

# Supplementary Materials: Maximizing Feature Distribution Variance for Robust Neural Networks

Anonymous Authors

## 1 ALGORITHM OF MFDV-SNN

The algorithm of proposed MFDV-SNN is shown below.

---

**Algorithm 1** : MFDV-SNN Training.

---

**Require:**

Dataset  $\mathcal{X}$  with batch size  $B$ , label  $y$ , hyper-parameter  $\lambda_1, \lambda_2$ , feature dimension  $D$ , noise parameter  $\epsilon$ .

**Ensure:**

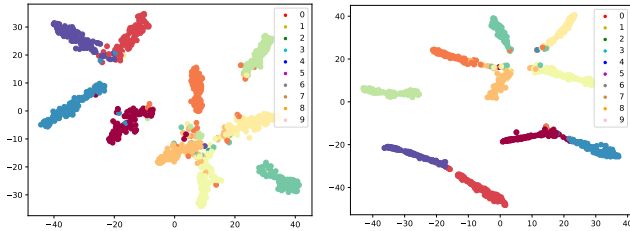
Trained model  $\mathcal{M}(\cdot; \theta)$  with parameters  $\theta$ ;

- 1: Initialize  $\theta$ ;
  - 2: **for**  $epoch = 1$  to  $E$  **do**
  - 3:   **for** each minibatch  $B$  **do**
  - 4:     Extract features  $F$  using any NN architectures;
  - 5:     Sample isotropic Gaussian noise  $z \sim \mathcal{N}(0, \Sigma)$ ;
  - 6:     Construct feature distribution  $h_{l+1} = h_l + z$ ;
  - 7:     Sample from the construct distribution;
  - 8:      $\theta \leftarrow \mathcal{L}_C - \lambda_1 \sum_{i=1}^D \ln(\bar{\sigma}_i) + \lambda_2 \tilde{w}^T \tilde{w}$ ;
  - 9:   **end for**
  - 10: **end for**
  - 11: **return**  $\theta$
- 

## 3 MODULE IMPLEMENTATION IN PYTHON

To show implementation details of *MFDV-SNN*, we attach it core Python code in the following. The complete code will be released after the final decision.

```
1 import torch
2 import torch.nn as nn
3 import torch.nn.functional as F
4 from torch.distributions.normal import Normal
5 class ResNet18_Stochastic(nn.Module):
6     """ Zero mean, trainable variance. """
7     def __init__(self, D, disable_noise=False):
8         super().__init__()
9         self.gen = GeneratorResNet18()
10        self.fc1=nn.Linear(512,D)
11        self.sigma = nn.Parameter(torch.rand(D),
12                                   requires_grad=True)
13        self.disable_noise = disable_noise
14
15    def forward(self, x):
16        x = self.gen(x)
17        x = F.relu(self.fc1(x))
18        if not self.disable_noise:
19            dist=Normal(0,F.softplus(self.sigma))
20            x_sample = dist.rsample()
21            x = x + x_sample
22        return x
```



**Figure 1: Tsne visualization of classification result on CIFAR-10 trained on ResNet-18. No defense (Left). MFDV-SNN (Right).**

## 2 TSNE VISUALIZATION

We visualize the classification results on the CIFAR-10 dataset trained on ResNet-18, as shown in Figure 1.

In practice, we sample 1000 data and visualize them using t-SNE. No defense means we do not add randomness into the model, and MFDV-SNN represents our proposed method. 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. The decision boundary is smoother than the standard model, with many discrete samples. These characteristics may significantly reduce the possible adversarial regions.