Sustainable Self-evolution Adversarial Training (Supplementary Materials)

Anonymous Authors

SUPPLEMENTARY MATERIALS

In this supplementary material, we provide more experimental results and insightful discussions in terms of three aspects,

- We conduct more experiments on CIFAR-100 dataset with adversarial training competitors for robustness and accuracy on clean samples;
- We demonstrate the impact of different amounts of training attack data at each training stage in continuous defense setting;
- we show visualization results of clean sample representations of our SSEAT model versus the baseline model.

Unless otherwise specified, the numbering of figures and tables is within the scope of the supplementary material. The notations are consistent with the main paper.

1 EXPREIMENT ON CIFAR-100 DATASET

Datasets. To better verify the effectiveness of our SSEAT method, we evaluate it on the CIFAR-100 dataset [6]. It contains 50,000 images for training and 10,000 images for testing, covering 100 different categories of objects. Our method uses only 100 images per category for training and all 10,000 images for testing, consistent with the CIFAR-10 setting. At each stage, training and test data are converted into attacks according to the corresponding attack algorithm. The converted training data part is used for SSEAT training, and the test part is only used for the final black-box test. In our experiments, we conduct two different attack orders for the CDS task: (1) Order-I: MIN, PGD, FGSM, SIM, BIM; (2) Order-II: FGSM, BIM, PGD, RFGSM, NIM, SIM, DIM.

Implementation Details. The model uses resnet18 [3] as the classification network structure. We implement Torchattacks [4] to generate 100 adversarial images per category for each adversarial attack strategy, resulting in a total of 10,000 images. On the CIFAR-100 dataset, before training, each image is resized to 32×32 pixels, and data augmentation is conducted, including horizontal flipping and random cropping. For the training hyperparameters of experiments, We use a batch size of 64. We train clean samples for 40 epochs and adversarial samples for 20 epochs. We train the model using the SGD optimizer with momentum 0.9 and weight decay 5×10^{-4}. In addition, we set the memory buffer size to 1000.

Competitors. We compare our SSEAT method with several adversarial training works, such as the PGD-AT [9], TRADES[12], MART[10], AWP [11], RNA [2], LBGAT [1], FSR [5] and ST [7].

Results Analyses.

On CIFAR-100 dataset, our SSEAT model can maintain the best defense performance against the attacks and competitive classification accuracy for clean samples. To verify the adaptive defense capabilities of deep models in the face of constantly generating diverse new adversarial samples, we conduct extensive experiments on the CIFAR-100 dataset using various attack sequences in the CDS task, comparing with several competitors and baselines. In the main text, to face various complex attacks, we have designed a novel Continuous Adversarial Defense (CAD) pipeline. In the initial phase, the model is trained using clean original data. In each subsequent phase, the model is exposed to a batch of new attack samples for training, enabling continuous adaptation to the evolving environment and data distribution. All attack results are reported under black box conditions. We report the model's robustness to multiple attack sequences and classification accuracy on raw data compared to several competitors , and our SSEAT method can beat all adversarial methods trained on all adversarial examples. As shown in Tab. 1 and Tab. 3, this proves

Table 1: Comparing results of our SSEAT method with other adversarial training competitors under CDS task Order-I on the CIFAR-100 dataset, including classification accuracy against attacks and standard accuracy on clean samples.

Method	MIM	PGD	FGSM	SIM	BIM	Clean
PGD-AT[9]	51.13	55.93	50.53	46.21	50.76	64.11
TRADES[12]	46.30	49.02	45.78	47.27	49.28	48.33
MART [10]	49.59	49.44	49.30	48.47	48.97	50.49
AWP [11]	47.07	48.96	42.12	45.97	40.36	59.38
RNA [2]	48.86	47.64	45.11	51.01	48.57	59.19
LBGAT [1]	53.62	52.50	48.94	54.18	55.89	**64.56**
FSR [5]	53.56	54.32	51.36	52.31	53.26	60.22
ST [7]	51.90	51.47	51.44	52.91	52.43	58.15
Baseline (CAD)	50.83	54.87	48.26	48.19	51.36	47.25
SSEAT (Ours)	**54.14**	**58.34**	**51.80**	**54.44**	**56.71**	63.37

Table 2: Comparing results of our SSEAT method with other adversarial training competitors under rearranged CDS task Order-I on the CIFAR-100 dataset, including classification accuracy against attacks and standard accuracy on clean samples.

Method	FGSM	PGD	SIM	BIM	MIM	Clean
PGD-AT[9]	50.53	55.93	46.21	50.76	51.13	64.11
TRADES[12]	45.78	49.02	47.27	49.28	46.30	48.33
MART [10]	49.30	49.44	48.47	48.97	49.59	50.49
AWP [11]	42.12	48.96	45.97	40.36	47.07	59.38
RNA [2]	45.11	47.64	51.01	48.57	48.86	59.19
LBGAT [1]	48.94	52.50	54.18	55.89	53.62	**64.56**
FSR [5]	51.36	54.32	52.31	53.26	53.56	60.22
ST [7]	51.44	51.47	52.91	52.43	51.90	58.15
Baseline (CAD)	50.67	57.74	52.89	56.53	52.36	48.75
SSEAT (Ours)	**54.21**	**60.09**	**56.35**	**58.98**	**56.97**	63.85

Table 3: Comparing results of our SSEAT method with other adversarial training competitors under CDS task Order-II on the CIFAR-100 dataset, including classification accuracy against attacks and standard accuracy on clean samples.

Method	FGSM	BIM	PGD	RFGSM	NIM	SIM	DIM	Clean
PGD-AT[9]	50.53	50.76	55.93	49.10	47.37	46.21	49.32	64.11
TRADES[12]	45.78	49.28	49.02	48.98	46.39	47.27	49.98	48.33
MART [10]	49.30	48.97	49.44	43.89	49.49	48.47	48.61	50.49
AWP [11]	42.12	40.36	48.96	44.56	44.17	45.97	43.34	59.38
RNA [2]	45.11	48.57	47.64	50.13	49.33	51.01	47.40	59.19
LBGAT [1]	48.94	55.89	52.50	49.28	52.03	54.18	49.68	**64.56**
FSR [5]	51.36	53.26	54.32	52.18	51.48	52.31	52.42	60.22
ST [7]	51.44	52.43	51.47	50.41	52.05	52.91	51.81	58.15
Baseline (CAD)	48.52	54.86	55.32	54.89	51.52	52.19	50.19	47.88
SSEAT (Ours)	**53.03**	**58.09**	**58.40**	**58.62**	**55.43**	**55.80**	**54.45**	63.01

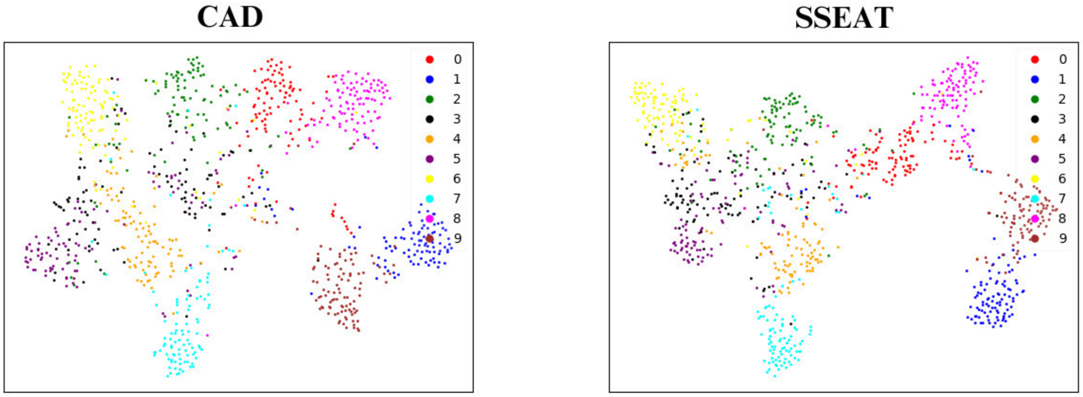

Figure 1: Visualization of clean examples representations for CAD and SSEAT by using t-SNE[8]. We use 1000 test images from CFAIR-10 dataset for visualization. Different colors represent different categories.

Table 4: The results of our SEAT with different numbers of training data in one training stage in the CAD pipeline. The first column "K" in the table represents the amount of training data used in adversarial training at each stage.

K	DIM	RFGSM	PGD	SIM	FGSM	clean
100	70.45	74.78	75.35	73.88	72.53	81.50
500	87.18	88.38	88.04	87.51	86.27	79.65
1000	92.24	92.24	92.13	91.95	92.79	78.80

that the SSEAT framework can perform well in a black box under the CDS conditions, which can effectively resist new attacks and have good recognition effect on clean samples.

Our SSEAT model is capable of adapting to more demanding CDS tasks. To further validate the effectiveness of our method, we intentionally disrupte the original attack sequence. As shown in Table 2, after rearranging the attack sequence for Order-I in the CDS task, the defensive performance does not change significantly. This clearly demonstrates that the SSEAT method is capable of effectively defending against various adversarial samples.

2 ABLATION STUDY

To better understand the insight of our proposed model and verify its effectiveness, we further conduct several ablation studies below to evaluate different design choices.

The impact of the amount of training data in each training stage. We design the number of data to be different in each training stage on the CIFAR-10 dataset. As shown in Tab. 4, when the number of attacks increases, the model robustness will be significantly improved, and the accuracy of clean samples will decrease slightly, which is in line with conventional facts. This fully demonstrates that our method has broad and universal model robustness when facing a large number of adversarial examples.

Visualization experiments of clean samples representations. We additionally visualize the representations of clean examples with CAD and SSEAT. As shown in Fig. 1, we can see that compared with CAD, SSEAT exhibits more distinct distances between classes in feature representations on clean samples, with closer intra-class distances, which enhances its discriminative ability.For example, in SSEAT, the distribution of red points representing Class 0 is more clustered, whereas in CAD, the distribution is more dispersed. This observation indicates that SSEAT obtains more discriminative representations of clean examples compared with CAD. These results indicate that our SSEAT can maintain high accuracy on clean samples.

3 LIMITATION AND FUTURE WORK

In the real world, data is often subject to noise and disturbances, necessitating models with strong robustness and adaptability. The SSEAT framework we propose aims to achieve autonomous model evolution and defense against various adversarial attacks, enabling models to effectively adapt to complex and changing environments, thereby enhancing their reliability and stability in practical applications. Additionally, SSEAT effectively mitigates the trade-off between accuracy on clean samples and robustness when facing multiple adversarial attacks. However, due to limited memory size, it is inevitable that the model's ability to recognize adversarial samples will gradually decline. A promising direction is to integrate the SSEAT framework with dynamic networks. This integration aims to preserve crucial parameters for recognizing clean samples during dynamic network expansion, thereby maintaining model accuracy on clean samples.We plan to explore this direction in the future.

REFERENCES

[1] Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. 2021. Learnable boundary guided adversarial training. In *Proceedings of the IEEE/CVF international conference on computer vision*. 15721–15730.

[2] Minjing Dong, Xinghao Chen, Yunhe Wang, and Chang Xu. 2022. Random normalization aggregation for adversarial defense. *Advances in Neural Information Processing Systems* 35 (2022), 33676–33688.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[4] Hoki Kim. 2020. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950* (2020).

[5] Woo Jae Kim, Yoonki Cho, Junsik Jung, and Sung-Eui Yoon. 2023. Feature separation and recalibration for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8183–8192.

[6] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[7] Qizhang Li, Yiwen Guo, Wangmeng Zuo, and Hao Chen. 2022. Squeeze training for adversarial robustness. *arXiv preprint arXiv:2205.11156* (2022).

[8] George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger, and Yuval Kluger. 2017. Efficient algorithms for t-distributed stochastic neighborhood embedding. *arXiv preprint arXiv:1712.09005* (2017).

[9] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[10] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. 2019. Improving adversarial robustness requires revisiting misclassified examples. In *International conference on learning representations*.

[11] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. 2020. Adversarial weight perturbation helps robust generalization. *Advances in neural information processing systems* 33 (2020), 2958–2969.

[12] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*. PMLR, 7472–7482.