

# Supplementary Materials: Balancing Generalization and Robustness in Adversarial Training via Steering through Clean and Adversarial Gradient Directions

## 1 Additional Results of Cosine Similarity in Baselines

In this section, we provide the distribution of cosine similarity values for the baseline adversarial training methods on CIFAR10 and CIFAR100 datasets. We observed that there are almost no convolutional layer parameters with a similarity less than 0 in AT as shown in Figure 3, 4, which leads to almost no improvement in AT-AGR compared to vanilla AT in the ablation experiments on GOP. However, there are still some GOP operations in AT-AGR (it refers to the fact that the GOP frequency is not 0 in AT-AGR), which suggests that it occurs in the batch normalization layer.

On the other hand, as shown in Figure 5, 6, and 7, similarity smaller than 0 is primarily seen in the more forward convolutional layers in the experimental results of TRADE and MART. This may indicate that the model already exhibits differences in the low-level features extracted from the clean and adversarial examples. And the adversarial training will dilute this difference, thereby enhancing robustness. Therefore, there are not too many cases where the cosine similarity is less than 0 in the backward convolutional layers of the model. Nevertheless, using GOP operations on only a small fraction of the gradient also improves generalization.

## 2 Algorithm

### 3 Proofs

**THEOREM 3.1.** *Let  $\nabla \mathcal{L}_n, \nabla \mathcal{L}_{adv}$  denote the natural and robust gradients, respectively. For arbitrary cases of  $-1 \leq \Psi(\nabla \mathcal{L}_n^{(i)}, \nabla \mathcal{L}_{adv}^{(i)}) < 0$  and  $0 \leq \Psi(\nabla \mathcal{L}_n^{(i)}, \nabla \mathcal{L}_{adv}^{(i)}) \leq 1$ , AGR with gradient  $G$  for one iteration, it holds that  $G$  induces a descent in both  $\mathcal{L}_n$  and  $\mathcal{L}_{adv}$ .*

**PROOF.** To show that the gradient direction  $G$  leads to a decrease in both  $\mathcal{L}_n$  and  $\mathcal{L}_{adv}$ , we have to guarantee that  $G$  is positively correlated with both  $\nabla \mathcal{L}_n$  and  $\nabla \mathcal{L}_{adv}$ . We separate the proof into two phases: negative correlation gradient pairs and positive correlation gradient pairs. Without loss of generality, we assume that the first  $m$  ( $0 < m \leq c$ ) terms of  $\nabla \mathcal{L}_{adv} = [\nabla \mathcal{L}_{adv}^{(1)}, \dots, \nabla \mathcal{L}_{adv}^{(m)}, \dots, \nabla \mathcal{L}_{adv}^{(c)}]$  are negatively correlated with  $\nabla \mathcal{L}_n = [\nabla \mathcal{L}_n^{(1)}, \dots, \nabla \mathcal{L}_n^{(m)}, \dots, \nabla \mathcal{L}_n^{(c)}]$ .

**Phase I: Negative correlation gradient pairs.**

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.  
 MM '24, October 28–November 1, 2024, Melbourne, VIC, Australia  
 © 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
 ACM ISBN 978-x-xxxx-xxxx-x/YY/MM  
<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

---

### Algorithm 1: Adversarial Training with AGR

---

**Input:** Training dataset  $S = x_1, x_2, \dots, x_n$ , Initial learning rate  $\eta$ , number of iterations  $T$ , adversarially trained model parameter  $\theta = (\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(c)})$ , step size of attack  $\epsilon_{step}$ , maximum perturbation  $\epsilon$

**Result:** Optimal robust model  $\theta_T$

---

```

1 for  $t = 1$  to  $T$  do
2   for  $i = 1$  to  $n$  do
3      $x'_i \leftarrow \text{PGD}(x_i, y_i, \epsilon_{step}, \epsilon)$ 
4   end
5   for  $j = 1$  to  $c$  do
6     Compute cosine similarity:
7      $cs = \Psi(\nabla_{\theta^{(i)}} \mathcal{L}_n(x), \nabla_{\theta^{(i)}} \mathcal{L}_{adv}(x'))$ 
8     if  $cs < 0$  then
9       Reconstruct the gradient  $G$  use Eq. ??
10    else
11      Reconstruct the gradient  $G$  use Eq. ??
12    end
13     $\theta_{t+1}^{(j)} \leftarrow \theta_t^{(j)} - \eta \cdot \text{Clip}(G)$ 
14  end
15 return  $\theta_T$ 

```

---

For  $i = 0, 1, \dots, m$ , we have  $-1 \leq \Psi(\nabla \mathcal{L}_n^{(i)}, \nabla \mathcal{L}_{adv}^{(i)}) < 0$ , then orthogonally project  $\nabla \mathcal{L}_{adv}^{(i)}$  along  $\nabla \mathcal{L}_n^{(i)}$  to obtain the new direction  $g^i$  as follows:

$$g^i = \nabla \mathcal{L}_{adv}^{(i)} - \frac{\langle \nabla \mathcal{L}_{adv}^{(i)}, \nabla \mathcal{L}_n^{(i)} \rangle}{\langle \nabla \mathcal{L}_n^{(i)}, \nabla \mathcal{L}_n^{(i)} \rangle} \nabla \mathcal{L}_n^{(i)}. \quad (1)$$

It is obvious that  $g^i$  is orthogonal to  $\nabla \mathcal{L}_n^{(i)}$ . Next, we show that  $g^i$  is positively correlated with  $\nabla \mathcal{L}_{adv}^{(i)}$ .

$$\begin{aligned}
 \langle \nabla \mathcal{L}_{adv}^{(i)}, g^i \rangle &= \langle \nabla \mathcal{L}_{adv}^{(i)}, \nabla \mathcal{L}_{adv}^{(i)} - \frac{\langle \nabla \mathcal{L}_{adv}^{(i)}, \nabla \mathcal{L}_n^{(i)} \rangle}{\langle \nabla \mathcal{L}_n^{(i)}, \nabla \mathcal{L}_n^{(i)} \rangle} \nabla \mathcal{L}_n^{(i)} \rangle \\
 &= \langle \nabla \mathcal{L}_{adv}^{(i)}, \nabla \mathcal{L}_{adv}^{(i)} \rangle - \langle \nabla \mathcal{L}_{adv}^{(i)}, \frac{\langle \nabla \mathcal{L}_{adv}^{(i)}, \nabla \mathcal{L}_n^{(i)} \rangle}{\langle \nabla \mathcal{L}_n^{(i)}, \nabla \mathcal{L}_n^{(i)} \rangle} \nabla \mathcal{L}_n^{(i)} \rangle \\
 &= \|\nabla \mathcal{L}_{adv}^{(i)}\|_2^2 - \frac{\langle \nabla \mathcal{L}_{adv}^{(i)}, \nabla \mathcal{L}_n^{(i)} \rangle}{\langle \nabla \mathcal{L}_n^{(i)}, \nabla \mathcal{L}_n^{(i)} \rangle} \cdot \langle \nabla \mathcal{L}_{adv}^{(i)}, \nabla \mathcal{L}_n^{(i)} \rangle \\
 &= \|\nabla \mathcal{L}_{adv}^{(i)}\|_2^2 - \frac{\|\nabla \mathcal{L}_{adv}^{(i)}\|_2^2 \cdot \|\nabla \mathcal{L}_n^{(i)}\|_2^2 \cdot \Psi^2(\nabla \mathcal{L}_n^{(i)}, \nabla \mathcal{L}_{adv}^{(i)})}{\|\nabla \mathcal{L}_n^{(i)}\|_2^2} \\
 &= \|\nabla \mathcal{L}_{adv}^{(i)}\|_2^2 - \|\nabla \mathcal{L}_{adv}^{(i)}\|_2^2 \cdot \Psi^2(\nabla \mathcal{L}_n^{(i)}, \nabla \mathcal{L}_{adv}^{(i)}) \geq 0
 \end{aligned} \quad (2)$$

### Phase II: Positive correlation gradient pairs.

For  $i = m, m+1, \dots, c$ , we have  $0 \leq \Psi(\nabla \mathcal{L}_n^{(i)}, \nabla \mathcal{L}_{adv}^{(i)}) \leq 1$ , then apply the cosine similarity-based interpolation to obtain the new direction  $\mathbf{g}^i$  as follows:

$$\mathbf{g}^i = cs \cdot \nabla \mathcal{L}_n^{(i)} + (1 - cs) \cdot \nabla \mathcal{L}_{adv}^{(i)}, \quad (3)$$

where  $cs$  is the cosine similarity between  $\nabla \mathcal{L}_{adv}^{(i)}$  and  $\nabla \mathcal{L}_n^{(i)}$ . Below we show that  $\mathbf{g}^i$  is positively correlated with both  $\nabla \mathcal{L}_{adv}^{(i)}$  and  $\nabla \mathcal{L}_n^{(i)}$ .

$$\begin{aligned} \langle \nabla \mathcal{L}_{adv}^{(i)}, \mathbf{g}^i \rangle &= \langle \nabla \mathcal{L}_{adv}^{(i)}, cs \cdot \nabla \mathcal{L}_n^{(i)} + (1 - cs) \cdot \nabla \mathcal{L}_{adv}^{(i)} \rangle \\ &= cs \cdot \langle \nabla \mathcal{L}_{adv}^{(i)}, \nabla \mathcal{L}_n^{(i)} \rangle + (1 - cs) \cdot \|\nabla \mathcal{L}_{adv}^{(i)}\|_2^2 \end{aligned} \quad (4)$$

by the  $0 \leq \Psi(\nabla \mathcal{L}_n^{(i)}, \nabla \mathcal{L}_{adv}^{(i)}) \leq 1$ , we have

$$\langle \nabla \mathcal{L}_{adv}^{(i)}, \mathbf{g}^i \rangle \geq 0 \quad (5)$$

In the similar way, we can obtain  $\langle \nabla \mathcal{L}_n^{(i)}, \mathbf{g}^i \rangle \geq 0$ .

Overall, each component of the new gradient direction  $\mathbf{G} = [\mathbf{g}^1, \mathbf{g}^2, \dots, \mathbf{g}^c]$  is positively correlated or orthogonal to the corresponding components of  $\nabla \mathcal{L}_{adv}^{(i)}$  and  $\nabla \mathcal{L}_n^{(i)}$ , implying that both  $\langle \mathbf{G}, \nabla \mathcal{L}_{adv} \rangle \geq 0$  and  $\langle \mathbf{G}, \nabla \mathcal{L}_n \rangle \geq 0$  hold.  $\square$

**THEOREM 3.2.** Assume that  $\mathcal{F}$  is a function space with the range  $[0, 1]$ , let  $\mathcal{D}^{N_s} = \{\mathbf{z}_n^s\}_{n=1}^{N_s}$  and  $\mathcal{D}^{N_a} = \{\mathbf{z}_n^a\}_{n=1}^{N_a}$  be two datasets of i.i.d samples drawn from the standard domain  $\mathcal{D}$  and adversarial domain  $\mathcal{T}$ . Then, given  $\lambda \in [0, 1]$  and for any  $\epsilon > 0$ , with probability at least  $1 - \epsilon$ ,

$$\begin{aligned} R_{\mathcal{D}}(f_{AGR}) - R_{S+S'}(f_{AGR}) &\leq 2\lambda \hat{\mathcal{R}}_S(\mathcal{F}) + 3\lambda \sqrt{\frac{\ln(2/\epsilon)}{2N_s}} \\ &\quad + (1 - \lambda) D_{\mathcal{F}}(\mathcal{D}, \mathcal{T}) + 2(1 - \lambda) \hat{\mathcal{R}}_{S'}(\mathcal{F}) \\ &\quad + 3(1 - \lambda) \sqrt{\frac{\ln(2/\epsilon)}{2N_a}} + \sqrt{\frac{\ln(1/\epsilon)}{2} \left( \frac{\lambda^2}{N_s} + \frac{(1 - \lambda)^2}{N_a} \right)} \\ &\leq 2c\lambda B \frac{(\sqrt{2d \log 2} + 1) \prod_{j=1}^d M_F(j)}{\sqrt{N_s}} + 3\lambda \sqrt{\frac{\ln(2/\epsilon)}{2N_s}} \\ &\quad + (1 - \lambda) D_{\mathcal{F}}(\mathcal{D}, \mathcal{T}) + 3(1 - \lambda) \sqrt{\frac{\ln(2/\epsilon)}{2N_a}} \\ &\quad + 2c(1 - \lambda) B \frac{(\sqrt{2d \log 2} + 1) \prod_{j=1}^d M_F(j)}{\sqrt{N_a}} \\ &\quad + \sqrt{\frac{\ln(1/\epsilon)}{2} \left( \frac{\lambda^2}{N_s} + \frac{(1 - \lambda)^2}{N_a} \right)} \end{aligned}$$

**PROOF.** To derive generalization error bounds using Rademacher complexity, we will use a very useful bounded differences inequality called McDiarmid Inequality, and here we present a two-domain extended version of it.

**LEMMA 3.3. (McDiarmid Inequality.)** Given two independent domains  $\mathcal{D}$  and  $\mathcal{T}$ , let  $\mathcal{D}^{N_s} = \{\mathbf{z}_n^s\}_{n=1}^{N_s}$  and  $\mathcal{D}^{N_a} = \{\mathbf{z}_n^a\}_{n=1}^{N_a}$  be  $N_s$

and  $N_a$  independent random variables from the domain  $\mathcal{D}$  and  $\mathcal{T}$ , respectively. If  $\mathcal{G} : (\mathcal{D})^{N_s} \times (\mathcal{T})^{N_a} \rightarrow \mathbb{R}$  satisfies

$$\sup_{\mathcal{D}^{N_s}, \mathcal{D}^{N_a}, \mathbf{z}_i^{j'}} |\mathcal{G}(\mathbf{z}_1^s, \dots, \mathbf{z}_i^j, \dots, \mathbf{z}_n^a) - \mathcal{G}(\mathbf{z}_1^s, \dots, \mathbf{z}_i^{j'}, \dots, \mathbf{z}_n^a)| \leq c_i^{(k)}$$

for any  $0 \leq i \leq N_s + N_a$ ,  $j, k \in \{s, a\}$ , and  $\epsilon > 0$ , then

$$\Pr[\mathcal{G}(\mathbf{z}_1^s, \dots, \mathbf{z}_n^a) - \mathbb{E}[\mathcal{G}(\mathbf{z}_1^s, \dots, \mathbf{z}_n^a)] \geq \epsilon] \leq e^{-2\epsilon^2 / \sum_{i=1}^{N_s} (c_i^s)^2 + \sum_{i=1}^{N_a} (c_i^a)^2}$$

**LEMMA 3.4.** Given a distribution  $\mathcal{D}$  and any  $\epsilon \in (0, 1]$ . Let  $\mathcal{F} \subseteq [0, 1]$  and set  $S = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}$  be i.i.d drawn from the  $\mathcal{D}$ , then with probability  $1 - \epsilon$  at least,

$$R_{\mathcal{D}}(f) - R_S(f) \leq 2\hat{\mathcal{R}}(\mathcal{F}) + \sqrt{\frac{\ln(1/\epsilon)}{2n}} \quad (6)$$

In addition, with probability  $1 - \epsilon$  at least we also have,

$$R_{\mathcal{D}}(f) - R_S(f) \leq 2\hat{\mathcal{R}}(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\epsilon)}{2n}} \quad (7)$$

Assume a fixed function  $f$ , by the definition of the supremum we get

$$R_{\mathcal{D}}(f) - R_{S+S'}(f) \leq \sup_g |R_{\mathcal{D}}(g) - R_{S+S'}(g)| \quad (8)$$

Denoting

$$\begin{aligned} \mathcal{G}(S, S') &= \sup_g |R_{\mathcal{D}}(g) - R_{S+S'}(g)| \\ &= \sup_g |R_{\mathcal{D}}(g) - \lambda R_S(g) - (1 - \lambda) R_{S'}(g)| \end{aligned} \quad (9)$$

It is accessible to deduce that  $\mathcal{G}(S, S')$  satisfies

$$c_i^s = \frac{\lambda}{N_s}, c_i^a = \frac{1 - \lambda}{N_a} \quad (10)$$

Next we apply the McDiarmid Inequality, we can obtain

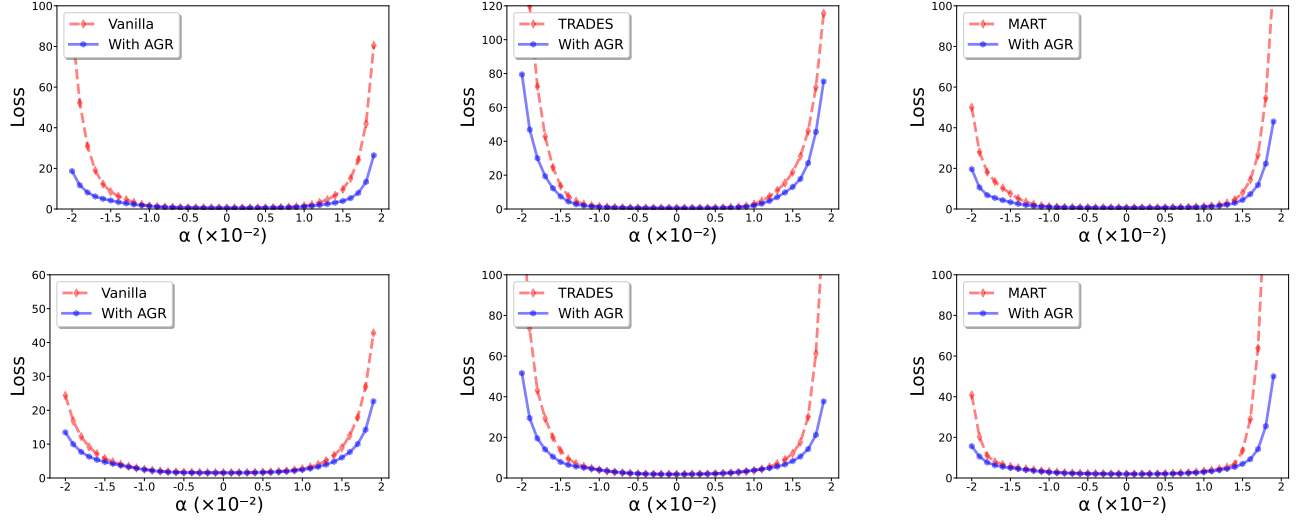
$$\Pr[\mathcal{G}(S, S') - \mathbb{E}[\mathcal{G}(S, S')] \geq \delta] \leq e^{-2\delta^2 / (\frac{\lambda^2}{N_s} + \frac{(1 - \lambda)^2}{N_a})} \quad (11)$$

For any  $\epsilon > 0$ , let the above probability be less than  $\epsilon$ , which means

$$\begin{aligned} \text{if and only if } \delta &\geq \sqrt{\frac{\ln(1/\epsilon)}{2} \left( \frac{\lambda^2}{N_s} + \frac{(1 - \lambda)^2}{N_a} \right)} \text{ there is a probability of} \\ \text{at least } 1 - \epsilon &\text{ we get the following,} \\ R_{\mathcal{D}}(g) - R_{S+S'}(g) &\leq \mathcal{G}(S, S') \leq \mathbb{E}[\mathcal{G}(S, S')] + \sqrt{\frac{\ln(1/\epsilon)}{2} \left( \frac{\lambda^2}{N_s} + \frac{(1 - \lambda)^2}{N_a} \right)} \\ &\leq \mathbb{E} \left[ \sup_g |R_{\mathcal{D}}(g) - \lambda R_S(g) - (1 - \lambda) R_{S'}(g)| \right] \\ &\leq \lambda \sup_g |R_{\mathcal{D}}(g) - R_S(g)| + (1 - \lambda) \sup_g |R_{\mathcal{D}}(g) - R_{S'}(g)| \\ &\quad + \sqrt{\frac{\ln(1/\epsilon)}{2} \left( \frac{\lambda^2}{N_s} + \frac{(1 - \lambda)^2}{N_a} \right)} \\ &\leq \lambda \sup_g |R_{\mathcal{D}}(g) - R_S(g)| + (1 - \lambda) \sup_g |R_{\mathcal{D}}(g) - R_{\mathcal{T}}(g)| \\ &\quad + (1 - \lambda) \sup_g |R_{\mathcal{T}}(g) - R_{S'}(g)| + \sqrt{\frac{\ln(1/\epsilon)}{2} \left( \frac{\lambda^2}{N_s} + \frac{(1 - \lambda)^2}{N_a} \right)} \end{aligned} \quad (12)$$

According to Lemma 3.4, The first term of the last inequality of Equation 12 satisfies

$$\sup_g |R_{\mathcal{D}}(g) - R_S(g)| \leq 2\hat{\mathcal{R}}_S(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\epsilon)}{2N_s}} \quad (13)$$



**Figure 1: Comparison of the weight loss landscape for vanilla adversarial training methods and AGR of PreAct-ResNet-18 trained on CIFAR10/100. The first row of graphs shows the visualization results of AT, TRADES, and MART on CIFAR10 respectively and the second row shows the results on CIFAR100. These curves are the change in loss when moving model weight in the direction of a randomly sampled from a Gaussian distribution with the step size of  $\alpha$ .**

Similarly, the third term satisfies

$$\sup_g |R_{\mathcal{T}}(g) - R_{S'}(g)| \leq 2\hat{\mathfrak{R}}_{S'}(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\epsilon)}{2N_a}} \quad (14)$$

While the second term we can denote as  $D_{\mathcal{F}}(\mathcal{D}, \mathcal{T})$ , which was proposed by [2] to measure the difference between two probability distributions named integral probability metric. Synthesizing the inequalities 12, 13, and 14 we get

$$\begin{aligned} R_{\mathcal{D}}(g) - R_{S+S'}(g) &\leq 2\lambda\hat{\mathfrak{R}}_S(\mathcal{F}) + 3\lambda\sqrt{\frac{\ln(2/\epsilon)}{2N_s}} \\ &\quad + (1-\lambda)D_{\mathcal{F}}(\mathcal{D}, \mathcal{T}) + 2(1-\lambda)\hat{\mathfrak{R}}_{S'}(\mathcal{F}) \\ &\quad + 3(1-\lambda)\sqrt{\frac{\ln(2/\epsilon)}{2N_a}} + \sqrt{\frac{\ln(1/\epsilon)}{2} \left( \frac{\lambda^2}{N_s} + \frac{(1-\lambda)^2}{N_a} \right)} \end{aligned} \quad (15)$$

Following **Theorem 1** in [1], we assume that the activation functions be 1-Lipschitz, positive-homogeneous, the Frobenius norm for each parameter matrix  $W_j$  is at most  $M_F(j)$ , number of classification categories  $c$ , and  $\sqrt{\sum_{i=1}^n \|x_i\|^2} \leq B (n \in \{N_s, N_a\})$ , then with

probability  $1 - \epsilon$  at least we have

$$\begin{aligned} R_{\mathcal{D}}(g) - R_{S+S'}(g) &\leq 2c\lambda B \frac{(\sqrt{2d \log 2} + 1) \prod_{j=1}^d M_F(j)}{\sqrt{N_s}} + 3\lambda\sqrt{\frac{\ln(2/\epsilon)}{2N_s}} \\ &\quad + (1-\lambda)D_{\mathcal{F}}(\mathcal{D}, \mathcal{T}) + 3(1-\lambda)\sqrt{\frac{\ln(2/\epsilon)}{2N_a}} \\ &\quad + 2c(1-\lambda)B \frac{(\sqrt{2d \log 2} + 1) \prod_{j=1}^d M_F(j)}{\sqrt{N_a}} \\ &\quad + \sqrt{\frac{\ln(1/\epsilon)}{2} \left( \frac{\lambda^2}{N_s} + \frac{(1-\lambda)^2}{N_a} \right)} \end{aligned} \quad (16)$$

□

## 4 Training Details

### 4.1 Environment

All experiments are implemented with Python 3.9.16 and PyTorch 2.0.0 on a machine with Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz CPU, 32GB RAM, and a Nvidia 3090 GPU.

### 4.2 Architecture

Most of the experiments used PreAct-ResNet-18 with a depth of 18 and 11.1M parameters. This network is widely used in image classification tasks due to its superior performance, so we use it as the main testing tool. As a variant of ResNet-18, it changes the model architecture by replacing the order of Conv-BN-ReLU in order to improve accuracy. We also used WideResNet-34 to test the algorithm performance with a depth of 34 and 46.2M parameters.



Figure 2: The loss change with respect to epochs of AT-AGR, TRADES-AGR, and MART-AGR of PreAct-ResNet-18 trained on Tiny Imagenet.

### 4.3 Hyperparameter Setting

In terms of hyperparameter settings for adversarial training, we trained PreAct-ResNet-18 and WideResNet-34 for 200 epochs by SGD with momentum of 0.9 and weight decay of  $5 \times 10^{-4}$ . The initial learning rate was 0.1, divided by 10 at the 100-th and 150-th epochs.

When trained up to 155 epochs, we incorporate AGR with baseline adversarial training methods as it has reached a very high level of robustness. We set the clipping threshold  $C$  as 0.1, which does not affect the generalization improvement but suppresses the robust overfitting.

## 5 More Results of Generalization and Robustness

In this section, we exhibit the additional experiments of our approach.

### 5.1 The Visualization of Generalization

As a remarkably effective measure of generalization, the weight loss landscape has been widely used, where the flatter the landscape the higher the generalization of the model. We provide visualization results comparing the generalization of AT-AGR, TRADES-AGR, and MART-AGR with the baseline adversarial training methods on CIFAR10 and CIFAR100. As shown in Figure 1, we can observe that the loss landscape of all methods combining AGR performs much flatter, which means that our proposed method can achieve a better generalization. Through the presentation of the results of loss change and weight loss landscape, all the results demonstrate the effectiveness and feasibility of our proposed method.

### 5.2 Evaluations on the Loss Change of AGR

We show the variation curves of loss on the more challenging dataset Tiny Imagenet when the proposed AGR combines AT, TRADES, and MART. As shown in Figure 2, we show that the proposed method can make loss converge to a flat range, which implies the effectiveness of our AGR training approach. However, it can be noted that our method is still unable to avoid the occurrence of overfitting in adversarial training, as can be seen in the figure, where the robust loss in the test set shows an increase.

## References

- [1] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. 2018. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*. PMLR, 297–299.
- [2] Chao Zhang, Lei Zhang, and Jieping Ye. 2012. Generalization bounds for domain adaptation. *Advances in neural information processing systems* 25 (2012).

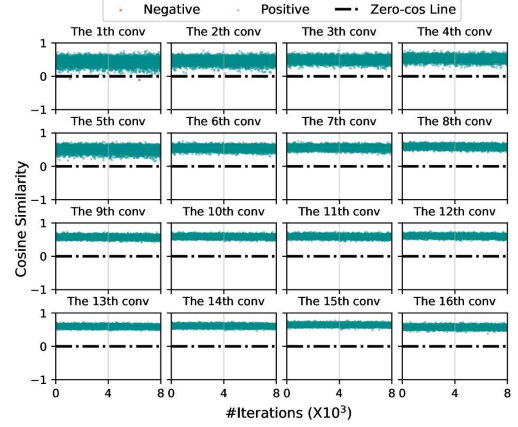


Figure 3: The cosine similarity of  $\nabla \mathcal{L}_n$  and  $\nabla \mathcal{L}_{adv}$  of weights of convolutional layers of PreAct-ResNet-18 trained on CIFAR10 by AT.

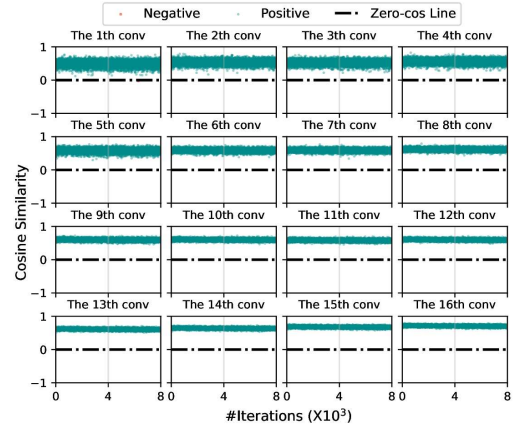


Figure 4: The cosine similarity of  $\nabla \mathcal{L}_n$  and  $\nabla \mathcal{L}_{adv}$  of weights of convolutional layers of PreAct-ResNet-18 trained on CIFAR100 by AT.

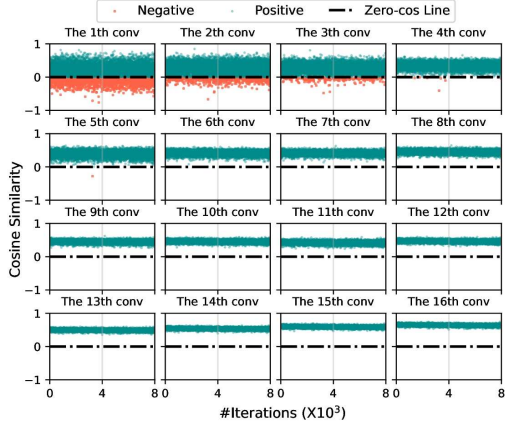


Figure 7: The cosine similarity of  $\nabla \mathcal{L}_n$  and  $\nabla \mathcal{L}_{adv}$  of weights of convolutional layers of PreAct-ResNet-18 trained on CI-FAR100 by MART.

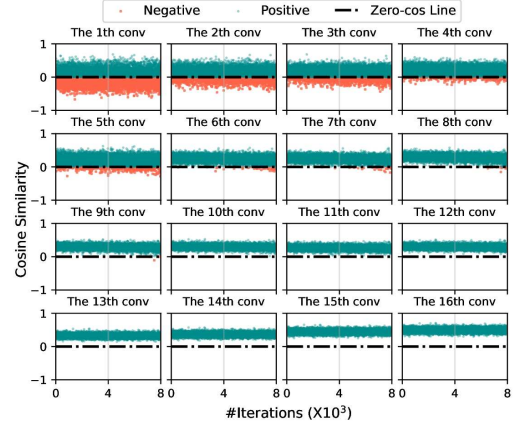


Figure 6: The cosine similarity of  $\nabla \mathcal{L}_n$  and  $\nabla \mathcal{L}_{adv}$  of weights of convolutional layers of PreAct-ResNet-18 trained on CI-FAR100 by TRADES.

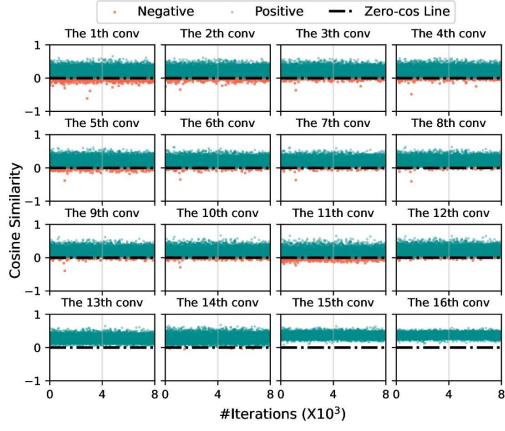


Figure 5: The cosine similarity of  $\nabla \mathcal{L}_n$  and  $\nabla \mathcal{L}_{adv}$  of weights of convolutional layers of PreAct-ResNet-18 trained on CI-FAR10 by MART.