Appendices for *Removing Adversarial Noise in Class Activation Feature Space*

A. Details of attack methods

In this section, we present supplementary information on details of attack methods. The projected gradient descent method (PGD), the decoupling direction and norm method (DDN), the Carlini and Wagner method (CW) and the spatial transform attack method (STA) are implemented by using Advertorch Toolbox. The translation-invariance input diversity method (TI-DIM), The autoattack method (AA) and the faster wasserstein attac method (FWA) are implemented from their open source codes. On *SVHN*, the main parameters of attacks are as follows:

- **CW**: We use the L_2 norm CW method to craft adversarial examples. The maximum number of iterations is 500. The confidence of the adversarial examples is 1. The initial value of the constant is 1.
- **DDN**: The number of iterations is 100. The factor to modify the norm at each iteration is 0.05. The number of quantization levels is 256.
- **PGD**: We use the L_∞ norm PGD method to craft adversarial examples. The default perturbation budget is 8/255. The default number of iterations is 40. The attack step size is 0.01.
- **TI-DIM**: The decay factor is 1. The default perturbation budget is 8/255. The default number of iterations is 40. The attack step size is 0.01.
- **AA**: The default perturbation budget is set to 8/255. The default number of iterations is set to 100.
- **STA**: The maximum number of iterations is set to 500. The number of search times to find the optimum is set to 20.
- **FWA**: The wasserstein adversarial examples are crafted by projected gradient descent (PGD) with dual projection. The default perturbation budget is set to 8/255. The number of iterations is set to 300. The learning rate is set to 0.1.

On *CIFAR-10*, the main parameters of attacks are as follows:

- **CW**: We use the L_2 norm CW method to craft adversarial examples. The maximum number of iterations is 500. The confidence of the adversarial examples is 1. The initial value of the constant is 1.
- **DDN**: The number of iterations is 100. The factor to modify the norm at each iteration is 0.05. The number of quantization levels is 256.
- **PGD**: We use the L_∞ norm PGD method to craft adversarial examples. The default perturbation budget is 8/255. The default number of iterations is 40. The attack step size is 0.01.
- **TI-DIM**: The decay factor is 1. The default perturbation budget is 8/255. The default number of iterations is 40. The attack step size is 0.01.
- **AA**: The default perturbation budget is 8/255. The default number of iterations is 100.
- **STA**: The maximum number of iterations is set to 500. The number of search times to find the optimum is set to 20.
- **FWA**: The wasserstein adversarial examples are crafted by projected gradient descent (PGD) with dual projection. The default perturbation budget is set to 8/255. The number of iterations is set to 300. The learning rate is set to 0.1.

B. Adversarial and restored examples

Defending against unseen types of attacks: In this section, we present supplementary information on defending against unseen types of attacks. Figure 1 show adversarial examples and restored examples on *CIFAR-10*. These adversarial examples are crafted by multiple attacks. These attacks include (i) pixel-constrained attacks: non-targeted DDN (DDN_N), non-targeted L_∞ norm PGD (PGD_N), targeted L_∞ norm PGD (PGD_T), non-targeted TI-DIM (TI-DIM_N) and non-targeted AA (AA_N), and (ii) spatial-constrained attacks: non-targeted STA (STA_N), targeted STA (STA_T) and non-targeted FWA (FWA_N). The categories corresponding to the class labels in *CIFAR-10* are as follows: 0) airplane, 1) car, 2) bird, 3) cat, 4) deer, 5) dog, 6) frog, 7) horse, 8) boat and 9) truck.

Cross-model defense results: In this section, we present supplementary information on cross-model results. In order to evaluate the cross-model defense capability of our method, we transfer our defense model to other classification models, i.e.,ResNet-50 and Wide-ResNet. Figure 2 and Figure 3 show adversarial examples against ResNet-50 and Wide-ResNet on *CIFAR-10* respectively. Their restored examples are also presented in the figures.

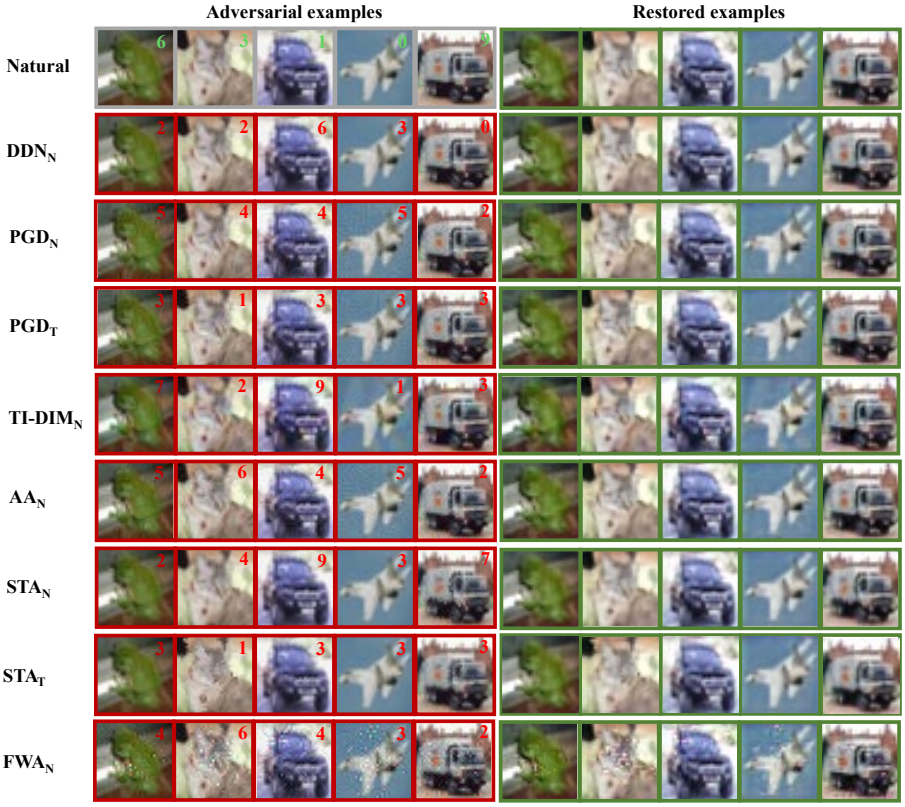

Figure 1. A visual illustration of adversarial examples and their restored examples. These adversarial examples are crafted by multiple attacks against VGG-19 on *CIFAR-10*.

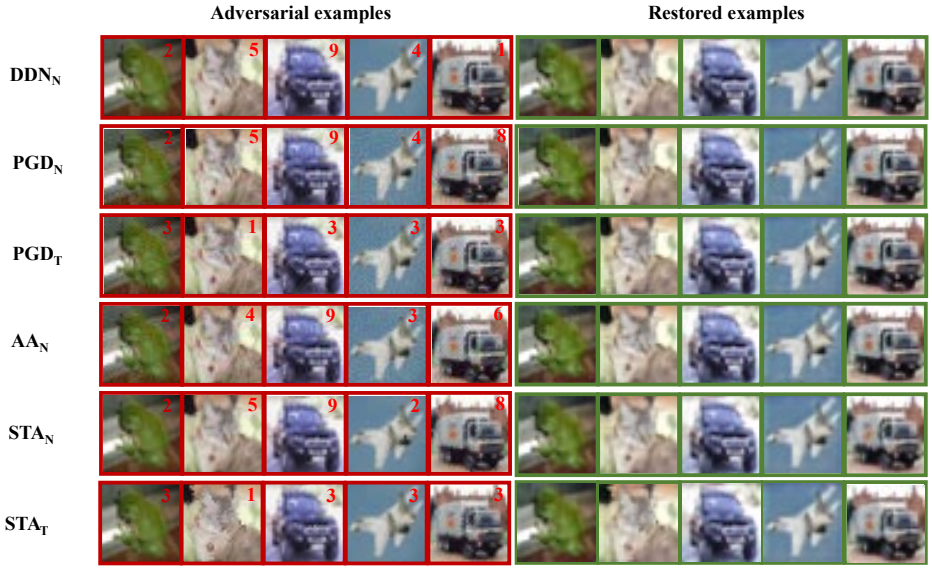

Figure 2. A visual illustration of adversarial examples and their restored examples. These adversarial examples are crafted by multiple attacks against ResNet-50 on *CIFAR-10*.

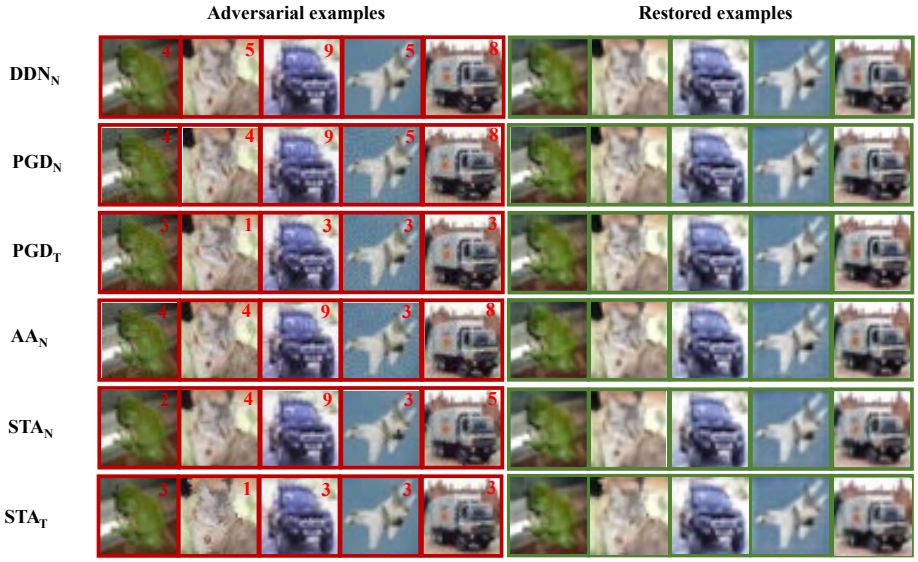

Figure 3. A visual illustration of adversarial examples and their restored examples. These adversarial examples are crafted by multiple attacks against Wide-ResNet on *CIFAR-10*.

C. Further Evaluations

To further demonstrate the effectiveness of the proposed CAFA for improving the adversarial robustness, we show the results of the proposed models trained using adversarial examples crafted by PGD (CAFD$^{\diamondsuit}$) and CW (CAFD$^{\sharp}$) in Table 1. Note that the architectures and training strategies of these defenses are the same as those of CAFD. In addition, in Table 1, we also present the results of previous defense methods when using adversarial examples crafted by PGD as adversarial training data. The results show that using CAFA can achieve a great defense performance and improve the generalization of the defense against unseen types of attacks.

Table 1. Classification error rates (percentage) on *CIFAR-10*. The target model is VGG-19.

	APE-G	HGD	AT	CAFD	CAFD$^{\diamondsuit}$	CAFD$^{\sharp}$
PGD$_N$	55.40	17.84	21.74	12.79	**12.61**	28.20
AA$_N$	56.20	18.34	23.90	**11.80**	12.87	28.80
STA$_N$	25.75	19.31	26.82	**18.19**	21.06	25.19
FWA$_N$	62.15	41.90	37.99	**35.59**	39.73	43.00