta}\right)^2\right]. \tag{A24}$$

□

Proof for Proposition 1

Proof. Consider the induced matrix norm $\sup \frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2}$ from a discrete space $\boldsymbol{s}\neq 0, \boldsymbol{s}\in\psi^{N_{l-1}}$ where $\psi=\{-1,0,1\}$, $\boldsymbol{s}\neq 0$ constrain the vector $\boldsymbol{s}$ not to be an all-zero vector. Now, let us examine the set $\{s|\boldsymbol{s}\neq 0,\boldsymbol{s}\in\psi^{N_{l-1}}\}$. Each element of $\boldsymbol{s}$ can get the maximum absolute value 1, which means that the set is in a hypersphere $\|\boldsymbol{s}\|_2^2\leq N_{l-1}$. The expansion of domain leads to an possible increase of supremum:

$$\sup_{\boldsymbol{s}\neq 0,\boldsymbol{s}\in\psi^{N_{l-1}}}\frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2} \leq \sup_{\|\boldsymbol{s}\|_2^2\leq N_{l-1}}\frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2}. \tag{A25}$$

We rewrite the supremum over $\|\boldsymbol{s}\|_2^2\leq N_{l-1}$ to the maximum of the traverse of the radius r.

$$\sup_{\|\boldsymbol{s}\|_2^2\leq N_{l-1}}\frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2} = \max_{r\leq\sqrt{N_{l-1}}}\sup_{\|\boldsymbol{s}\|_2=r}\frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2}. \tag{A26}$$

By substituting $\hat{\boldsymbol{s}}=\boldsymbol{s}/r$ into the right hand side of Eq. A26, we have:

$$\begin{aligned} \max_{r\leq\sqrt{N_{l-1}}}\sup_{\|\boldsymbol{s}\|_2=r}\frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2} &= \max_{r\leq\sqrt{N_{l-1}}}\sup_{\|\hat{\boldsymbol{s}}\|_2=1}\frac{r\|\boldsymbol{W}^l\hat{\boldsymbol{s}}\|_2}{r\|\hat{\boldsymbol{s}}\|_2} \\ &= \max_{r\leq\sqrt{N_{l-1}}}\sup_{\|\boldsymbol{s}\|_2=1}\frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2} \\ &= \sup_{\|\boldsymbol{s}\|_2=1}\|\boldsymbol{W}^l\boldsymbol{s}\|_2. \end{aligned} \tag{A27}$$

According to theory of matrix norms [Malek-Shahmirzadi, 1983], $\sup_{\|\boldsymbol{s}\|_2=1}\|\boldsymbol{W}^l\boldsymbol{s}\|_2$ is actually the spectral norm of $\boldsymbol{W}^l$ and meanwhile the largest singular value of $\boldsymbol{W}^l$: $\sigma^{\max}(\boldsymbol{W}^l)$. Combining Eq. A25,A26,A27, we have:

$$\sup_{\boldsymbol{s}\neq 0,\boldsymbol{s}\in\psi^{N_{l-1}}}\frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_2}{\|\boldsymbol{s}\|_2} \leq \sup_{\|\boldsymbol{s}\|_2=1}\|\boldsymbol{W}^l\boldsymbol{s}\|_2 = \sigma^{\max}(\boldsymbol{W}^l). \tag{A28}$$

□

Proof for Proposition 2

Proof. The set $\{s|\boldsymbol{s} \neq 0, \boldsymbol{s} \in \chi^{N_{l-1}}, \chi = \{0,1\}\}$ is a subset of $\|\boldsymbol{s}\|_\infty = 1$. Thus,

$$\sup_{\boldsymbol{s}\neq 0, \boldsymbol{s}\in\chi^{N_{l-1}}} \|\boldsymbol{W}^l\boldsymbol{s}\|_\infty \leq \sup_{\|\boldsymbol{s}\|_\infty=1} \|\boldsymbol{W}^l\boldsymbol{s}\|_\infty = \sup_{\|\boldsymbol{s}\|_\infty=1} \frac{\|\boldsymbol{W}^l\boldsymbol{s}\|_\infty}{\|\boldsymbol{s}\|_\infty} = \|\boldsymbol{W}^l\|_\infty, \tag{A29}$$

where $\|\boldsymbol{W}^l\|_\infty$ is the induced l_∞ matrix norm of $\boldsymbol{W}^l$. Similarly, we also have:

$$\sup_{\boldsymbol{s}\neq 0, \boldsymbol{s}\in\chi^{N_{l-1}}} \|-\boldsymbol{W}^l\boldsymbol{s}\|_\infty \leq \sup_{\|\boldsymbol{s}\|_\infty=1} \|\boldsymbol{W}^l\boldsymbol{s}\|_\infty = \|\boldsymbol{W}^l\|_\infty. \tag{A30}$$

Based on the theory of norm equivalence, the following inequality holds: [Golub and Van Loan, 2013]:

$$\|\boldsymbol{W}^l\|_\infty \leq \sqrt{N_{l-1}}\|\boldsymbol{W}^l\|_2. \tag{A31}$$

Thus, the following statement is satisfied:

$$\sup_{\boldsymbol{s}\neq 0, \boldsymbol{s}\in\chi^{N_{l-1}}} \|\boldsymbol{W}^l\boldsymbol{s}\|_\infty + \sup_{\boldsymbol{s}\neq 0, \boldsymbol{s}\in\chi^{N_{l-1}}} \|-\boldsymbol{W}^l\boldsymbol{s}\|_\infty \leq 2\|\boldsymbol{W}^l\|_\infty \leq 2\sqrt{N_{l-1}}\|\boldsymbol{W}^l\|_2. \tag{A32}$$

□

B Details of Implementation

The architectures used to illustrate the vulnerability in Sec. 3 of the main text are VGG-11[1] and WideResNet-16[2] with inference timestep T=8. The leak factor λ is set to 1.0 in SNN. They are trained on the CIFAR-10 and CIFAR-100 datasets [Krizhevsky et al., 2009]. In Sec. 5 of the main text, the validation of our method is also based on the aforementioned two architectures. The training process lasts for 200 epochs. Batch normalization are used in the network to overcome the gradient vanishing or explosion for deep SNNs as suggested by Zheng et al. [2021]. In the training process, stochastic gradient descent are deployed, and the initial learning rate is set to 0.1. The learning rate uses a cosine annealing schedule with $T_{\max}$ equaling the max number of epochs. The vanilla models are trained without a regularized adversarial training scheme, so a decay of weights of 5e-4 is added to each training iteration to improve the overall accuracy. The image data is first normalized by the means and variances of the three channels and then fed into SNNs to trigger spikes. All the experiments are conducted on the PyTorch platform [Paszke et al., 2019] on NVIDIA GeForce RTX 3090.

To overcome the non-differentiable problem and enable SNN training, the surrogate gradient function is applied to supply gradients for training and BPTT attack.

$$\frac{\partial \boldsymbol{s}^l(t)}{\partial \boldsymbol{m}^l(t^-)} = \frac{1}{\kappa^2}\max\left(\kappa - \left|\boldsymbol{m}^l(t^-)\right|, 0\right) \tag{A33}$$

In our implementation, we choose $\kappa = 1.0$ for all the experiments. Note that we adapt and modify the implementation of gradient-based attacks of the torchattacks Python package [Kim, 2020] as we need to perform successful attacks on SNNs.

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

We compare our methods with the state-of-the-art models and report the results in Tab. A1. One can find that our proposed training scheme outperforms the others in terms of both clean accuracy and perturbed accuracy. The evaluation is based on the VGG-11 experiments on the CIFAR-100 dataset. The noise budget has been fixed to $\epsilon = 8/255$ for FGSM and $\alpha = 0.01, step = 7$ for PGD. The attack is based on the surrogate gradient produced by BPTT. The performance of accuracy attacked by FGSM is 25.86% for our work, higher than that proposed by Sharmin et al. [2020] (15.5%) and Kundu et al. [2021] (22.0%). Apart from that, our clean accuracy (70.89%) is higher than that proposed by Sharmin et al. [2020] (64.4%) and Kundu et al. [2021] (65.1%). This implies that our proposed methods can bring better generalization compared to the SOTA robust models.

[1]The VGG-11 used follows the implementation in https://github.com/nitin-rathi/hybrid-snn-conversion.

[2]The WideResNet-16 used follows the implementation in https://github.com/xternalz/WideResNet-pytorch. (MIT license)

Table A1: Comapare with state-of-the-art work on adversarial robustness of SNN.

	Our RAT scheme	Sharmin et al. [2020]	Kundu et al. [2021]	Vanilla training
FGSM	25.86	15.5	22.0	5.30
PGD	10.38	6.3	7.5	0.02
Clean	70.89	64.4	65.1	73.33
Additional Cost	Regularized Training	-	-	-

Table A2: Checklist for characteristic behaviors caused by obfuscated and masked gradients.

Items to identify gradient obfuscation	CBA	BPTR	BPTT
(1) Single-step attack performs better compared to iterative attacks	Fail	Pass	Pass
(2) Black-box attacks performs better compared to white-box attacks	NA	Pass	Pass
(3) Increasing perturbation bound can't increase attack strength	NA	Pass	Pass
(4) Unbounded attacks can't reach $\sim 100\%$ success	NA	Pass	Pass
(5) Adversarial example can be found through random sampling	NA	Pass	Pass

It is worth noting that although our training algorithm improves the robustness of SNNs, it comes at a cost compared to the work of Sharmin et al. [2020] and Kundu et al. [2021]. The cost is mainly reflected in the training time. First, our training includes time to generate adversarial noise. Adversarial learning is a common scheme to improve robustness, and generating adversarial examples using only BPTT differentiable approximation in SNN is a time-consuming operation. Our algorithm mitigates the increase in training time by mixing in a faster yet efficient BPTR approximation. In addition, the orthogonal regularization of the weights is computed every update, which also increases the training time. Solutions to reduce the time consumption of regularization include sampling fewer weights for regularizing, or reducing the number of regularization updates.

D Analysis of Gradient Obfuscation

We design and summarize three differentiable approximations, i.e. CBA, BPTT, BPTR, which can be deployed in gradient-based attacks to show the vulnerability of SNNs. The main concern of the gradient obfuscation lies in the inaccurate of updating gradients. In particular, the performance of the three differentiable approximations was checked against the five tests that can identify gradient obfuscation as done in Kundu et al. [2021]. Our analysis is mainly based on the quantification results in Tab. 1 and Tab. 2 in the main text. Also, this will explain the reason why we choose BPTT and BPTR in the procedure of the mixed training.

As shown in Tab. 1, for all the trials, the performance of single-step FGSM is worse than its iterative counterpart PGD except for that of the WRN-16 experiment for CIFAR-100 (Attacked Accuracy: FGSM 37.68% v.s. PGD 43.87%). Thus, the CBA approximation has the potential not to provide powerful enough attacks.

Hence, the rest of the analysis is about BPTT and BPTR. The results in Tab. 1 and Tab. 2 certify the success of BPTT and BPTR approximation in terms of Test(1) in Table R2. To verify Test(2) we conduct black-box attacks on the proposed models and the vanilla ones. The black-box perturbation performs weaker in Table 2, and Test(2) is satisfied. To verify Test(3)(4) we analyze VGG-11 on CIFAR-10 with increasing attack bound. In Fig. A1, the classification accuracy decreases as we increase ϵ and finally reaches an accuracy of random guessing. As suggested in Kundu et al. [2021], Test(5) "can fail only if gradient-based attacks cannot provide adversarial examples for the model to misclassify". To sum up, we found no gradient obfuscation for the BPTT and BPTR approximation, which are suitable for adversarial training and testing.

E Analysis of Computational Cost

The additional computational cost of SNN adversarial training is mainly reflected in the choice of gradient approximation. Here we evaluate the computational time of adversarial testing. The adversarial testing means that model should forward twice and backward once. During the tests, we

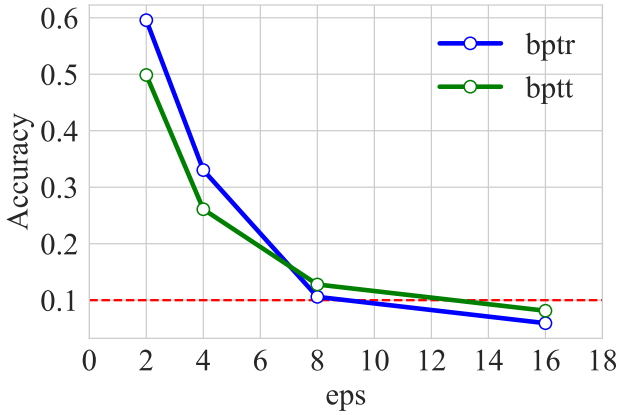

Figure A1: Performance of untargeted FGSM attacks with different approximation. The red dashed line represents the accuracy of random guess (10% for CIFAR-10).

Table A3: Ablation study and time cost for mixed adversarial training

	Mixed AT	BPTT AT	BPTR AT
FGSM	45.23	48.93	22.0
RFGSM	64.61	68.15	7.5
Attack Cost (sec. per epoch)	25.48	38.09	13.23

fix the mini-batch size to 64 and run the test on a NVIDIA 3090 GPU. The results are presented in Tab. A2. We find that among the three proposed approximations, CBA is the most efficient as it only propagates through spiking neurons as if there is only ReLU activation. BPTR is almost as efficient as CBA. Considering that BPTR is more powerful than CBA, BPTR is a fairly good attack for SNN. Whereas, BPTT costs nearly 3 $\times$ of what CBA takes to complete testing.

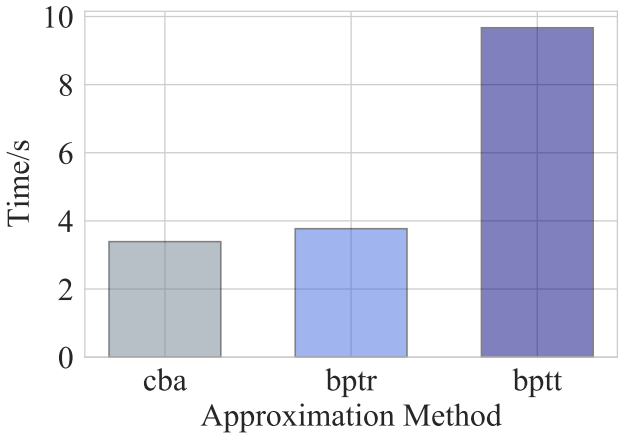

Figure A2: Computational time of adversarial testing.

F Societal Impact and Limitations

As our work is about evaluating and strengthening the robustness of SNN, there is no apparent negative social impact. Our proposed method improves adversarial robustness, which has a far more positive social impact. As regards limitations, our method may lose robustness when encountering unseen adversarial attacks. As a result, adversarial training may require more types of attacks.

References

Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020.

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*, pages 5209–5218, 2021.

Massoud Malek-Shahmirzadi. A characterization of certain classes of matrix norms. *Linear and Multilinear Algebra*, 13(2):97–99, 1983.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 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 *European Conference on Computer Vision*, pages 399–414. Springer, 2020.

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, pages 11062–11070, 2021.