Appendix: Robust Spiking Neural Networks by Temporal Mutual Information

Supplementary Material

1. More Related Work

1.1. Information Bottleneck (IB) Principle

More recently, [24] have construed deep learning as a successive information-extraction process quantified by the Information Bottleneck (IB) principle [23]. Their objective is to demystify the workings of deep learning by visualizing the behavior of the compression and prediction terms. Information bottleneck encourages the model to learn an optimal representation by diminishing the irrelevant parts of the input variable that do not contribute to the prediction [21]. It shows that when the valid information of the input cannot be well extracted by the model, that is, the mutual information between the input data and its related latent representation is high, the neural network cannot generalize the out-of-distribution data (such as adversarial example) well [30]. This has spurred numerous studies applying the IB principle to diverse tasks, such as adversarial robustness learning [29, 30], learning minimal sufficient representation [14, 25], exploring the generalization error [20], and interpreting the learning dynamics of DNNs [3, 19]. However, since mutual information measures the correlation between two distributions, despite the effectiveness of the IB principle, calculating mutual information in high dimensions on a single image in ANN is challenging. Therefore, some recent techniques ease the constraint by transforming mutual information calculation into a network optimization problem without explicitly estimating mutual information [6, 22]. Alexenders *et al*. [2] demonstrate a variational approximation to parameterize the IB model. Peng *et al*. [17] propose a variational discriminator bottleneck for stable adversarial learning. However, these methods aim to learn a satisfied representation rather than calculating the mutual information explicitly. Some approaches have been developed to estimate the mutual information by converting a single image into a distribution [6, 18]. However, these methods may degrade class-aware downstream task performance when sampling from batches [6] or obtain inaccurate mutual information caused by noise distribution [13]. Other methods focus on replacing the mutual information with the output entropy to select adversarial examples [27, 30]. However, entropy is essentially the lower bound of mutual information [30].

1.2. Mutual Information for Adversarial Attack

There are some other works [5, 31] directly connect mutual information to model robustness without IB principle. Zhao *et al*. enhance the adversarial robustness by maximizing the natural mutual information and minimizing the adversarial mutual information of the adversarial example [30]. Zhou *et al*. constrain MI between original and adversarial samples at the sample level [31]. Atsague *et al*. use MI between the probabilistic predictions of natural and adversarial examples as a regularization term to the standard adversarial training [5]. These methods focus on the mutual information between different examples, rather than based on the original information transmitted by the model. In contrast, we leverage mutual information to study the information transmitted by the model, based on which we show that the upper bound of the robustness error for a deep neural network is determined by the mutual information between the input feature and latent representation.

1.3. Temporal SNN Work

Our work is not an extension of prior temporal dynamics analysis, but introduces a robustness-oriented, information theoretic perspective with different conclusions and implications. Prior works analyze temporal dynamics for representation learning [12], training efficiency [12], or supervision [28]. In contrast, we study adversarial robustness and establish a connection between temporal information flow and a robustness error bound.

2. More Preliminary

2.1. Dircet Training Strategy for SNNs

Three direct training strategies for SNN are introduced here, including formal training strategies (i.e., spatial-temporal backpropagation (STBP [26]), temporal efficient training (TET [1])) and adversarial training strategy (i.e., SNN-RAT [9]).

STBP. STBP unrolls the SNN over time-steps and accumulates gradient at each time-step [16, 26]. The loss function of standard direct training $\mathcal{L}_{SDT}$ used in STBP method is:

$$\mathcal{L}_{SDT} = \mathcal{L}_{CE}(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{O}(t), y_{true}), \tag{1}$$

where $\boldsymbol{O}(t)$ represents the output spike of the output layer, T is the total simulation time, $\mathcal{L}_{CE}$ denotes the cross-entropy loss, and y_{true} represents the correct label.

TET. TET follows the backpropagation rule as it is in STBP, while coming up with a new kind of loss function $\mathcal{L}_{TET}$ to realize temporal efficient training. It constrains the output spike at each moment to be close to the target distribution. It is described as [1]:

$$\mathcal{L}_{TET} = \frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{CE}(\boldsymbol{O}(t), y_{true}), \tag{2}$$

where $\boldsymbol{O}(t)$ represents the output spike of the output layer, T is the total simulation time, $\mathcal{L}_{CE}$ denotes the cross-entropy loss, and y_{true} represents the correct label.

SNN-RAT. SNN-RAT proposes a regularized adversarial training scheme with low computational overheads to enhance the robustness of SNNs when attacked by adversarial examples. Specifically, SNN-RAT updates weights to the target of the orthogonal matrix to propose an orthogonal regularization $\mathcal{L}(\boldsymbol{W})$, and arguments the training dataset with adversarial examples during training. It follows the backpropagation rule as it is in STBP.

2.2. Adversarial Attack Methods

Given a classification model f with dataset $(\boldsymbol{x}, y_{true})$, where $\boldsymbol{x}$ is the clean image and y_{true} is the corresponding correct label. The adversarial attack aims to generate an adversarial example $\hat{\boldsymbol{x}}$ that satisfies:

$$f(\hat{\boldsymbol{x}}) \neq f(\boldsymbol{x}) \quad s.t. \quad ||\hat{\boldsymbol{x}} - \boldsymbol{x}||_p \leq \epsilon, \tag{3}$$

where$||\cdot||_p$ is the L_p-norm, we use L_∞-norm on our work, and ϵ limits the strength of the perturbation to a level that is indistinguishable to the human eye. Here we consider three classic adversarial attack algorithms: Fast Gradient Sign Method (FGSM) [10], Projected Gradient Descent (PGD) [15], and their Rate Gradient Approximation Attack (RGA) variants [7] proposed for SNNs.

FGSM. The Fast Gradient Sign Method (FGSM) is designed to perturb the original data $\boldsymbol{x}$ by taking a single step along the sign direction of the gradient with respect to the loss function. The objective is to induce a noticeable change in the linear output through this perturbation, ultimately leading to the misguidance of the neural network. This process can be formally articulated as follows:

$$\hat{\boldsymbol{x}} = \boldsymbol{x} + \epsilon \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x}), y_{true})), \tag{4}$$

where $\text{sign}(\cdot)$ is an odd mathematical function that extracts the sign of a real number.

PGD. PGD attack is the iterative variant of FGSM. It first starts from a random perturbation in the L_p-norm constraint around the original sample $\boldsymbol{x}$, then takes a gradient iteration step in the sign direction to achieve the greatest loss output, it can be formalized as follows:

$$\hat{\boldsymbol{x}}^0 = \boldsymbol{x} + \text{U}(-\epsilon, +\epsilon), \tag{5}$$

$$\hat{\boldsymbol{x}}^{k+1} = \text{Clip}_{\boldsymbol{x},\epsilon}\{\hat{\boldsymbol{x}}^k + \alpha \cdot \text{sign}(\nabla_{\hat{\boldsymbol{x}}^k} \mathcal{L}(f(\hat{\boldsymbol{x}}^k), y_{true}))\}, \tag{6}$$

where k is the iterative step, α is step size for each attack iteration, ϵ controls the perturbation level. $\text{U}(\cdot)$ is a uniform function, $\text{Clip}_{x,\epsilon}\{\boldsymbol{x}\}$ is the function which performs per-pixel clipping of the image $\hat{\boldsymbol{x}}$, so the result will be in L_∞-norm ϵ-neighbourhood of the original image $\boldsymbol{x}$.

RGA. The above adversarial attack methods all use the surrogate gradient of STBP [26] to generate adversarial examples when applied to SNNs, while the RGA attack takes advantage of the rate coding features propagated between the layers of the SNN model to generate adversarial perturbations. Let vectors $\boldsymbol{r}^l$ and $\boldsymbol{I}^l$ to denote the firing rates and average input currents of all neurons in layer l, respectively, and use vector $\boldsymbol{r}^0$ to denote the input image. The gradient propagates from the loss function to the input image in RGA attack is formulated as:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{r}^0} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{r}^L} \left(\prod_{l=1}^{L} \frac{\partial \boldsymbol{r}^l}{\partial \boldsymbol{I}^l} \frac{\partial \boldsymbol{I}^l}{\partial \boldsymbol{r}^{l-1}}\right). \tag{7}$$

In all attack methodologies under consideration, we examine two distinct scenarios: white-box attack and black-box attack. In a white-box attack, the adversary possesses full access to the model's topology, model parameters, and gradients. Conversely, in a black-box attack, the attacker is limited to acquiring only basic information about the model.

3. More Experiments

3.1. More Experimental Setup

TMI regularizer training. After TMI regularizer implementation, we embed it to the objective function (Eq.(17)). When training the SNN, we calculate the gradients via spatial-temporal backpropagation (STBP [26]). Mathematically, this backpropagation chain rule is described as

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}} = \sum_t \frac{\partial \mathcal{L}}{\partial \boldsymbol{S}(t)} \frac{\partial \boldsymbol{S}(t)}{\partial \boldsymbol{U}(t)} \frac{\partial \boldsymbol{U}(t)}{\partial \boldsymbol{I}(t)} \frac{\partial \boldsymbol{I}(t)}{\partial \boldsymbol{W}}, \tag{8}$$

where $\boldsymbol{U}(t)$ is the membrane potential at time t, $\boldsymbol{I}(t)$ denotes the pre-synaptic current input, which is the product of synaptic weight $\boldsymbol{W}$ and spiking input $\boldsymbol{S}(t)$. Because $\frac{\partial \boldsymbol{S}(t)}{\partial \boldsymbol{U}(t)}$ is the gradient of the non-differentiability step function, and use the surrogate gradient described below to estimate it.

$$\frac{\partial \boldsymbol{S}(t)}{\partial \boldsymbol{U}(t)} = \frac{1}{\gamma^2} \max(0, \gamma - |\boldsymbol{U}(t) - V_{th}|), \tag{9}$$

where γ denotes the constraint factor that determines the sample range to activate the gradient. We set $\gamma = 1.0$ and threshold $V_{th} = 1.0$ in Eq. (9) following the general settings [8].

In our work, the training process lasts for 300 epochs for all experiments. Batch normalization are used in the network to overcome the gradient vanishing or explosion. Adam optimizer is deployed, and the initial learning rate is set to 0.01. The learning rate uses a cosine annealing schedule with T_{max} equaling the max number of epochs. 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 on NVIDIA GeForce RTX 3090.

Table 1. Test accuracy (%) comparison of SNN models with/without TMI regularizer on **CIFAR-10 dataset**. The original accuracy of each defense is described in the column "Natural". "-**TMI**" represents embedding our proposed TMI regularizer into training, "-**H**" represents using output entropy [31], "-**AIMIE**" represents calculating mutual information by histogram method [11].

Dataset	Networks	Methods	Natural	white-box				black-box				Gaussian-noise
				FGSM	FGSM-RGA	PGD	PGD-RGA	FGSM	FGSM-RGA	PGD	PGD-RGA	
CIFAR-10	VGGSNN	STBP	91.81	23.00	21.17	2.38	2.10	45.44	42.92	30.18	30.12	66.56
		STBP-H	92.14	22.56	21.34	2.27	2.24	40.56	40.10	28.34	28.21	66.67
		STBP-AIMIE	91.69	23.16	21.39	2.39	2.61	41.33	41.60	29.67	30.97	66.29
		STBP-TMI(Ours)	**92.39**	**25.80**	**24.19**	**4.36**	**4.26**	**48.95**	**47.86**	**32.78**	**31.95**	**67.61**
		TET	92.59	27.29	28.85	6.05	8.38	55.39	55.17	46.07	46.62	65.09
		TET-H	92.23	26.32	25.68	5.89	7.13	55.18	55.08	45.38	45.45	66.80
		TET-AIMIE	92.91	25.94	25.13	5.62	8.11	56.90	56.35	46.53	46.63	67.46
		TET-TMI(Ours)	**92.93**	**30.99**	**33.21**	**10.94**	**9.86**	**58.56**	**58.32**	**48.32**	**47.45**	**69.51**
	AlexNet	STBP	91.39	22.67	22.68	3.14	3.34	51.33	50.23	33.64	33.78	**64.86**
		STBP-H	**92.14**	22.14	22.12	3.03	3.08	51.35	50.24	32.16	32.21	60.12
		STBP-AIMIE	90.30	22.51	22.18	3.30	3.98	51.40	50.35	34.19	34.23	58.39
		STBP-TMI(Ours)	91.57	**24.63**	**24.34**	**5.72**	**5.53**	**53.89**	**52.38**	**34.91**	**35.75**	63.41
		TET	92.02	27.90	15.79	3.49	3.54	55.08	53.67	44.16	44.18	58.32
		TET-H	92.05	26.51	16.33	3.14	3.26	55.05	53.99	43.67	43.78	58.85
		TET-AIMIE	91.80	26.88	17.68	3.57	3.87	51.15	51.00	39.67	40.52	58.79
		TET-TMI(Ours)	**92.10**	**28.88**	**19.30**	**4.26**	**5.36**	**56.79**	**55.89**	**45.83**	**45.14**	**60.12**

Table 2. Test accuracy (%) comparison of SNN models with/without TMI regularizer on **DVS-CIFAR10 dataset**. The original accuracy of each defense is described in the column "Natural". "-**TMI**" represents embedding our proposed TMI regularizer into training, "-**H**" represents using output entropy [31], "-**AIMIE**" represents calculating mutual information by histogram method [11].

Dataset	Networks	Methods	Natural	white-box				black-box				Gaussian-noise
				FGSM	FGSM-RGA	PGD	PGD-RGA	FGSM	FGSM-RGA	PGD	PGD-RGA	
DVS-CIFAR10	VGGSNN	STBP	78.71	62.10	60.61	32.40	36.90	72.41	71.20	72.50	74.20	7.74
		STBP-H	64.60	50.30	47.60	24.30	23.30	59.78	60.50	51.33	52.80	7.73
		STBP-AIMIE	64.41	51.32	57.80	25.51	23.60	59.73	60.60	51.11	53.20	7.65
		STBP-TMI(Ours)	**77.50**	**63.42**	**65.20**	**34.41**	**37.50**	**75.82**	**75.90**	**73.51**	**73.80**	**8.13**
		TET	67.43	55.50	52.50	32.81	35.10	63.61	64.23	57.58	59.20	6.41
		TET-H	67.41	53.80	55.80	33.59	30.30	62.03	61.91	55.87	57.50	7.07
		TET-AIMIE	64.00	55.23	51.80	35.81	31.90	61.22	63.90	55.20	59.10	6.78
		TET-TMI(Ours)	**67.62**	**57.31**	**57.20**	**38.20**	**37.92**	**64.44**	**65.89**	**59.42**	**60.20**	**7.32**
	AlexNet	STBP	63.39	49.41	53.30	22.52	25.91	56.33	55.30	45.79	46.91	5.80
		STBP-H	**63.48**	48.25	50.21	21.51	27.28	56.82	56.20	45.56	**47.82**	5.55
		STBP-AIMIE	63.31	49.92	53.38	23.10	27.81	57.81	55.30	46.03	47.73	5.63
		STBP-TMI(Ours)	62.11	**50.83**	**54.31**	**26.23**	**28.43**	**58.65**	**57.32**	**47.92**	47.48	**6.22**
		TET	65.11	55.42	55.42	39.19	42.00	61.83	60.41	54.81	52.00	6.11
		TET-H	**65.62**	55.13	55.61	36.71	39.71	61.61	60.62	54.34	49.71	5.50
		TET-AIMIE	65.00	54.32	54.73	37.50	41.92	59.88	59.84	55.13	55.08	5.72
		TET-TMI(Ours)	65.47	**57.63**	**56.22**	**43.29**	**43.65**	**62.63**	**61.24**	**56.31**	**56.63**	**6.91**

3.2. More Performance for Various Attack Types

White box attack. The experimental results of our method on different datasets (i.e., CIFAR-10, DVS-CIFAR10, Tiny-ImageNet) with different training stragetieies when attacked by white box adversarial attack are recorded in Table 1, Table 2, and Table 3, respectively. We can observe, compared with the original method, the training method with our TMI regularizer can effectively improve the robustness of the model. We can observe that when attack by RGA, which is based on the firing rates of the model. Our method can still improve the defense effect of the original model (e.g., Table 3, on Tiny-ImageNet with VGGSNN, when attacked by FGSM-RGA, the defense accuracy of our method (STBP-TMI) is 3.42% while it is 2.34% of original model (STBP)). This may due to our TMI regularizer changing the information transfer on the temporal characteristics, thereby affecting the model's firing rate to further defense RGA attacks.

Black box attack. Table 1, Table 2, and Table 3 also show the model performances on black box attack. We use the same model trained with a different seed to generate black box perturbed images. In black box setting, for all the models on both datasets, we can also observe the superior performance of our TMI regularizer.

Gaussian noise attack. The above robustness analysis mainly focuses on adversarial perturbations. In this part, we evaluate the robustness of the model to common noise (*i.e.*, gaussian noise with mean 0 and variance 1). The experimental results are recorded in Table 1, Table 2, and Table 3. From these tables we can observe, on all datasets, after adding the TMI regularizer, the robustness of SNN to common noise is further improved.

3.3. Effectiveness of Proposed TMI regularizer in SNNs

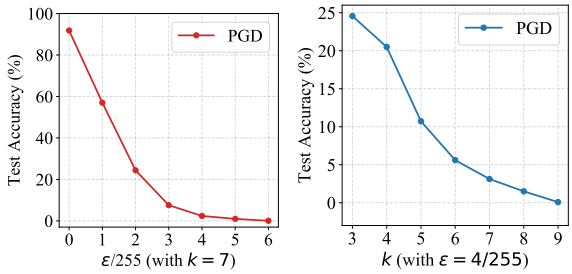
Comparison with AIMIE. AIMIE uses the histogram partition method to calculate mutual information between different spike trains [11]. We compare our TMI regularizer with AIMIE in the same training mechanism in Table 1 on the main paper, where we observe that the robustness improvement of our mutual information is obviously higher than AIMIN in SNN, *e.g.*, when under PGD attack, the robustness improvement of our MI is 0.40% while AIMIE is

Table 3. Test accuracy (%) comparison of SNN models with/without TMI regularizer on **Tiny-ImageNet dataset**. The original accuracy of each defense is described in the column "Natural". "-**TMI**" represents embedding our proposed TMI regularizer into training, "-**H**" represents using output entropy [31], "-**AIMIE**" represents calculating mutual information by histogram method [11].

Dataset	Networks	Methods	Natural	white-box				black-box				Gaussian-noise
				FGSM	FGSM-RGA	PGD	PGD-RGA	FGSM	FGSM-RGA	PGD	PGD-RGA	
Tiny-ImageNet	VGGSNN	STBP	45.19	2.44	2.34	0.10	0.11	16.05	16.01	9.34	9.78	7.74
		STBP-H	45.35	2.73	2.84	0.13	0.10	16.19	16.11	10.17	10.16	7.73
		STBP-AIMIE	45.10	2.65	2.54	0.12	0.11	16.08	16.01	9.31	9.63	7.65
		STBP-TMI(Ours)	**45.81**	**4.26**	**4.12**	**1.85**	**1.28**	**18.32**	**17.65**	**11.19**	**11.07**	**8.13**
		TET	50.06	4.87	4.57	0.31	0.33	16.23	16.15	9.07	9.09	6.40
		TET-H	50.08	4.98	4.87	0.16	0.23	16.26	16.23	9.04	9.17	7.07
		TET-AIMIE	49.82	4.56	4.43	0.13	0.15	16.14	16.03	9.01	9.02	6.78
		TET-TMI(Ours)	**50.63**	**6.38**	**6.23**	**1.86**	**1.34**	**18.67**	**118.45**	**11.32**	**11.04**	**7.30**
	AlexNet	STBP	42.31	1.28	1.01	0.06	0.11	15.39	14.36	10.94	10.35	5.80
		STBP-H	**43.34**	1.35	1.29	0.05	0.11	15.71	14.33	11.34	11.40	5.55
		STBP-AIMIE	42.29	1.32	1.26	0.04	0.07	15.78	14.56	11.31	11.23	5.63
		STBP-TMI(Ours)	42.70	**3.40**	**3.23**	**1.84**	**1.04**	**17.86**	**16.34**	**12.43**	**12.34**	**6.22**
		TET	47.32	3.98	3.48	0.21	0.36	**15.66**	14.32	9.01	9.16	6.11
		TET-H	**48.25**	3.73	3.89	0.22	0.44	15.27	14.02	9.41	9.06	5.50
		TET-AIMIE	47.40	3.72	3.68	0.14	0.15	15.18	14.65	9.23	9.25	5.72
		TET-TMI(Ours)	47.97	**5.67**	**5.14**	**1.16**	**1.03**	15.59	**15.20**	**11.32**	**10.98**	**7.32**

-0.58% on CIFAR-100. The results validate the effectiveness our mutual information calculation method and mutual information implementation method.

The reliability of proposed TMI regularizer. Note that the robustness improvement by TMI regularizer is not caused by the so-called "obfuscated gradients" [4]. This can be verified by four phenomenons [4]: (1) Iterative attack (*e.g.*, PGD) has a higher success rate (lower accuracy) than single-step attack (*e.g.*, FGSM) as shown in Table 1. (2) White box attack performs better compared to black box attack. This can be observed in Table 1. (3) Increasing perturbation bound can increase attack strength, (4) Unbounded attacks can reach $\sim$ 100% success. In Fig. 1a, we analyze VGGSNN on CIFAR-10 with increasing attack bound ϵ when attacked by PGD. As shown in Fig. 1a, the classification accuracy of VGGSNN decreases as we increase ϵ and finally reaches an accuracy of $\sim 0\%$. We also evaluate the VGGSNN performance with increased attack strength by increasing the number of iterations K of PGD, and find that the model's robustness decreases with increasing k as shown in Fig. 1b. The above analysis verifies that the TMI regularizer improves the robustness of SNN effectively and reliably rather than gradient obfuscation.



(a) (b)

Figure 1. The performance of VGGSNN trained by STBP-TMI when attacked by PGD. The dataset is CIFAR-10, the network is VGGSNN. (a) Under different perturbations ϵ. (b) Under different iterative steps k.

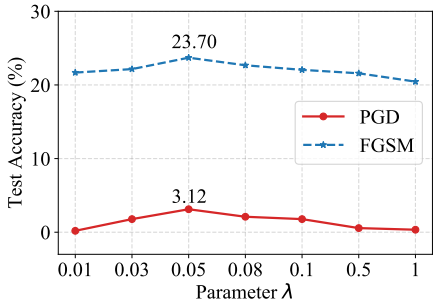


Figure 2. The sensitivity to regularization parameter λ. The dataset is CIFAR-10, and the network is VGGSNN.

3.4. More Ablation Study

3.4.1. Sensitivity to regularization parameter λ.

We investigate the regularization parameter λ which controls the strength of the regularization. We present the results in Fig. 2 for different $\lambda \in [0.01, 1]$. As Fig. 2 shown, we choose $\lambda = 0.05$ for further experiments.

3.4.2. Sensitivity to KDE bandwidth.

We fix the bandwidth to 0.4 to match the normalized spike activity range [0, 1] and to align with common practice in density estimation. It produced stable MI across datasets and architectures. We now vary the bandwidth in [0.2, 0.4, 0.6, 0.8]. Both accuracy [72.89, 72.83, 72.96, 71.45] and robustness [12.23, 12.46, 11.34, 11.12] change smoothly, indicating low sensitivity to bandwidth.

References

[1] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2897–2905, 2018. 1

[2] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin

Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017. 1

[3] Rana Ali Amjad and Bernhard C Geiger. Learning representations for neural network-based classification using the information bottleneck principle. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2225–2239, 2019. 1

[4] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proc. of International Conference on Machine Learning*, pages 274–283. PMLR, 2018. 4

[5] Modeste Atsague, Olukorede Fakorede, and Jin Tian. A mutual information regularization for adversarial training. In *Asian Conference on Machine Learning*, pages 188–203. PMLR, 2021. 1

[6] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *Proc. of International Conference on Machine Learning*, pages 531–540. PMLR, 2018. 1

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

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

[9] Jianhao Ding, Tong Bu, Zhaofei Yu, Tiejun Huang, and Jian Liu. SNN-RAT: Robustness-enhanced spiking neural network through regularized adversarial training. *Advances in Neural Information Processing Systems*, 35:24780–24793, 2022. 1

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2

[11] Ekaterina D Gribkova, Baher A Ibrahim, and Daniel A Llano. A novel mutual information estimator to measure spike train correlations in a model thalamocortical network. *Journal of neurophysiology*, 120(6):2730–2744, 2018. 3, 4

[12] Youngeun Kim, Yuhang Li, Hyoungseob Park, Yeshwanth Venkatesha, Anna Hambitzer, and Priyadarshini Panda. Exploring temporal information dynamics in spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8308–8316, 2023. 1

[13] Artemy Kolchinsky, Brendan D Tracey, and David H Wolpert. Nonlinear information bottleneck. *Entropy*, 21(12): 1181, 2019. 1

[14] Yunan Li, Huizhou Chen, Guanwen Feng, and Qiguang Miao. Learning robust representations with information bottleneck and memory network for rgb-d-based gesture recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20968–20978, 2023. 1

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 2

[16] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36 (6):51–63, 2019. 1

[17] Xue Bin Peng, Angjoo Kanazawa, Sam Toyer, Pieter Abbeel, and Sergey Levine. Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow. *arXiv preprint arXiv:1810.00821*, 2018. 1

[18] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *Proc. of International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019. 1

[19] Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019. 1

[20] Ohad Shamir, Sivan Sabato, and Naftali Tishby. Learning and generalization with the information bottleneck. *Theoretical Computer Science*, 411(29-30):2696–2711, 2010. 1

[21] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017. 1

[22] Xudong Tian, Zhizhong Zhang, Shaohui Lin, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Farewell to mutual information: Variational distillation for cross-modal person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1522–1531, 2021. 1

[23] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *IEEE information theory workshop (itw)*, pages 1–5. IEEE, 2015. 1

[24] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 1

[25] Haoqing Wang, Xun Guo, Zhi-Hong Deng, and Yan Lu. Rethinking minimal sufficient representation in contrastive learning. In *Proc. of Computer Vision and Pattern Recognition*, pages 16041–16050, 2022. 1

[26] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018. 1, 2

[27] Mengting Xu, Tao Zhang, Zhongnian Li, and Daoqiang Zhang. InfoAT: Improving adversarial training using the information bottleneck principle. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):1255–1264, 2022. 1

[28] Jini Yang, Beomseok Oh, Seungryong Kim, and Sunok Kim. Spikematch: Semi-supervised learning with temporal dynamics of spiking neural networks. *arXiv preprint arXiv:2509.22581*, 2025. 1

[29] Penglong Zhai and Shihua Zhang. Adversarial information bottleneck. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):221–230, 2022. 1

[30] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. *Advances in Neural Information Processing Systems*, 33:14435–14447, 2020. 1

[31] Dawei Zhou, Nannan Wang, Xinbo Gao, Bo Han, Xiaoyu Wang, Yibing Zhan, and Tongliang Liu. Improving adversarial robustness via mutual information estimation. In *Proc. of International Conference on Machine Learning*, pages 27338–27352. PMLR, 2022. 1, 3, 4