Rethinking Feature Uncertainty in Stochastic Neural Networks for Adversarial Robustness

Hao Yang\textsuperscript{1}, Min Wang\textsuperscript{1}, Zhengfei Yu\textsuperscript{1}, Yun Zhou\textsuperscript{1,*}
\textsuperscript{1}National university of Defense Technology
\{yanghao,wangminwm,yuzhengfei19, zhouyun\}@nudt.edu.cn

Abstract
It is well-known that deep neural networks (DNNs) have shown remarkable success in many fields. However, when adding an imperceptible magnitude perturbation on the model input, the model performance might get rapid decrease. To address this issue, a randomness technique has been proposed recently, named Stochastic Neural Networks (SNNs). Specifically, SNNs inject randomness into the model to defend against unseen attacks and improve the adversarial robustness. However, existed studies on SNNs mainly focus on injecting fixed or learnable noises to model weights/activations. In this paper, we find that the existed SNNs performances are largely bottlenecked by the feature representation ability. Surprisingly, simply maximizing the variance per dimension of the feature distribution leads to a considerable boost beyond all previous methods, which we named maximize feature distribution variance stochastic neural network (MFDV-SNN). Extensive experiments on well-known white- and black-box attacks show that MFDV-SNN achieves a significant improvement over existing methods, which indicates that it is a simple but effective method to improve model robustness.

1 Introduction
Deep learning models have shown promising results in many fields. However, a small perturbation on the input, which is hard to identify by human eye, can subvert the model’s predictions easily. The perturbed data is called the adversarial example [Goodfellow et al., 2015]. This phenomenon that attack leads to a decrease in model performance has attracted many researchers and security service organizations to build safer and more secure models.

Among the recently developed defense methods, Stochastic Neural Networks (SNNs) have shown great potentials for building robust models. As we all know, stochastic noise helps models avoid getting trapped into local minima during training phases [Hopfield, 1982; Srivastava et al., 2014]. Within the aforementioned characteristics of stochastic noise, SNNs transform deterministic models into stochastic models by injecting randomness noises into model activations/weights. Within more uncertainty, the model can explore more profound network representation. When the model converges, the uncertainty of the noise will force the model to be in a stable state with relatively locally optimal. Thus, it enhances robustness of the model.

Inspired by the above analysis, a natural idea was came up: Does larger uncertainty lead to higher robustness? In this paper, we construct the SNN model via injecting isotropic noise into the feature layer, and further explore the feature uncertainty for adversarial robustness in SNNs. It is exciting that simply maximizing per dimension of the feature distribution brings huge model robustness. It is also worth emphasizing that currently most defense methods are based on adversarial training - a data argumentation method that uses adversarial examples generated by the attack method to retrain the networks [Kurakin et al., 2017; Madry et al., 2018]. Although adversarial training is simple and proved effective for improving model robustness, it significantly leads to higher computational costs and more training time. Moreover, mixed clean examples and adversarial examples might “obfuscate” the clean data gradient information. Thus, improving adversarial robustness is at the expense of clean data accuracy.

On the contrary, our model does not require adversarial training to achieve robustness, which means it does not require high computation costs and degrade the clean data accuracy. To sum up, our contributions are as follows:

- We propose a simple and efficient stochastic neural network which maximize the feature distribution variance (MFDV-SNN) for improving adversarial robustness.
- To the best of our knowledge, we are the first to explore the effect of feature uncertainty in SNNs on adversarial robustness.
- The proposed method does not require adversarial training and can maintain the strong representational capability while requiring only a low computational cost.
- Extensive experiments on white- and black-box attacks
show our proposed method is compelling.

2 Related Work

2.1 Adversarial Attack

Many methods have been proposed to attack deep learning models. Researchers further divided attack methods into white- and black-box attacks according to whether they can obtain the gradient information.

White-box attack: White-box attacks mean that the attacker knows the model gradient information. A simple yet effective white-box attack method is called Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2015], which adds a small perturbation in the direction of the sign of the gradient updates. It can be formulated as

$$\tilde{x}' = \tilde{x} + \epsilon \cdot \text{sgn}(\nabla_{\tilde{x}}L(h(\tilde{x}), y)) \quad (1)$$

where $\tilde{x}$ denotes the input image, $\epsilon$ denotes the perturbation strength, $L$ denotes the loss function and $h(\cdot)$ denotes the target model. Kurakin further update the one-step attack FGSM to multi-step attack named Basic Iterative Method (BIM) [Kurakin et al., 2017]. Compared to FGSM, BIM uses a smaller step size to explore the possible adversarial direction. Furtherly, Madry updates BIM by random initializing the input point, which is one of the most strongest first-order attack named Projected Gradient Descent (PGD) [Madry et al., 2018]. It can be formulated as

$$\tilde{x}_{t+1}^{\text{PGD}} = \Pi_{x' \in B_s} (\tilde{x}' + \alpha \cdot \text{sgn}(\nabla_{\tilde{x}}L(h(\tilde{x}'), y))) \quad (2)$$

where $\Pi_{x' \in B_s}$ is the projection operation which force the adversarial example in the $L_p$ ball $s$ around $\tilde{x}$, and $\alpha$ is the step size. Another strong first-order attack algorithm is called CW attack [Carlini and Wagner, 2017] which finds adversarial example by solving the following optimization function formulated as

$$\min{\|\delta\|_p + c \cdot h(\tilde{x} + \delta)} \text{ s.t. } \tilde{x} + \delta \in [0, 1]^n \quad (3)$$

where $p$ is the norm distance, commonly choosing from $\{0, 2, \infty\}$.

Black-box attack: Unlike white-box attacks, black-box attackers can only access the model through queries. There are mainly two ways to fool a model. One is to train a substitute of the model [Papernot et al., 2017], in which attackers query from the target model and generate a synthesized dataset with input and the corresponding output. Due to the transferability of adversarial examples, attackers can attack alternative models and target models. The limitation of this method is that it cannot execute multiple queries in reality. The other is to estimate the gradients via multiple queries to the targeted model [Su et al., 2019]. Among them, zero-order optimization [Chen et al., 2017] algorithms aim to estimate the gradients of the target model directly.

2.2 Stochastic Defense

Nowadays, SNNs have shown promising results via injecting fixed or learnable noise into model activations/weights. RSE [Liu et al., 2018] uses ensemble tricks to improve model robustness. Specifically, they inject Gaussian noise to multi-layers during training and then perform multiple forward passes to test it. It means they only need training one time and can be viewed as an ensemble model. RSE mainly uses fixed variance by hand-tuned. Parameter noise injection (PNI) [He et al., 2019] further proposed to learn a sensitive parameter to control the variance. Moreover, L2P [Jeddi et al., 2020] updates PNI by alternating training a noise module and a neural network module, which they called “alternating back-propagation.”

2.3 Connection to Similar Work

In this section, we mainly discuss some similar but different works to clarify the contribution of our work. The most related work is proposed by Yu [Yu et al., 2021], in which they propose to use a max-margin entropy loss to regularize the feature distribution. However, we need to highlight some main differences between them. Firstly, the construct way of the Gaussian layer is different. In their work, a parallel Gaussian mean and Gaussian variance may meet mistakes when increasing the Gaussian variance. In this situation, the Gaussian mean and variance share the same parameters on the feature layer but different optimization ambitions which may pass on obfuscate gradient information to the last feature layer matrix through back-propagation. Secondly, the margin $b$ in their paper largely restricted the model representation ability, and it means the authors given a solid prior information which can not ensure rationality. A similar situation can be seen in the deep variational information bottleneck (VIB) [Alemi et al., 2017] model that uses a definite Gaussian variance, which also restricts the representation ability for model robustness.

To sum up, while previous researches often use fixed feature uncertainty, we highlight that we are the first to explore the relationship between feature uncertainty and model robustness. The key finding of our research is that larger uncertainty will bring higher robustness. That is why the proposed method MFDV-SNN achieves state-of-the-art results compared to previous various SNN-based models. Moreover, the proposed method is simple and efficient, motivating researchers to rethink the feature uncertainty in SNNs for adversarial robustness.

In practice, we only need to maximize the unbounded feature distribution variance to achieve significant improvement. The unbounded high variance will be self-adaptive to the model architecture and dataset and then explore a more stable representation. We also need to claim that the unbounded variance will not collapse since the gradient is $-\frac{1}{\sigma}$, the overlarge variance will not back-propagation gradient information to update the network parameter. Extensive experiments on white- and black-box attacks confirm that the proposed key point is important enough.

3 Methodology

In this section, we introduce in detail the implementation process of the proposed MFDV-SNN. The critical point is shown in Figure 1.

3.1 Stochastic layer

As illustrated in Figure 1, we take the last three-layer as a detailed description. The data passes through a neural net-
work to get $h_l$, then $h_l$ pass through a layer which is usually a linear layer and gets the feature extractor $h_{l+1}$. We do not directly build a parallel Gaussian mean and Gaussian variance layer as [Yu et al., 2021] since increasing the variance may obfuscate the original feature layer. Instead, we keep the original feature plus a maximum variance Gaussian distribution, where the Gaussian mean is a zero matrix. Thus, the $h_{l+1}$ can not only maintain the original feature representation ability but also obtain uncertainty via injected noise. The idea behind this is that the original feature can preserve the data manifold well for downstream tasks such as image classification. However, it is not appropriate within the adversarial setting, as a large proportion of information is redundant and even harmful to model robustness. Thus, we need considerable uncertainty to explore a more robustness representation. What is more, we use a non-information prior to initializing the Gaussian variance. In practice, we sample the same dimension with the feature dimension from the uniform distribution. Then, we establish a Gaussian distribution $z$, and sample from the Gaussian distribution plus with the original hidden representation $h_{l+1}$. Finally, get the logit of the network.

3.2 Loss Function
As discussed above, we need to maximize the feature uncertainty during model training. The simplest implementation is to add the built Gaussian variance to the loss function. It needs to emphasize that we do not directly assign a parametric variance, e.g., var = 20. One reason is that this will hinder the model convergence, and it may lead to model collapse at the training beginning. Another reason is that we do not know the deterministic uncertainty that model needs, which is relevant to model architecture and the datasets.

Instead, we initialize the Gaussian variance with the uniform distribution values from zero to one. As the network is trained, the variance increases gradually at small steps $\Delta$. The optimization process will iterate dynamically, which means that once the model converges to local minima, greater uncertainty will force it to explore a more stable state. Finally, it will adapt to a more robust state automatically.

Thus, the loss function can be formulated as

$$\mathcal{L} = \mathcal{L}_C - \lambda_1 \sum_{i=1}^{D} \ln (\tilde{\sigma}_i + \Delta_i) + \lambda_2 \tilde{u}^T \tilde{w}$$  \hspace{1cm} (4)$$

where $\mathcal{L}_C$ is the cross-entropy loss, and $D$ denotes the feature dimension of the penultimate layer. While adding all the dimensions of the variance simply may meet our requirements, we take the $\ln (\tilde{\sigma}_i)$ operation. It has two advantages. One is to facilitate the derivative calculation of the gradient. The other is to slow down the numerical change of variance. In particular, although we emphasize that the network will not collapse, the sudden and significant change in the variance value at the beginning of the network training is not conducive to the back propagation of the gradient. The $\ln (\tilde{\sigma}_i)$ operation can be used as an active protection measure. The last item is $L_2$ regularization to penalty weights over-large, $\lambda_1$ controls the power of variance, $\lambda_2$ controls the power of weights penalty.

4 Experiments

4.1 Datasets & Adversarial Attacks
Three well-known datasets are used in our experiments: SVHN, CIFAR-10, and CIFAR-100. The SVHN dataset consists of 73K training data and 26K testing data of size 32x32x3 with ten classes. The CIFAR-10 dataset consists of 50K training data and 10K testing data of size 32x32x3 with ten classes. For the CIFAR-100 dataset, the size of training data and testing data is the same as CIFAR-10, but with one-hundred classes. The attack algorithms in our experiments contain white- and black-box attacks. For white-box attacks, we use FGSM, PGD$_{10}$, C&W, and stronger PGD$_{100}$ to evaluate the efficiency of the proposed MFDV-SNN. For black-box attacks, n-Pixel attacks and more powerful Square attack are used in our experiment.

4.2 Experimental Setup
The main backbone in our experiment is ResNet-18 [He et al., 2016], the attacks follows [He et al., 2019; Jeddi et al., 2020; Eustratiadis et al., 2021]. In Table 1 and Table 2, the attack strength $\epsilon$ of FGSM and PGD is $8/255$. For PGD attack, the steps we set $k = 10$ and the $\alpha$ we set to $\epsilon/10$, which follows [He et al., 2019; Eustratiadis et al., 2021]. For C&W attack, the learning rate we set $\alpha = 5 \cdot 10^{-4}$, the number of iterations $k = 1000$, initial constant $c = 10^{-3}$ and maximum binary steps $b_{\text{max}} = 9$ same as [Jeddi et al., 2020; Eustratiadis et al., 2021]. For the n-Pixel attack, we set the population size $N = 400$ and maximum number $b_{\text{max}} = 75$ same as [Jeddi et al., 2020], and we conduct a stronger 5-pixel attack which is not implemented in their setting. For Square attack, we refer to the implementation from [Croce and Hein, 2020]. All experiments are performed on the Pytorch platform of version 1.7.0.

4.3 Comparison to Prior Stochastic Defenses
In this section, we list the comparative stochastic defense methods. Adv-BNN [Liu et al., 2019]: A combination of Bayesian neural network with adversarial training. PNI [He et al., 2019]: Injecting Gaussian noise to multi-layers. L2P [Jeddi et al., 2020]: Updating PNI by learning a perturbation injection module and alternating training the noise and network module. SE-SNN [Yu et al., 2021]: Introducing a margin entropy loss. What is more, there are partial comparisons against WCA-Net [Eustratiadis et al., 2021] and IAAT [Xie et al., 2019].
## Against White-box Attacks.

In this section, we mainly focus on evaluating the proposed MFDV-SNN against white-box attacks. In Table 1, we compare the state-of-the-art SNNs for FGSM and PGD attacks on CIFAR-10 with a ResNet-18 backbone. The results show that the proposed MFDV-SNN outperforms all comparison algorithms. It is worth noticing that most relevant SE-SNN have not released the official code and the result of ResNet-18. Figure 2(a) in their paper [Yu et al., 2021] shows that SE-SNN is not so strong that the accuracy decreases so fast even under the weak FGSM attack. For a fair comparison, we extract the result from [Eustratiadis et al., 2021]. The results show that we have about 10.4% and 11.5% improvement compared with the previous best defense method WCA-Net. Compared to SE-SNN, for the FGSM attack, we have about 15.3% improvement. Compared with other recent state-of-the-art stochastic defense methods, the proposed MFDV-SNN exceeds them significantly. What is more, Adv-BNN, PNI, and L2P for contrast involve adversarial training. However, this operation improves the model’s robustness at the expense of the accuracy of clean data. Instead, the proposed MFDV-SNN need not adversarial training, which means that our method need lower computation cost and lower training time. We can see from Table 1 that we achieve the highest accuracy on clean data, which means the more profound network optimization point explored is conducive for neural networks on the traditional classification task.

For Table 2, we choose a larger dataset CIFAR-100 with one-hundred classes. The results show that the proposed MFDV-SNN still outperforms the defense methods for contrast, and the accuracy of clean data is also the highest. For the FGSM attack, compared with the best stochastic defense methods Adv-BNN and L2P, we have an improvement of about 36.3%. For PGD attack, compared with the best defense method Adv-BNN, we have about 27.6% improvement.

### Table 1: Comparison of state-of-the-art SNNs for FGSM and PGD attacks on CIFAR-10 with a ResNet-18 backbone.

| Method          | Clean | FGSM  | PGD  |
|-----------------|-------|-------|------|
| Adv-BNN [Liu et al., 2019] | 82.2  | 60.0  | 53.6 |
| PNI [He et al., 2019]        | 87.2  | 58.1  | 49.4 |
| L2P [Jeddi et al., 2020]    | 85.3  | 62.4  | 56.1 |
| SE-SNN [Yu et al., 2021]    | 92.3  | 74.3  |     |
| WCA-Net [Eustratiadis et al., 2021] | 93.2  | 77.6  | 71.4 |
| MFDV-SNN (Ours)            | 93.7  | 85.7  | 79.6 |

### Table 2: Comparison of state-of-the-art SNNs for FGSM and PGD attacks on CIFAR-100 with a ResNet-18 backbone.

| Method          | Clean | FGSM  | PGD  |
|-----------------|-------|-------|------|
| Adv-BNN [Liu et al., 2019] | 58.0  | 30.0  | 27.0 |
| PNI [He et al., 2019]        | 61.0  | 27.0  | 22.0 |
| L2P [Jeddi et al., 2020]    | 50.0  | 30.0  | 26.0 |
| IAAT [Xie et al., 2019]     | 63.9  | -    | 18.5 |
| MFDV-SNN (Ours)            | 69.4  | 47.1  | 37.3 |

## Against Black-box Attacks.

In this section, we evaluate the proposed MFDV-SNN on the well-known black-box n-Pixel attack. This attack relies on evolutionary optimization and is derivative-free. The attack strength is controlled by the number of pixels it comprises. From the results in Table 4, we can see that the proposed MFDV-SNN outperforms other state-of-the-art methods in all attack strengths. More specifically, for 1−, 2−, 3−pixel attack, compared with the best defense method Adv-BNN, the proposed MFDV-SNN has improvement about 24.5%, 27.3%, respectively. We add a more potent 5-pixel attack to evaluate the proposed MFDV-SNN’s efficiency further.

### Table 3: Comparison of state-of-the-art SNNs for white-box C&W attack on CIFAR-10 with a ResNet-18 backbone.

| Method   | Clean | FGSM  | PGD  |
|----------|-------|-------|------|
| Adv-BNN  | 82.2  | 87.2  | 85.3 |
| PNI      |       |       |      |
| L2P      |       |       |      |
| MFDV-SNN | 93.7  | 85.3  |      |

### Table 4: Comparison of state-of-the-art methods for black box n-Pixel attack on CIFAR-10 with a ResNet-18 backbone.

| Strength | Adv-BNN | PNI | L2P | MFDV-SNN |
|----------|---------|-----|-----|----------|
| Clean    | 82.2    | 87.2 | 85.3 | 93.7     |
| k=0      | 78.9    | 66.9 | 83.6 | 88.8     |
| k=0.1    | 78.1    | 66.1 | 84   | 87.9     |
| k=1      | 65.1    | 34   | 76.4 | 87.2     |
| k=2      | 49.1    | 16   | 66.5 | 86.6     |
| k=5      | 16      | 0.08 | 34.8 | 83.5     |

For Table 3, the attack algorithm follows foolbox1, a public attack library, to evaluate the efficiency of the proposed MFDV-SNN. The results show that the proposed MFDV-SNN significantly exceeds the existed stochastic defense methods even when the confidence level $k$ is 5. Overall, although the implementation of the proposed MFDV-SNN is simple, the improvements are significant.

### Stronger Attacks.

In this section, we conduct two stronger white- and black-box attacks: PGD$^{100}$ and Square attack. For White-box PGD$^{100}$ attack, the iterative steps are set as 100, which means it uses a smaller step size to explore the adversarial example. The black-box Square attack comprises the attacked data in minor localized square-shaped updates. The results are shown in Table 5.

### 4.4 Comparison to State-of-the-Art

We compare MFDV-SNN with several state-of-the-art defense methods proposed in recent years. Among them, some methods are SNNs, and some are not. We present the results in Table 6. For a fair comparison, some results are extracted from the original paper and [Eustratiadis et al., 2021]. All experiments are conducted on CIFAR-10 with the PGD attack.

1https://github.com/bethgelab/foolbox
Following [He et al., 2019], the prerequisite is that the gradient-based (e.g., PGD and FGSM) attack cannot find adversarial examples. However, Figure 2 (right) indicates that when we increase the distortion bound, our method can still be broken.

**Criterion 5:** Increasing the distortion bound does not increase the success rate.

**Refutation:** The experiment in Figure 2 (left) shows that increasing the distortion bound increases the attack success rate.

**EOT-attack:** Athalye [Athalye et al., 2018] further claimed that the false gradient could not protect the model well when the attackers use the gradient, which is the expectation over a series of transformations. Following [Jeddi et al., 2020; Pinot et al., 2019], we use a Monte Carlo method which expects the gradient over 80 simulations of different transformations on the CIFAR-10 dataset with the backbone ResNet-18. Results show that PNI, Adv-BNN, L2P can have 79.2% robustness, which outperforms the compared methods and confirms that the proposed MFDV is not of gradient obfuscation. To ensure the efficiency of the proposed MFDV-SNN result is not a stochastic gradient, we adopt 15 MC sampling at the test phase in our experiments, which number we find in experiment is stable enough.

**4.6 Inspection of Generalization**

In this section, many experiments are conducted to evaluate the generalization of the proposed model. More specifically, we first explore the dataset size influence on the proposed MFDV-SNN, as shown in Table 7. Three different sizes are used in this experiment. **SVHN:** a relatively small dataset. **CIFAR-10:** a medium dataset with 60k training data and 10k testing data. **CIFAR-100:** a large dataset with one-hundred classes. The experiments are based on the backbone ResNet-18. The results show that MFDV-SNN has a great generalization to different dataset sizes.

For Table 8 and Table 9, we mainly explore the impact of the network architecture on the proposed MFDV-SNN. More specifically, in practice, we explore how the depth and breadth of the network affect our method. **ResNet-18:** a common used backbone in stochastic defense. **ResNet-50:** a deeper ResNet architecture than ResNet-18. **WRN-34-10:**
Table 7: Generalization study for FGSM and PGD attacks on various datasets: SVHN, CIFAR-10 and CIFAR-100. In which we use ResNet-18 as a backbone.

| Dataset      | Model       | Clean | FGSM | PGD | Clean | FGSM | PGD | Clean | FGSM | PGD |
|--------------|-------------|-------|------|-----|-------|------|-----|-------|------|-----|
| SVHN         | No Defense  | 94.9  | 18.6 | 5.9 | 92.9  | 21.3 | 2.3 | 68.8  | 12.8 | 1.5 |
|              | MFDV-SNN    | 94.0  | 86.1 | 82.7 | 93.7  | 85.7 | 79.6 | 69.4  | 47.1 | 37.3 |
| CIFAR-10     | No Defense  | 68.5  | 49.4 | 41.7 | 67.8  | 39.8 | 31.2 | 56.3  | 28.1 | 19.1 |
|              | MFDV-SNN    | 69.8  | 53.1 | 47.7 | 70.7  | 45.3 | 38.4 | 59.6  | 32.4 | 23.5 |
| CIFAR-100    | No Defense  | 72.0  | 48.5 | 41.2 | 70.5  | 43.7 | 36.5 | 59.2  | 32.0 | 22.9 |
|              | MFDV-SNN    | 74.1  | 53.7 | 47.8 | 73.0  | 48.6 | 39.6 | 61.6  | 34.5 | 25.5 |

Table 8: Generalization study for PGD attacks on various architectures: ResNet-18, ResNet-50 and WRN-34-10. In which we use CIFAR-10 dataset as a backbone.

| Architecture | PGD(\(\epsilon/255\)) | Clean | 1   | 2   | 4   | 8   | 16  | 32  | 64  | 128 |
|--------------|-------------------------|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| ResNet-18    | No Defense              | 68.5  | 32.4 | 15.5 | 5.2 | 1.5 | 0.2 | 0.0 | 0.0 | 0.0 |
|              | MFDV-SNN                | 69.5  | 51.6 | 47.0 | 43.0 | 37.3 | 24.1 | 8.9 | 1.8 | 0.4 |
| ResNet-50    | No Defense              | 71.4  | 26.2 | 12.1 | 4.6 | 1.7 | 0.4 | 0.0 | 0.0 | 0.0 |
|              | MFDV-SNN                | 70.6  | 46.2 | 40.6 | 35.5 | 25.5 | 12.9 | 3.0 | 0.5 | 0.0 |
| WRN-34-10    | No Defense              | 72.0  | 24.7 | 8.5  | 2.0 | 1.0 | 0.6 | 0.0 | 0.0 | 0.0 |
|              | MFDV-SNN                | 71.2  | 54.3 | 51.7 | 48.5 | 42.7 | 28.1 | 9.8 | 1.4 | 0.2 |

Table 9: Generalization study for PGD attacks on various architectures: ResNet-18, ResNet-50 and WRN-34-10. In which we use CIFAR-100 dataset as a backbone.

| Architecture | PGD(\(\epsilon/255\)) | Clean | 1   | 2   | 4   | 8   | 16  | 32  | 64  | 128 |
|--------------|-------------------------|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| ResNet-18    | No Defense              | 68.5  | 32.4 | 15.5 | 5.2 | 1.5 | 0.2 | 0.0 | 0.0 | 0.0 |
|              | MFDV-SNN                | 69.5  | 51.6 | 47.0 | 43.0 | 37.3 | 24.1 | 8.9 | 1.8 | 0.4 |
| ResNet-50    | No Defense              | 71.4  | 26.2 | 12.1 | 4.6 | 1.7 | 0.4 | 0.0 | 0.0 | 0.0 |
|              | MFDV-SNN                | 70.6  | 46.2 | 40.6 | 35.5 | 25.5 | 12.9 | 3.0 | 0.5 | 0.0 |
| WRN-34-10    | No Defense              | 72.0  | 24.7 | 8.5  | 2.0 | 1.0 | 0.6 | 0.0 | 0.0 | 0.0 |
|              | MFDV-SNN                | 71.2  | 54.3 | 51.7 | 48.5 | 42.7 | 28.1 | 9.8 | 1.4 | 0.2 |

4.7 Ablation Study of MFDV-SNN

As shown in Figure 3, we conduct experiments to check the efficiency of the proposed MFDV-SNN more abundantly, we used both CIFAR-10 and CIFAR-100 datasets in our experiments. The results show that our proposed MFDV-SNN always maintains superior performance.

4.8 TSNE Visualization

We visualize the classification result on the CIFAR-10 dataset trained on ResNet-18, as shown in Figure 4. In practice, we sample 1000 data and visualize them. The visualization results show that the proposed MFDV-SNN learns a more robust architecture that achieves intra-class compactness and performs better even in inter-class separation. It is proved that the proposed method, with unbounded high variance, can maintain high uncertainty and adaptively learn a deeper network representation. The uncertainty will also help the network avoid local optima and explore the global optima, thereby improving the robustness of the model and even the classification ability.

5 Conclusion

In this paper, we propose a simple stochastic neural network named Maximizes Feature Distribution Variance (MFDV-SNN), which significantly exceeds the existing state-of-the-art defense algorithms. Specifically, we build the feature layer to a non-informative unbounded Gaussian distribution that maximizes the Gaussian variance during model training. The proposed method does not rely on adversarial training. It has lower computation costs and training time than adversarial training-based algorithms. Extensive experiments on white- and black-box attacks show that MFDV-SNN achieves considerable performance gains and outperforms existing methods, which motivates researchers to rethink feature uncertainty for adversarial robustness in future research.

Figure 3: Ablation study on CIFAR-10 dataset with the backbone ResNet-18. FGSM attack (Left). PGD attack (Right).

Figure 4: T-SNE visualization of classification result on CIFAR-10 trained on ResNet-18. No defense (Left). MFDV-SNN (Right).
References

[Addepalli et al., 2020] Sravanti Addepalli, Vivek B. S., Arya Baburaj, Gaurang Sriramanan, and R. Venkatesh Babu. Towards achieving adversarial robustness by enforcing feature consistency across bit planes. In CVPR, 2020.

[Alemi et al., 2017] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In ICLR, 2017.

[Athalye et al., 2018] Anish Athalye, Nichola Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In ICML, 2018.

[Carlini and Wagner, 2017] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In SP, 2017.

[Chen et al., 2017] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In ACM, 2017.

[Croce and Hein, 2020] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In ICML, 2020.

[Eustratiadis et al., 2021] Panagiotis Eustratiadis, Henry Gouk, Da Li, and Timothy Hospedales. Weight-covariance ensemble of diverse parameter-free attacks. In ICML, 2021.

[Goodfellow et al., 2015] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In ICLR, 2015.

[He et al., 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

[He et al., 2019] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In CVPR, 2019.

[Hopfield, 1982] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences, 1982.

[Jeddi et al., 2020] Ahmadreza Jeddi, Mohammad Javad Shafiee, Michelle Karg, Christian Scharfenberger, and Alexander Wong. Learn2perturb: An end-to-end feature perturbation learning to improve adversarial robustness. In CVPR, 2020.

[Kurakin et al., 2017] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In ICLR, 2017.

[Lécuyer et al., 2019] Mathias Lécuyer, Vaggelis Atidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In SP, 2019.

[Liu et al., 2018] Xuanqing Liu, Minhai Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In ECCV, 2018.

[Liu et al., 2019] Xuanqing Liu, Yao Li, Chongrui Wu, and Cho-Jui Hsieh. Adv-bnn: Improved adversarial defense through robust bayesian neural network. In ICLR, 2019.

[Madry et al., 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In ICLR, 2018.

[Mustafa et al., 2019] Aamir Mustafa, Salman H. Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Adversarial defense by restricting the hidden space of deep neural networks. In ICCV, 2019.

[Pang et al., 2020] Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. In ICLR, 2020.

[Papernot et al., 2017] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In ACM, 2017.

[Pinot et al., 2019] Rafael Pinot, Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cédric Gouy-Pailler, and Jamal Atif. Theoretical evidence for adversarial robustness through randomization. In NIPS, 2019.

[S. and Babu, 2020] Vivek B. S. and R. Venkatesh Babu. Single-step adversarial training with dropout scheduling. In CVPR, 2020.

[Song et al., 2020] Chuanbiao Song, Kun He, Jiadong Lin, Liwei Wang, and John E. Hopcroft. Robust local features for improving the generalization of adversarial training. In ICLR, 2020.

[Srivastava et al., 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. The journal of machine learning research, 2014.

[Su et al., 2019] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput., 2019.

[Wang et al., 2020] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In ICLR, 2020.

[Xie et al., 2019] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In CVPR, 2019.

[Yu et al., 2021] Tianyuan Yu, Yongxin Yang, Da Li, Timothy Hospedales, and Tao Xiang. Simple and effective stochastic neural networks. In AAAI, 2021.

[Zhang et al., 2019] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In ICML, 2019.