Boosting the Robustness-Accuracy Trade-off of SNNs by Robust Temporal Self-Ensemble

Jihang Wang[*1,2], **Dongcheng Zhao**[*3,4], **Ruolin Chen**[1,2], **Qian Zhang**[1,2,3,4†], **Yi Zeng**[1,2,3,4†]

[1]Brain-inspired Cognitive AI Lab, Institute of Automation, Chinese Academy of Sciences, Beijing, China.
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China.
[3]Beijing Institute of AI Safety and Governance, Beijing, China.
[4]Beijing Key Laboratory of Safe Al and Superalignment, Beijing, China.
{wangjihang2021, chenruolin2024, q.zhang, yi.zeng}@ia.ac.cn, dongcheng.zhao@beijing-aisi.ac.cn

Abstract

Spiking Neural Networks (SNNs) offer a promising direction for energy-efficient and brain-inspired computing, yet their vulnerability to adversarial perturbations remains poorly understood. In this work, we revisit the adversarial robustness of SNNs through the lens of temporal ensembling, treating the network as a collection of evolving sub-networks across discrete timesteps. This formulation uncovers two critical but underexplored challenges—the fragility of individual temporal sub-networks and the tendency for adversarial vulnerabilities to transfer across time. To overcome these limitations, we propose Robust Temporal self-Ensemble (RTE), a training framework that improves the robustness of each sub-network while reducing the temporal transferability of adversarial perturbations. RTE integrates both objectives into a unified loss and employs a stochastic sampling strategy for efficient optimization. Extensive experiments across multiple benchmarks demonstrate that RTE consistently outperforms existing training methods in robust-accuracy trade-off. Additional analyses reveal that RTE reshapes the internal robustness landscape of SNNs, leading to more resilient and temporally diversified decision boundaries. Our study highlights the importance of temporal structure in adversarial learning and offers a principled foundation for building robust spiking models.

Introduction

Deep Neural Networks (DNNs) have achieved remarkable success in processing high-dimensional data and delivering state-of-the-art performance across a wide range of tasks (Goodfellow et al. 2016; He et al. 2016). These models typically rely on continuous-valued activations for inter-layer communication. While effective, this design departs significantly from the signaling mechanisms of biological neural systems and incurs high energy costs, particularly in large-scale or resource-constrained settings.

To address this gap, Spiking Neural Networks (SNNs) have emerged as a brain-inspired computational paradigm that more faithfully captures the temporal dynamics and event-driven nature of biological neurons (Maass 1997; Shen et al. 2023). In contrast to conventional DNNs, SNNs transmit information via discrete spike trains distributed across space and time. Their asynchronous, event-triggered processing and multiplication-free synaptic operations enable substantially higher energy efficiency, particularly when deployed on neuromorphic hardware designed to exploit their sparse, binary signaling patterns (Pei et al. 2019; Roy, Jaiswal, and Panda 2019; Li et al. 2023).

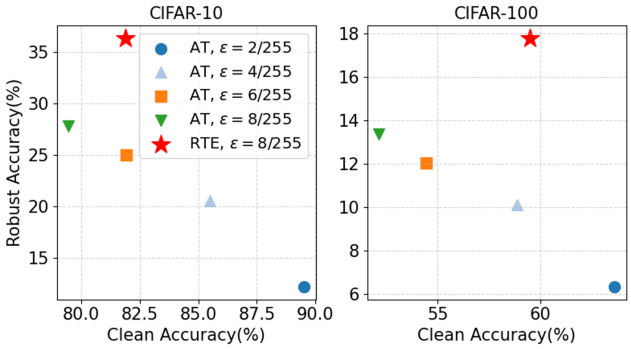

Figure 1: Comparison of clean and robust accuracy under AutoPGD attacks for AT and RTE.

Despite these advantages, SNNs have been shown to be similarly vulnerable to adversarial perturbations as their DNN counterparts (Bu et al. 2023; Hao et al. 2023; Liang et al. 2021; Lun et al. 2025). This susceptibility highlights a critical gap between current SNN models and the robustness of biological neural systems, limiting their reliability in safety-critical applications. To improve adversarial robustness of SNNs, most existing work has focused on adversarial training (AT) with small perturbation budgets (Ding et al. 2022, 2024b,a; Liu et al. 2024), typically $\varepsilon = 2/255$ or $4/255$ on standard benchmarks such as CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009). Although AT has been widely adopted in both artificial and spiking neural networks (Madry et al. 2018), small-budget perturbations often fail to sufficiently explore the neighborhood of training examples, resulting in limited robustness.

A straightforward remedy is to increase the perturbation budget ε during training. Empirical results show that this

[*]These authors contributed equally.
[†]Corresponding author.

can improve robust accuracy, but often at the expense of a substantial drop in clean accuracy, as shown in Figure 1. This trade-off becomes especially severe on more challenging datasets such as CIFAR-100, where stronger perturbations degrade generalization on clean inputs despite robustness gains. These observations reveal a fundamental limitation of current AT strategies for SNNs—simply increasing attack strength is insufficient to achieve a favorable robustness–accuracy balance.

To improve the robustness-accuracy trade-off, we present a novel training framework grounded in a temporal ensemble interpretation of SNNs. Specifically, we formalize the output of a spiking network as an implicit ensemble of temporal sub-networks, each corresponding to neural activity at a specific timestep (Ding et al. 2025; Zhao et al. 2025). This formulation reveals a natural inductive structure within SNNs, wherein predictions emerge from a composition of temporally distributed decision paths. Leveraging this insight, we introduce **Robust Temporal self-Ensemble (RTE)**, a training methodology that jointly (i) improves the adversarial robustness of individual temporal sub-networks and (ii) suppresses the transferability of vulnerabilities across timesteps. RTE achieves this via an adjustable regularization scheme that operates across temporal components, effectively mitigating temporal vulnerabilities to adversarial directions. Our main contributions are summarized as follows:

- We formulate the output of an SNN as a temporal self-ensemble, and propose the RTE framework to simultaneously improve the robustness of individual sub-networks and suppress vulnerability transfer across timesteps.
- RTE introduces an adjustable regularization mechanism that enables SNNs to explore broader local neighborhoods in the input space while maintaining high clean accuracy with minimal compromise.
- Extensive evaluations on standard benchmarks demonstrate that RTE is a scalable and effective defense strategy. It consistently achieves a better robustness–accuracy trade-off, particularly on more complex datasets like CIFAR-100.

Related Work

Adversarial Defense in Spiking Neural Networks

Efforts to improve the adversarial robustness of SNNs fall into three broad categories: regularization-based training, stochastic modeling, and input encoding strategies. Regularization-based methods introduce additional loss terms or impose constraints to stabilize model behavior under perturbations. For instance, Regularized Adversarial Training (RAT) applies parseval regularization to bound the Lipschitz constant (Ding et al. 2022), Membrane Potential Perturbation Dynamics (MPPD) leverages neuron dynamics to stabilize SNNs (Ding et al. 2024a), and Sparsity Regularization (SR) uses sparsity-aware smoothing techniques (Liu et al. 2024). Stochastic modeling injects randomness into network dynamics to improve robustness. Examples include randomized smoothing for rate-coded SNNs (Mukhoty et al. 2024), stochastic synaptic gating (Ding et al. 2024b), and heterogeneity in time constants (Wang et al. 2025). While effective in certain settings, these methods may introduce significant inference variance. Encoding-based strategies such as signed rate encoding (Mukhoty, AlQuabeh, and Gu 2025) and frequency-domain transformations (Xu et al. 2024) aim to distribute or normalize input information. Nevertheless, they primarily address input-level perturbations and do not provide internal robustness guarantees.

Ensemble-Based Robustness and Temporal Modeling in SNNs

Ensemble-based adversarial defenses have shown promise in improving robustness by promoting prediction diversity and reducing cross-model vulnerability transfer (Pang et al. 2019; Yang et al. 2020, 2021; Deng and Mu 2023). For example, TRS maximizes gradient disagreement across ensemble members (Yang et al. 2021), and DVERGE explicitly minimizes overlap in adversarial subspaces (Yang et al. 2020). Recently, several works have extended ensemble concepts to temporal modeling in SNNs, interpreting network outputs as implicit ensembles across timesteps (Ding et al. 2025; Zhao et al. 2025). These methods typically focus on enforcing temporal consistency or stability. However, they often overlook inter-timestep diversity and fail to explicitly suppress temporal vulnerability transfer—an issue particularly critical under ensemble-based robustness. In contrast, our proposed RTE framework builds on a principled temporal ensemble view of SNNs. RTE directly targets the robustness of individual temporal sub-networks and introduces regularization to decouple adversarial influence across time. This formulation enables scalable and architecture-agnostic adversarial defense, achieving superior robustness–accuracy trade-offs compared to prior AT-based and ensemble-inspired methods.

Notations and Preliminaries

Adversarial Attacks

We consider untargeted adversarial attacks constrained by an ℓ_p-norm ball. Let $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^C$ denote a multi-class classifier, and let $\boldsymbol{x} \in \mathbb{R}^d$ be an input sample with ground-truth label y. Given a perturbation budget $\varepsilon > 0$, an adversarial example $\boldsymbol{x'}$ is defined as the solution to the following optimization problem:

$$\boldsymbol{x'} = \arg \max_{\|\boldsymbol{x'} - \boldsymbol{x}\|_p \leq \varepsilon} \mathcal{L}(\boldsymbol{f}(\boldsymbol{x'}), y) \tag{1}$$

where $\mathcal{L}$ denotes the cross-entropy (CE) loss. Standard attack algorithms include FGSM (Goodfellow, Shlens, and Szegedy 2015), PGD (Madry et al. 2018), AutoPGD and AutoAttack (Croce and Hein 2020).

Spiking Neural Networks

Spiking Neural Networks process information over discrete timesteps $t = 1, \ldots, T$ using binary spike trains. Each layer l produces a spike output $\boldsymbol{s}^l(t) \in \{0, 1\}$ governed by the

Leaky Integrate-and-Fire (LIF) neuron model:

$$\boldsymbol{V}^l(t+1) = \lambda \boldsymbol{V}^l(t) \odot (1 - \boldsymbol{s}^l(t)) + (1-\lambda)\boldsymbol{W}^l \boldsymbol{s}^{l-1}(t+1) \tag{2}$$

$$\boldsymbol{s}^l(t+1) = H(\boldsymbol{V}^l(t+1) - V_{\text{th}}) \tag{3}$$

where $\boldsymbol{V}^l(t)$ denotes the membrane potential, λ is the leak rate, $\boldsymbol{W}^l$ is the synaptic weight matrix, and $H(\cdot)$ is the Heaviside step function, acting as a binary thresholding mechanism.

For static inputs such as images, direct encoding is commonly employed, where the same input $\boldsymbol{x}$ is injected into the network at every timestep. The final prediction is computed by aggregating output activations over T timesteps:

$$\boldsymbol{f}(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{W}^L \boldsymbol{s}^{L-1}(t) \tag{4}$$

Due to the non-differentiability of $H(\cdot)$, training SNNs typically relies on surrogate gradient techniques, which provide differentiable approximations to $H(\cdot)$ and allow optimization via backpropagation through time. We provide the specific surrogate gradient functions in Appendix A.

Methods

The output of an SNN, as defined in Eq. 4, can be reformulated as a temporal average over per-timestep predictions:

$$\boldsymbol{f}(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{f}_t(\boldsymbol{x}), \quad \text{where} \quad \boldsymbol{f}_t(\boldsymbol{x}) = \boldsymbol{W}^L \boldsymbol{s}^{L-1}(t) \tag{5}$$

This formulation induces a natural interpretation of SNNs as temporal ensembles, where each $\boldsymbol{f}_t(\boldsymbol{x})$ corresponds to the network's prediction at timestep t. We define each $\boldsymbol{f}_t(\boldsymbol{x})$ as the output of a distinct *temporal sub-network*.

Leveraging this temporal ensemble structure, we introduce an adversarial training framework that jointly optimizes two complementary objectives: (i) reducing the individual adversarial vulnerability of each temporal sub-network, and (ii) minimizing the transferability of adversarial perturbations across timesteps.

Sub-Network Vulnerability Minimization

Most existing approaches evaluate adversarial robustness solely based on the aggregated SNN output $\boldsymbol{f}(\boldsymbol{x})$, without explicitly accounting for vulnerabilities at individual timesteps. If we adopt cross-entropy (CE) loss in Eq. 1, we obtain the following upper bound:

$$\begin{aligned}
&\max_{\|\boldsymbol{x}'-\boldsymbol{x}\|_p \le \varepsilon} \mathcal{L}_{\text{CE}}(\boldsymbol{f}(\boldsymbol{x}'), y) \\
&\le \max_{\|\boldsymbol{x}'-\boldsymbol{x}\|_p \le \varepsilon} \sum_{t=1}^{T} \frac{\mathcal{L}_{\text{CE}}(\boldsymbol{f}_t(\boldsymbol{x}'), y)}{T} \\
&\le \sum_{t=1}^{T} \max_{\|\boldsymbol{x}'-\boldsymbol{x}\|_p \le \varepsilon} \frac{\mathcal{L}_{\text{CE}}(\boldsymbol{f}_t(\boldsymbol{x}'), y)}{T}
\end{aligned} \tag{6}$$

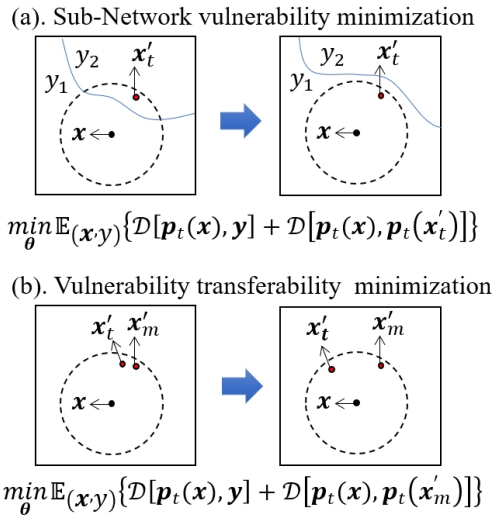

Figure 2: (a) Sub-network vulnerabilities are characterized by their most sensitive input perturbation $\boldsymbol{x}'_t$. (b) Minimizing vulnerability transferability reduces shared weakness across sub-networks, enhancing ensemble robustness.

The first inequality follows from (Deng et al. 2022). Eq. 6 reveals that evaluating adversarial robustness based solely on $\boldsymbol{f}(\boldsymbol{x})$ may underestimate the vulnerabilities of individual temporal sub-networks $\boldsymbol{f}_t(\boldsymbol{x})$.

Let $\boldsymbol{p}_t(\boldsymbol{x}) = \sigma(\boldsymbol{f}_t(\boldsymbol{x}))$ denote the predicted class distribution at timestep t, where σ is the softmax function. We quantify the maximum perturbation-induced distributional shift using a distance metric $\mathcal{D}$:

$$\max_{\|\boldsymbol{x}'-\boldsymbol{x}\|_p \le \varepsilon} \mathcal{D}[\boldsymbol{p}_t(\boldsymbol{x}), \boldsymbol{p}_t(\boldsymbol{x}')] \tag{7}$$

Moreover, the adversarial loss of $\boldsymbol{f}_t(\boldsymbol{x})$ under $\mathcal{D}$ can be upper-bounded via the triangle inequality:

$$\begin{aligned}
\max_{\|\boldsymbol{x}'-\boldsymbol{x}\|_p \le \varepsilon} \mathcal{D}[\boldsymbol{p}_t(\boldsymbol{x}'), \boldsymbol{y}] \le \mathcal{D}[\boldsymbol{p}_t(\boldsymbol{x}), \boldsymbol{y}] \\
+ \max_{\|\boldsymbol{x}'-\boldsymbol{x}\|_p \le \varepsilon} \mathcal{D}[\boldsymbol{p}_t(\boldsymbol{x}), \boldsymbol{p}_t(\boldsymbol{x}')]
\end{aligned} \tag{8}$$

Thus, minimizing the right-hand side of Inequality 8 serves as an effective strategy to reduce the adversarial vulnerability of temporal sub-networks (Figure 2a).

Vulnerability Transferability Minimization

To further mitigate vulnerability propagation across temporal sub-networks, we define $\boldsymbol{x}'_t$ as the input within an ℓ_p-bounded neighborhood of $\boldsymbol{x}$ that induces the maximum distributional shift for sub-network $\boldsymbol{f}_t$:

$$\boldsymbol{x}'_t = \arg \max_{\|\boldsymbol{x}'-\boldsymbol{x}\|_p \le \varepsilon} \mathcal{D}[\boldsymbol{p}_t(\boldsymbol{x}), \boldsymbol{p}_t(\boldsymbol{x}')] \tag{9}$$

We evaluate transferability by applying $\boldsymbol{x}'_t$ to another temporal sub-network $\boldsymbol{f}_m$ $(m \neq t)$. The resulting adversarial loss can be upper bounded via the triangle inequality:

$$\mathcal{D}[\boldsymbol{p}_m(\boldsymbol{x}'_t), \boldsymbol{y}] \le \mathcal{D}[\boldsymbol{p}_m(\boldsymbol{x}), \boldsymbol{y}] + \mathcal{D}[\boldsymbol{p}_m(\boldsymbol{x}), \boldsymbol{p}_m(\boldsymbol{x}'_t)] \tag{10}$$

We minimize the right-hand side of Eq. 10 to reduce the vulnerability transferability from $\boldsymbol{f}_t$ to $\boldsymbol{f}_m$. A lower value indicates that $\boldsymbol{f}_t$'s vulnerability does not significantly affect $\boldsymbol{f}_m$, implying disjoint adversarial regions and reduced temporal correlation (Figure 2b).

By combining Eq. 8-10, we can quantify the cross-timestep distributional shift of the temporal sub-networks in SNNs using a transferability matrix:

$$\mathcal{L}^{(t,m)} = \mathcal{D}[\boldsymbol{p}_m(\boldsymbol{x}), \boldsymbol{p}_m(\boldsymbol{x}'_t)] \tag{11}$$

where the diagonal elements of the matrix $\mathcal{L}^{(t,t)}$ represent the self-robustness of each sub-network $\boldsymbol{f}_t(\boldsymbol{x})$. Lower $\mathcal{L}^{(t,t)}$ values indicate stronger resistance to perturbations within the neighborhood of $\boldsymbol{x}$ (Figure 2a). The off-diagonal elements $\mathcal{L}^{(t,m)}(t \neq m)$ represent the inter-subnetwork transferability of distributional shift. Smaller values reflect lower inter-timestep correlation and higher temporal diversity.

Robust Temporal Self-Ensemble

Building upon the vulnerability metrics in Eq. 8 and 10, we formulate an adversarial training objective that simultaneously mitigates the intrinsic vulnerability of each temporal sub-network and suppresses the transferability of adversarial perturbations across timesteps. Specifically, we define the Robust Temporal self-Ensemble (RTE) objective as:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)} \sum_{t=1}^{T} \sum_{m=1}^{T} \{\mathcal{D}[\boldsymbol{p}_t(\boldsymbol{x}), \boldsymbol{y}] + \mathcal{D}[\boldsymbol{p}_t(\boldsymbol{x}), \boldsymbol{p}_t(\boldsymbol{x}'_m)]\} \tag{12}$$

Here, $\boldsymbol{\theta}$ denotes the trainable parameters of the SNN. The first term encourages accurate predictions at each timestep, while the second term jointly reduces sub-network vulnerability and inter-timestep vulnerability alignment.

Directly computing the adversarial examples $\boldsymbol{x}'_m$ for all $m \in \{1, 2, ..., T\}$ is computationally prohibitive, particularly due to the temporal coupling introduced by Batch Normalization (BN) layers, which require full-sequence inputs across T steps for a single forward pass (Duan et al. 2022).

To alleviate this overhead, we adopt a stochastic sampling strategy. At each training iteration, a single timestep m is randomly selected to generate the adversarial example $\boldsymbol{x}'_m$. We then optimize the following surrogate objective:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)} \sum_{t=1}^{T} \{\mathcal{D}[\boldsymbol{p}_t(\boldsymbol{x}), \boldsymbol{y}] + \mathcal{D}[\boldsymbol{p}_t(\boldsymbol{x}), \boldsymbol{p}_t(\boldsymbol{x}'_m)]\} \tag{13}$$

This approximation substantially reduces computational cost by limiting the perturbation generation to a single sub-network per iteration, while still offering effective gradient signals for improving robustness.

In practice, the raw distance metric $\mathcal{D}$ may be non-smooth and suboptimal for training. We therefore adopt a smoothed variant—its squared form—and further relax it using the Pinsker inequality when $\mathcal{D}$ is the ℓ_1 distance:

$$\frac{1}{2}\|\boldsymbol{p} - \boldsymbol{q}\|_1^2 \leq \mathrm{KL}(\boldsymbol{p}\|\boldsymbol{q}) \tag{14}$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ denote arbitrary predicted class distributions. This yields a KL-based formulation that aligns with distillation-style regularization (Ding et al. 2025), enhancing both optimization stability and theoretical interpretability. We introduce a regularization coefficient γ to balance clean and robust performance, and define the final training loss of RTE as:

$$\mathcal{L}_{\mathrm{RTE}} = \frac{1}{T} \sum_{t=1}^{T} \{\mathrm{KL}[\boldsymbol{p}_t(\boldsymbol{x})\|\boldsymbol{y}] + \gamma\, \mathrm{KL}[\boldsymbol{p}_t(\boldsymbol{x})\|\boldsymbol{p}_t(\boldsymbol{x}'_m)]\} \tag{15}$$

The overall training procedure is summarized in Algorithm 1.

Algorithm 1: Robust Temporal self-Ensemble Training

Require: SNN with T timesteps, trainable parameter $\boldsymbol{\theta}$, training epochs E, number of batched input-label pairs B, learning rate η, perturbation magnitudes ε, PGD iteration steps K, PGD step size α.

Initialization SNN
for $e = 1$ **to** E **do** ▷ Training iteration
  for $b = 1$ **to** B **do**
    Sample batched input data $(\boldsymbol{x}, y)$
    Randomly sample $m \in \{1, 2, ..., T\}$
    $\boldsymbol{x}'_{\boldsymbol{m}} \leftarrow \boldsymbol{x} + \boldsymbol{U}(-\varepsilon, \varepsilon)$, where $\boldsymbol{U}$ denotes sampling from a uniform distribution.
    for $k = 1$ **to** K **do**
      Update $\boldsymbol{x}'_{\boldsymbol{m}}$ by $\boldsymbol{x}'_{\boldsymbol{m}} \leftarrow \Pi_{\mathbb{B}(\boldsymbol{x},\varepsilon)}\left[\alpha \operatorname{sign}\left(\nabla_{\boldsymbol{x}'_{\boldsymbol{m}}} \mathrm{KL}[\boldsymbol{p}_m(\boldsymbol{x})\|\boldsymbol{p}_m(\boldsymbol{x}'_m)]\right)\right] + \boldsymbol{x}'_{\boldsymbol{m}}$, where $\Pi_{\mathbb{B}(\boldsymbol{x},\varepsilon)}$ projects the input onto the bounded interval $\mathbb{B}(\boldsymbol{x}, \varepsilon)$.
    end for
    Calculate $\mathcal{L}_{\mathrm{RTE}}$ using $\boldsymbol{x}'_{\boldsymbol{m}}$ as Eq. 15
    Calculate gradients via backpropagation and update the parameters:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{RTE}}$$

  end for
end for

Experiments

We evaluate our method on four benchmark datasets: SVHN (Netzer et al. 2011), CIFAR-10, CIFAR-100, and Tiny-ImageNet(Deng et al. 2009). We use VGG-9 (Duan et al. 2022) for SVHN, SEWResNet-19 (Fang et al. 2021) for CIFAR-10/100, and SEWResNet-20 for Tiny-ImageNet. Unless otherwise specified, all SNNs are configured with a membrane decay rate $\lambda = 0.5$ and a spiking threshold $V_{\text{th}} = 0.5$. Models are trained on NVIDIA A100 GPUs, and detailed hyperparameters are provided in Appendix B.

We assess adversarial robustness under l_∞-bounded perturbations. We adopt standard white-box attacks including FGSM, PGD$_k$, and AutoPGD with both cross-entropy (APGD$^{\text{CE}}$) and DLR (APGD$^{\text{DLR}}$) losses, and use the Square attack (Andriushchenko et al. 2020) for black-box evaluation. Perturbation budgets are set to $\varepsilon = 8/255$ for SVHN and CIFAR, and $\varepsilon = 4/255$ for Tiny-ImageNet. Following (Liu et al. 2024), we evaluate with both surrogate gradi-

Datasets	Methods	Clean	FGSM	PGD_{10}	PGD_{50}	$APGD_{50}^{CE}$	$APGD_{50}^{DLR}$	Square
SVHN T=4	AT	92.81	67.90	54.42	49.79	40.99	44.19	51.30
	RAT	92.78	66.71	55.45	51.53	44.59	44.69	56.71
	AT+MPPD	92.62	66.78	53.51	49.79	40.33	42.81	50.11
	AT+SR	90.68	68.36	**62.17**	**60.31**	**53.63**	48.16	53.98
	TRADES	**94.00**	**71.98**	59.36	55.30	48.58	48.14	57.04
	RTE	93.59	71.38	59.78	56.56	50.63	50.06	58.43
	RTE+SR	92.82	69.43	59.89	57.24	52.36	**51.03**	**58.54**
CIFAR -10 T=4	AT	79.04	51.17	44.99	41.87	26.92	29.75	32.66
	RAT	80.24	52.70	46.81	44.41	31.71	34.74	41.99
	AT+MPPD	79.20	50.87	44.63	41.96	26.59	30.03	31.43
	AT+SR	78.55	**55.01**	**51.61**	**50.82**	**41.40**	**41.21**	49.36
	TRADES	81.80	53.03	47.09	45.37	34.68	36.79	46.53
	RTE	**81.90**	53.14	48.07	46.33	36.38	38.60	47.05
	RTE+SR	80.85	53.45	49.05	47.73	40.31	40.55	**52.79**
CIFAR -100 T=4	AT	51.82	25.86	22.20	21.07	13.08	15.05	17.20
	RAT	54.99	**28.96**	25.69	24.38	17.34	18.65	23.98
	AT+MPPD	52.09	25.07	21.77	20.81	13.05	15.20	18.11
	AT+SR	51.72	28.44	**26.12**	25.07	18.57	19.23	23.58
	TRADES	56.97	27.67	24.70	23.95	17.66	17.39	24.69
	RTE	**59.50**	28.66	25.45	24.28	17.77	18.80	26.60
	RTE+SR	58.10	28.90	26.07	**25.52**	**20.18**	**19.95**	**30.24**
Tiny-ImageNet T=4	AT	42.36	24.12	22.50	22.17	10.93	12.58	12.47
	RAT	46.58	**29.37**	**27.68**	27.32	16.78	17.25	17.99
	AT+MPPD	42.77	24.11	22.25	21.98	10.93	12.70	11.14
	AT+SR	43.10	27.04	25.65	25.46	17.22	17.70	18.55
	TRADES	47.76	27.68	25.85	26.08	16.61	16.59	20.39
	RTE	**50.46**	28.20	26.56	26.19	17.80	19.17	22.24
	RTE+SR	48.78	28.67	27.48	**27.39**	**20.15**	**20.26**	**25.04**

Table 1: Comparison with the state-of-the-art training methods. Square attack is a black-box attack while other attacks are white-box attacks. The worst-case accuracy across all attacks is underlined and used as the robustness evaluation metric.

ents and rate gradient approximation (RGA) (Bu et al. 2023), and report the worst-case accuracy across variants(Detailed results can be found in the Appendix D).

Comparison with State-of-the-Art Methods

We compare our proposed RTE method with two standard baselines, AT and TRADES, as well as several state-of-the-art regularization-based adversarial defense methods for SNNs, including RAT, MPPD, and SR. For fair comparison, the loss function between the model's output and the true label is formulated using the framework of TET (Deng et al. 2022). The training perturbation bounds are set to $\varepsilon = 8/255$ for SVHN, CIFAR-10, and CIFAR-100, and $\varepsilon = 4/255$ for Tiny-ImageNet. Consistent with their original implementations, SR and MPPD are combined with AT (denoted as AT+SR and AT+MPPD). Specific regularization settings are detailed in Appendix C.

As shown in Table 1, RTE achieves the highest clean accuracy on CIFAR-100 and Tiny-ImageNet, with improvements of 2.53% and 2.70%, respectively. On SVHN, its clean accuracy is slightly lower than that of TRADES by 0.41%. For robustness evaluation, we notice that under weaker white-box attacks such as FGSM and PGD, the measured robust accuracy sometimes exceeds that under the black-box Square attack. This counterintuitive result suggests the presence of gradient obfuscation in SNNs. Although different methods appear to excel under different attack, we use the worst-case accuracy across all attacks as the final robustness measure, with the results underlined in Table 1. The worst-case robust accuracy serves as the tightest upper bound of a model's true robustness and is therefore the most reliable metric for robustness evaluation.

RTE consistently demonstrates superior robustness on SVHN and Tiny-ImageNet, and performs comparably to the best-performing AT+SR method on CIFAR datasets. When further combined with SR (denoted RTE+SR), our method achieves stronger robustness and surpasses AT+SR on CIFAR-100. Although RTE+SR lags slightly behind AT+SR on CIFAR-10 in worst-case robustness (by 0.90%), it surpasses AT+SR in clean accuracy by 2.30%.

Importantly, prior methods such as AT+SR often sacrifice clean accuracy for robustness. In contrast, RTE achieves a better trade-off. Following (Wu et al. 2024), we adopt the sum of clean and robust accuracy as the trade-off metric. As illustrated in Figure 3, RTE+SR achieves the best trade-off on CIFAR-10 and significantly outperforms all baselines on other datasets. Notably, on CIFAR-100 and Tiny-ImageNet, RTE+SR improves the trade-off metric by 3.92 and 4.80 points respectively over the strongest baseline. We attribute these improvements to the enhanced smoothness introduced

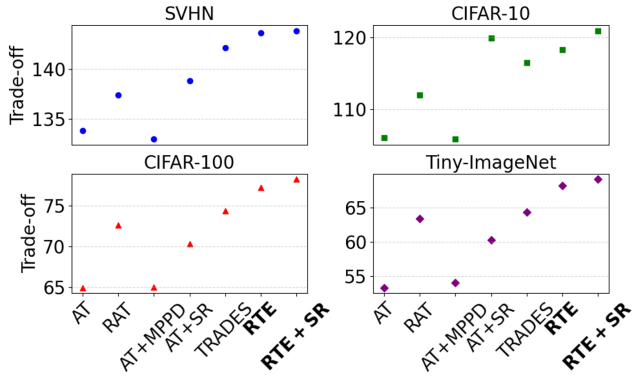

Figure 3: Robustness-accuracy trade-off achieved by different training methods.

by SR, which has been shown to facilitate robust ensemble representations (Yang et al. 2021).

In terms of training cost, RTE shares similar computational overhead with other methods due to the use of PGD-based adversarial perturbations generation. As detailed in Appendix E, RTE only incurs slightly more cost than AT, making it a practical and scalable defense strategy.

Trade-off under Varying Regularization Strength

To analyze the robustness-accuracy trade-off behavior, we vary the regularization coefficient γ for both TRADES and our proposed RTE method. TRADES is selected as a primary comparison baseline due to its wide adoption and explicit trade-off control via γ. Figure 4 illustrates the clean and robust accuracy on CIFAR-10 and CIFAR-100 under different γ values. Each point represents a particular setting of RTE (stars) or TRADES (triangles), and the dashed lines connect points of equal trade-off levels. RTE consistently outperforms TRADES across the trade-off spectrum, achieving more favorable clean-robustness balances, particularly on CIFAR-100. These results demonstrate that RTE provides a consistently better robustness-accuracy trade-off than TRADES across a range of regularization strengths.

Effect of Temporal Resolution on RTE

RTE is explicitly designed to enhance the ensemble effect among temporal sub-networks in SNNs. As the number of timesteps increases, more diverse sub-network predictions contribute to the final output, which can improve adversarial robustness. To examine this effect, we investigate how varying the timestep count T impacts the robustness-accuracy trade-off across four representative training methods: AT, RAT, TRADES, and RTE.

As shown in Figure 5, increasing T from 4 to 12 consistently improves the trade-off performance for all methods on CIFAR-10 and CIFAR-100. However, RTE consistently achieves the best trade-off across all T values, highlighting its superior ability to leverage temporal diversity. This result suggests that RTE not only benefits from the inherent ensemble nature of SNNs but also scales effectively with increasing temporal resolution.

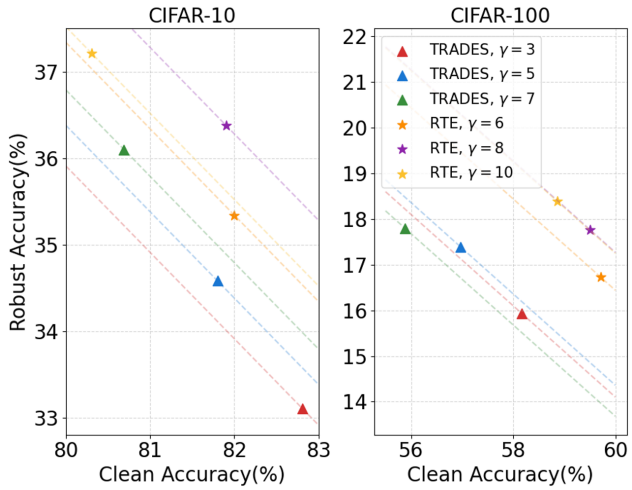

Figure 4: Clean and robust accuracy of RTE (stars) and TRADES (triangles) on CIFAR-10 and CIFAR-100 across different γ. Dashed lines indicate equal trade-off levels (clean + robust accuracy).

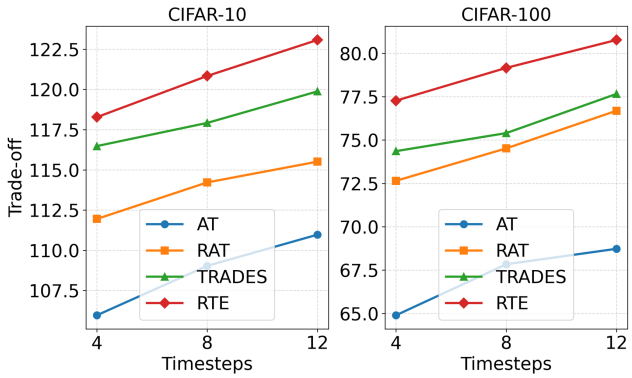

Figure 5: Robustness–accuracy trade-off of SNNs trained with different methods under varying timestep counts T on CIFAR-10 and CIFAR-100.

Temporal Robustness Analysis through Transferability Matrices

To better understand the internal dynamics of adversarial robustness in SNNs' temporal sub-networks, we analyze the transferability matrices according to Eq. 11.

Figure 6 presents the resulting matrices for four training strategies: AT, RAT, TRADES and our proposed RTE. The SNNs in the figure are all trained on the CIFAR-10 dataset with a timestep count of 8. The results show that the adversarial transferability between sub-networks shows a decreasing trend as the timestep gap increases, which is consistent with general intuition. Notably, both the diagonal and off-diagonal elements of the transferability matrix for RTE are in the lowest range. These findings highlight the effectiveness of RTE in promoting self-robustness and diversity in temporal sub-networks. By explicitly decoupling vulnerability transferability across timesteps, RTE effectively strength-

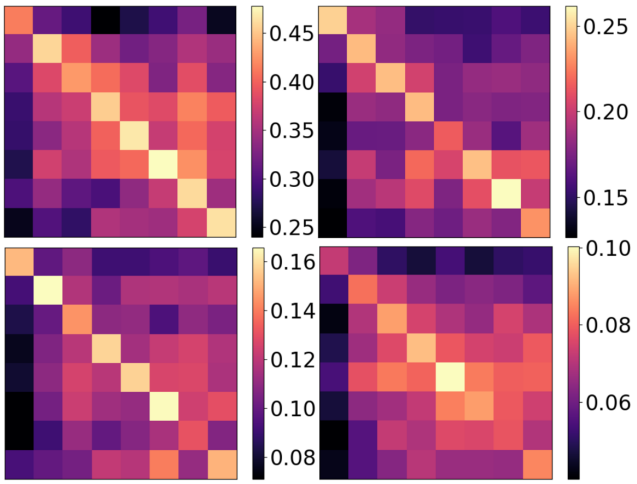

Figure 6: Transferability matrices of AT (upper left), RAT (upper right), TRADES (lower left) and RTE (lower right). Lower diagonal values indicate better self-robustness of sub-networks, while lower off-diagonal values indicate reduced cross-timestep adversarial transferability.

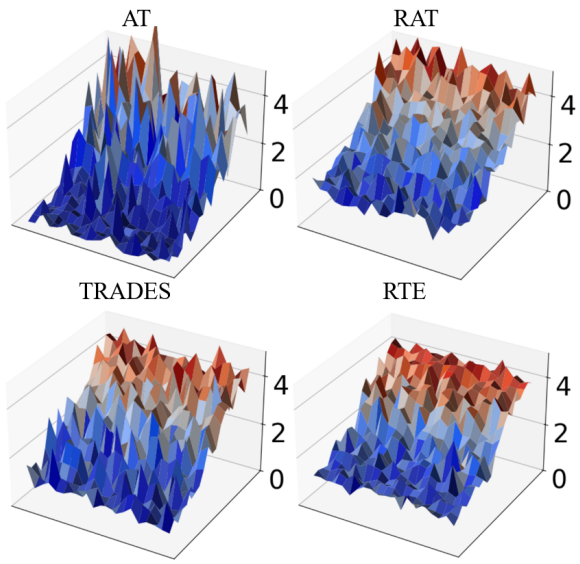

Figure 7: Loss surface visualization of SNNs trained with different methods on a CIFAR-100 test sample.

ens the ensemble robustness of SNNs.

In Figure 6 we define $\mathcal{D}$ in Eq. 11 as the KL divergence. We also provide the transferability matrices under l_2-norm distance metric in the Appendix F, where the key patterns are consistent with those obtained using KL divergence.

Loss Landscape and Temporal Feature Analysis

Loss Surface Visualization. We visualize the loss landscapes of SNNs trained with different methods on a representative CIFAR-100 test sample. As shown in Figure 7, the SNN trained with RTE exhibits a significantly smoother and flatter loss surface, along with lower loss values compared to other methods. This suggests that RTE helps stabilize model predictions against input perturbations, indicating stronger adversarial robustness. We also visualize the loss surfaces of sub-networks at different timesteps in Appendix G.

Feature Space Visualization. To further evaluate the robustness of temporal sub-networks, we use t-SNE to visualize the final-layer feature distributions at different timesteps. Figure 8 compares the feature embeddings from the 2nd, 3rd, and 4th timesteps of SNNs trained with AT (top row) and RTE (bottom row), using 1,000 CIFAR-10 test samples. The RTE-trained sub-networks consistently exhibit more compact and class-discriminative clusters across timesteps. This demonstrates that RTE improves not only the overall robustness, but also the representation consistency and class separability of individual sub-networks. In Appendix H, we provide additional visualizations of the feature space after temporal ensemble for more methods.

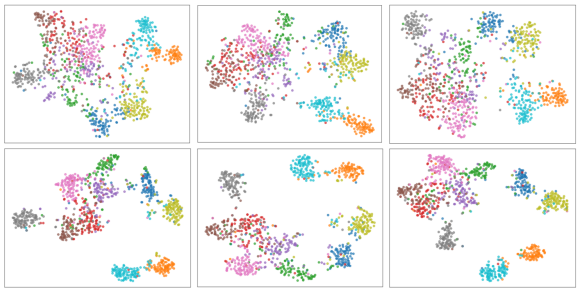

Figure 8: t-SNE visualization of final-layer features from sub-networks on CIFAR-10. Each column from left to right represents the 2nd to 4th timesteps, respectively. Top: SNN trained with AT; Bottom: SNN trained with RTE.

Conclusion

In this work, we revisit adversarial robustness in SNNs through the lens of temporal self-ensembling. By interpreting SNNs as a sequence of temporally evolving sub-networks, we reveal two intertwined robustness challenges, the insufficient resilience of individual sub-networks and the vulnerability propagation across timesteps. To address these issues, we propose Robust Temporal self-Ensemble (RTE), a novel training framework that simultaneously strengthens each sub-network's robustness and suppresses the temporal alignment of adversarial vulnerabilities. RTE introduces a principled loss formulation inspired by ensemble learning and employs a stochastic optimization strategy to maintain computational efficiency. Empirical results across four benchmarks demonstrate that RTE consistently achieves state-of-the-art robustness-accuracy trade-offs, while significantly enhancing the diversity and independence among temporal sub-networks. Our analyses further show that RTE reshapes the internal robustness landscape of SNNs by reducing cross-timestep vulnerability transfer, ultimately fostering more reliable and adversarially robust spiking models.

Acknowledgments

This study is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB1010302), the General Program of the National Natural Science Foundation of China (Grant No. 32571284) and the State Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology (Grant No. JS202401).

References

Andriushchenko, M.; Croce, F.; Flammarion, N.; and Hein, M. 2020. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, 484–501. Springer.

Bu, T.; Ding, J.; Hao, Z.; and Yu, Z. 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.

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

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Deng, S.; Li, Y.; Zhang, S.; and Gu, S. 2022. Temporal Efficient Training of Spiking Neural Network via Gradient Reweighting. In *International Conference on Learning Representations*.

Deng, Y.; and Mu, T. 2023. Understanding and improving ensemble adversarial defense. *Advances in Neural Information Processing Systems*, 36: 58075–58087.

Ding, J.; Bu, T.; Yu, Z.; Huang, T.; and Liu, J. 2022. Snn-rat: Robustness-enhanced spiking neural network through regularized adversarial training. *Advances in Neural Information Processing Systems*, 35: 24780–24793.

Ding, J.; Pan, Z.; Liu, Y.; Yu, Z.; and Huang, T. 2024a. Robust Stable Spiking Neural Networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 11016–11029. PMLR.

Ding, J.; Yu, Z.; Huang, T.; and Liu, J. K. 2024b. Enhancing the robustness of spiking neural networks with stochastic gating mechanisms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 492–502.

Ding, Y.; Zuo, L.; Jing, M.; He, P.; and Deng, H. 2025. Rethinking spiking neural networks from an ensemble learning perspective. *arXiv preprint arXiv:2502.14218*.

Duan, C.; Ding, J.; Chen, S.; Yu, Z.; and Huang, T. 2022. Temporal effective batch normalization in spiking neural networks. *Advances in Neural Information Processing Systems*, 35: 34377–34390.

Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; and Tian, Y. 2021. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34: 21056–21069.

Goodfellow, I.; Bengio, Y.; Courville, A.; and Bengio, Y. 2016. *Deep learning*, volume 1. MIT Press.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Hao, Z.; Bu, T.; Shi, X.; Huang, Z.; Yu, Z.; and Huang, T. 2023. Threaten spiking neural networks through combining rate and temporal information. In *The Twelfth International Conference on Learning Representations*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. *Technical Report*.

Li, J.; Shen, G.; Zhao, D.; Zhang, Q.; and Zeng, Y. 2023. Firefly: A high-throughput hardware accelerator for spiking neural networks with efficient dsp and memory optimization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 31(8): 1178–1191.

Liang, L.; Hu, X.; Deng, L.; Wu, Y.; Li, G.; Ding, Y.; Li, P.; and Xie, Y. 2021. Exploring adversarial attack in spiking neural networks with spike-compatible gradient. *IEEE transactions on neural networks and learning systems*, 34(5): 2569–2583.

Liu, Y.; Bu, T.; Ding, J.; Hao, Z.; Huang, T.; and Yu, Z. 2024. Enhancing Adversarial Robustness in SNNs with Sparse Gradients. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 30738–30754. PMLR.

Lun, L.; Feng, K.; Ni, Q.; Liang, L.; Wang, Y.; Li, Y.; Yu, D.; and Cui, X. 2025. Towards Effective and Sparse Adversarial Attack on Spiking Neural Networks via Breaking Invisible Surrogate Gradients. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 3540–3551.

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

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Mukhoty, B.; AlQuabeh, H.; and Gu, B. 2025. Improving Generalization and Robustness in SNNs Through Signed Rate Encoding and Sparse Encoding Attacks. In *The Thirteenth International Conference on Learning Representations*.

Mukhoty, B.; AlQuabeh, H.; Masi, G. D.; Xiong, H.; and Gu, B. 2024. Certified Adversarial Robustness for Rate Encoded Spiking Neural Networks. In *The Twelfth International Conference on Learning Representations*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 4. Granada.

Pang, T.; Xu, K.; Du, C.; Chen, N.; and Zhu, J. 2019. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, 4970–4979. PMLR.

Pei, J.; Deng, L.; Song, S.; Zhao, M.; Zhang, Y.; Wu, S.; Wang, G.; Zou, Z.; Wu, Z.; He, W.; et al. 2019. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature*, 572(7767): 106–111.

Roy, K.; Jaiswal, A.; and Panda, P. 2019. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784): 607–617.

Shen, G.; Zhao, D.; Dong, Y.; and Zeng, Y. 2023. Brain-inspired neural circuit evolution for spiking neural networks. *Proceedings of the National Academy of Sciences*, 120(39): e2218173120.

Wang, J.; Zhao, D.; Du, C.; He, X.; Zhang, Q.; and Zeng, Y. 2025. Random heterogeneous spiking neural network for adversarial defense. *iScience*, 28(6).

Wu, K.; Yao, M.; Chou, Y.; Qiu, X.; Yang, R.; Xu, B.; and Li, G. 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*, 2748–2756.

Xu, M.; Ma, D.; Tang, H.; Zheng, Q.; and Pan, G. 2024. FEEL-SNN: Robust spiking neural networks with frequency encoding and evolutionary leak factor. *Advances in Neural Information Processing Systems*, 37: 91930–91950.

Yang, H.; Zhang, J.; Dong, H.; Inkawhich, N.; Gardner, A.; Touchet, A.; Wilkes, W.; Berry, H.; and Li, H. 2020. Dverge: diversifying vulnerabilities for enhanced robust generation of ensembles. *Advances in Neural Information Processing Systems*, 33: 5505–5515.

Yang, Z.; Li, L.; Xu, X.; Zuo, S.; Chen, Q.; Zhou, P.; Rubinstein, B.; Zhang, C.; and Li, B. 2021. TRS: Transferability Reduced Ensemble via Promoting Gradient Diversity and Model Smoothness. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 17642–17655. Curran Associates, Inc.

Zhao, D.; Shen, G.; Dong, Y.; Li, Y.; and Zeng, Y. 2025. Improving stability and performance of spiking neural networks through enhancing temporal consistency. *Pattern Recognition*, 159: 111094.