A ALGORITHM FOR CALCULATING THE FEATURE ATTRIBUTION CORRELATION MATRIX

We present the complete algorithm of calculating the feature attribution correlation matrix in Algorithm 2. For each class, we first calculate the feature attribution vectors for each test adversarial sample, then calculate the mean of these vectors as the feature attribution vector of this class. Finally, we calculate the cosine similarity of the vectors as the measure of cross-class feature usage for each pair of two classes.

Algorithm 1: Feature Attribution Correlation Matrix

Input: A DNN classifier f with feature extractor g and linear layer W; **Test** dataset $D = \{D_y : y \in \mathcal{Y}\}$; Perturbation margin ϵ;
Output: A correlation matrix C measuring the cross-class feature usage

```
/* Record robust feature attribution                    */
```
for $y \in \mathcal{Y}$ **do**
  $A^y \leftarrow (0, \cdots, 0)$ `/* initialization as a n-dim vector */`
  for $x \in D_y$ **do**
    $\delta \leftarrow \arg\max_{\|\delta\| \leq \epsilon} \ell_{\text{CE}}(f(x+\delta), y)$ `/* untargeted PGD Attack */`
    A^y **+=** $g(x+\delta) \odot W[y]$ `/* point-wise multiplication */`
  $A^y \leftarrow A^y$ **/** $|D_y|$ `/* Average */`
for $1 \leq i, j \leq |\mathcal{Y}|$ **do**
  $C[i,j] \leftarrow \frac{A^i \cdot A^j}{\|A^i\|_2 \cdot \|A^j\|_2}$ `/* Cosine similarity */`
return C

B MORE FEATURE ATTRIBUTION CORRELATION MATRICES AT DIFFERENT EPOCHS

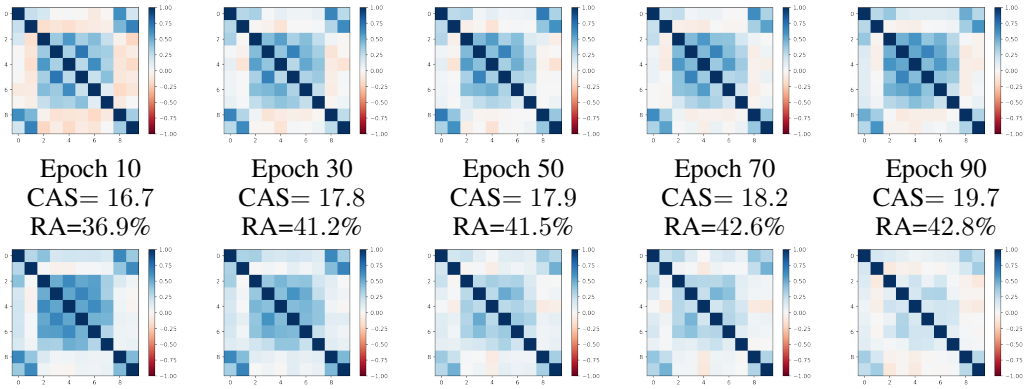


Figure 5: Feature attribution correlation matrices, and their corresponding robust accuracy (RA), CAS at different epochs.

We present more feature attribution correlation matrices at different epochs in Figure 5. The training detail is the same as that of our experiment (Section 5.2), and the test robust accuracy is plotted in Figure 1(b) (red line, $\epsilon = 8/255$). From the matrices we can see that at the initial stage of AT (10th - 90th Epochs), the model has already learned several cross-class features, and the overlapping effect of class-wise feature attribution achieves the highest at the 110th epoch among the shown matrices. However, for the later stages, where the model starts overfitting, this overlapping effect gradually vanishes, and the model tends to make decisions with fewer cross-class features.

C REGARDING EXTREMELY LARGE ϵ

While our interpretation is consistent with the fact that for practically used $\epsilon \in [0, 8/255]$, larger ϵ leads to more significant robust overfitting in AT, it is also compatible with the phenomenon of for extremely large $\epsilon(> 8/255)$, the effect of robust overfitting begins to decline (Wei et al., 2023a). We justify this below.

Recall that our main interpretation for robust overfitting is that during the initial stage of AT, the model learns both class-specific and cross-class features. As training progresses and the robust loss decreases, the model begins to forget cross-class features, which leads to robust overfitting. Regarding AT with extremely large ϵ , as we proved in Theorem 1, the more rigid robust loss makes the model even harder to learn cross-class features at the initial stage of AT. Given that fewer cross-class features are learned, the forgetting effect of these features is weakened, thus mitigating robust overfitting.

This claim is verified by the following study. We conduct additional experiments on AT with extremely large perturbation bounds $\epsilon = 12/255$ and $16/255$, and compare them with $\epsilon = 8/255$. We report the CAS and robust accuracy at the **10th, best, and last epochs** in the following table.

Table 2: Comparison of robust accuracy (RA) and CAS on AT with large ϵ.

Epoch	10	Best	Last
ϵ for AT	CAS / RA	CAS / RA	CAS / RA
$8/255$	$16.7/36.9\%$	$25.6/47.8\%$	$9.0/42.5\%$
$12/255$	$15.6/29.8\%$	$18.9/38.7\%$	$8.7/34.1\%$
$16/255$	$14.4/23.8\%$	$17.5/31.3\%$	$8.4/28.1\%$

The table shows that the CAS (usage of cross-class features) of large ϵ is less than that of $\epsilon = 8/255$ during the initial stage of AT (10th Epoch). This verifies our claim that the more rigid robust loss of large ϵ makes it even harder for the model to learn cross-class features at the initial stage of AT. Furthermore, the CAS of the best Epoch for large ϵ is significantly smaller than that of $\epsilon = 8/255$, further supporting our claim that these models struggle to learn cross-class features. Comparing the gap of CAS between the best and last epochs, we find that the gap for large ϵ is smaller than that of $\epsilon = 8/255$, which is consistent with the gap between the best and last robust accuracy. Therefore, we can conclude that the mitigation of robust overfitting with large ϵ can be explained by the less forgetting of cross-class features, which is compatible with our interpretation.

D MORE COMPARISON UNDER VARIOUS SETTINGS

D.1 COMPARISON ON MORE DATASETS

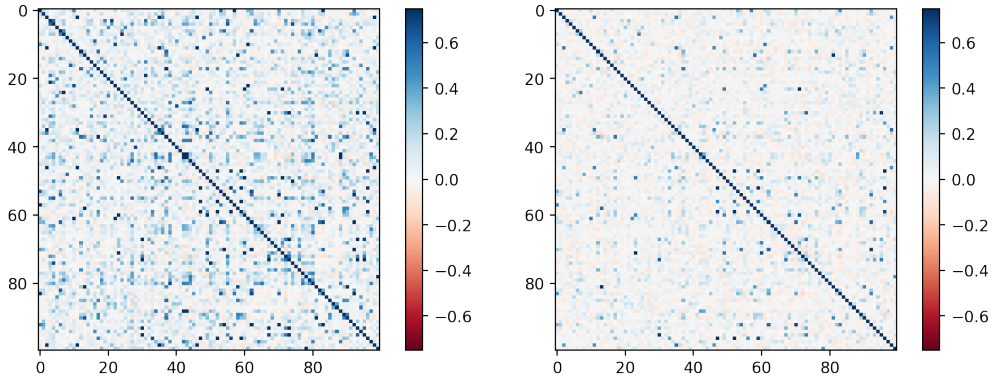
We illustrate the comparison of the feature attribution correlation matrices and the corresponding robust accuracy and CAS of the best checkpoint and the last checkpoint on the **CIFAR-100** and the **TinyImagenet** datasets in Figure 6 and Figure 7, respectively. We can see that there are still significant differences between matrices and CAS derived from the best and the last checkpoint of AT on other datasets.

D.2 COMPARISON ON ℓ_2-NORM AT

We show the comparison of the feature attribution correlation matrices of the best checkpoint and the last checkpoint of **ℓ_2-norm AT** ($\epsilon = 128/255$) on CIFAR-10 dataset in Figure 8 (a)(b). We can see that there are still significant differences between matrices and CAS derived from the best and the last checkpoint of ℓ_2-norm AT.

D.3 COMPARISON ON TRANSFORMER ARCHITECTURE

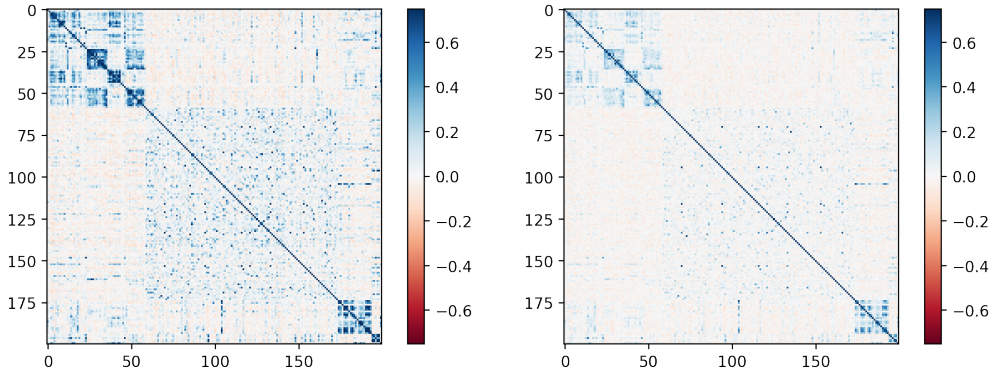
We show the comparison of the feature attribution correlation matrices of the best checkpoint and the last checkpoint of AT on CIFAR-10 dataset with **Vision Transformer architecture** (Deit-Ti Touvron



(a) CIFAR-100 Best
RA= 24.7%, CAS= 569

(b) CIFAR-100 Last
RA= 19.6%, CAS= 352

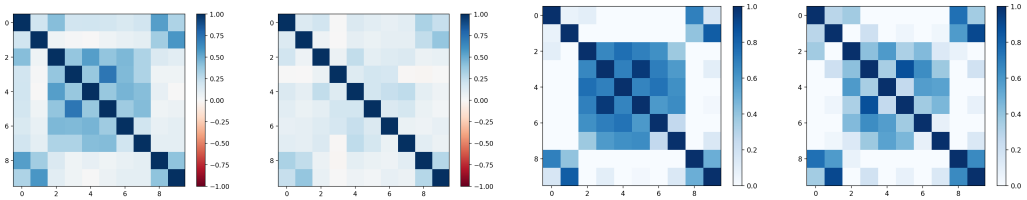
Figure 6: Feature attribution correlation matrices on CIFAR-100 dataset.



(a) TinyImagenet Best
RA= 18.0%, CAS= 1548

(b) TinyImagenet Last
RA= 14.4%, CAS= 998

Figure 7: Feature attribution correlation matrices on ℓ_2-norm AT and Visual Transformer architecture.



(a) ℓ_2-AT best (b) ℓ_2-AT last (c) DeiT-Ti best (d) DeiT-Ti last

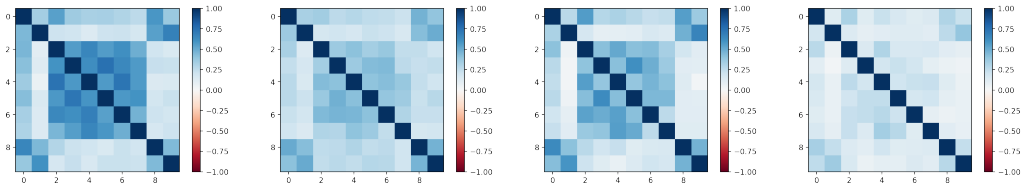
Figure 8: Feature attribution correlation matrices on ℓ_2-norm AT and Visual Transformer architecture.

et al. (2021)) in Figure 8 (c)(d). We can see that there are still significant differences between matrices and CAS derived from the best and the last checkpoint of AT with transformer architecture.

D.4 INSTANCE-WISE ANALYSIS

We also conduct a similar study by calculating the feature attribution correlation matrices for the best and the last checkpoints of ℓ_∞ and ℓ_2-AT and their corresponding CAS **instance-wisely**, and the results are shown in Figure 9. When considering classes i and j, for each sample x from class i, we identify its most similar counterpart x' from class j. We then calculate their cosine similarity and average the results over all samples in class i.

In this context, x' can be interpreted as the sample in class j that shares the most cross-class features with x among all samples in class j. This metric provides a meaningful way to quantify the utilization of cross-class features. We did attempt to average over all sample pairs (x, x') in classes i and j, but due to high variance among samples, each element in the correlation matrix C hovered near 0 throughout all epochs in adversarial training, rendering it unable to provide meaningful information.



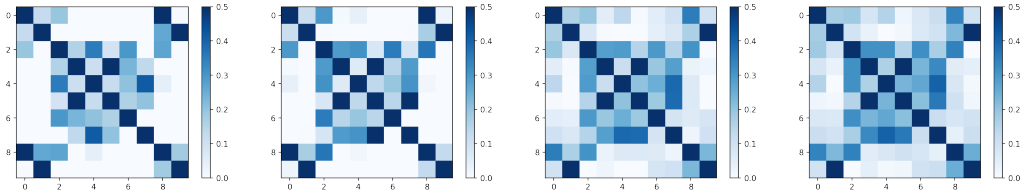
(a) ℓ_∞-AT best CAS=34.9 (b) ℓ_∞-AT last CAS=25.6 (c) ℓ_2-AT best CAS=27.0 (d) ℓ_2-AT last CAS=14.9

Figure 9: **Instance-wise** feature attribution correlation matrices

Consistent with the results for class-wise attribution vectors, it is still observed that there is a significant decrease in the usage of cross-class features from the best checkpoint to the last for both ℓ_∞ and ℓ_2-AT. This observation further substantiates our understanding of robust overfitting.

D.5 REGULAR TRAINING

We also extend our experimental scope to include regular training on the CIFAR-10 dataset. The experimental settings mirror those outlined in Section 5, with the sole distinction being the absence of perturbations in regular training. The results are shown in Figure 10. Specifically, considering that regular training prioritizes natural generalization and exhibits minimal robustness, we have calculated the feature attribution vectors using clean examples. These vectors were computed for epochs $\{50, 100, 150, 200\}$. Notably, the results reveal a lack of clear differences between them, particularly in the latter stages (150th and 200th), where the training tends to converge. This observation is consistent with the characteristic of regular training, which typically does not exhibit overfitting.

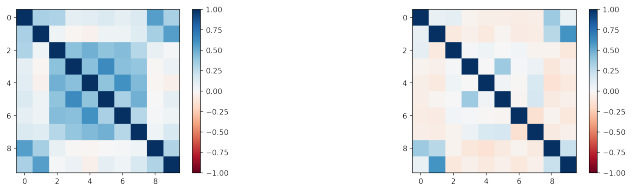


(a) Epoch 50 CAS=7.3 (b) Epoch 100 CAS=8.4 (c) Epoch 150 CAS=9.8 (d) Epoch 200 CAS=10.2

Figure 10: Feature attribution correlation matrices for **regular training** at different stages. Color bar scaled to $[0, 0.5]$.

D.6 CATASTROPHIC OVERFITTING IN FAST-AT

In addition to robust overfitting in adversarial training, there is also a phenomenon called *Catastrophic Overfitting* (Wong et al., 2020) observed in Fast (single step) adversarial training, where the model quickly decreases its robustness after a certain epoch of training. We also extend our investigations to include Fast-AT for the CIFAR-10 dataset, employing an ℓ_∞-norm perturbation bound of $\epsilon = 8/255$. The feature attribution correlation matrices before and after the catastrophic overfitting are shown in Figure 11. It is clear that after catastrophic overfitting, there is a significant reduction in the usage of cross-class features. This observation aligns with our understanding, indicating that the model also tends to forget cross-class features after exhibiting catastrophic overfitting.



(a) Before Catastrophic Overfitting (b) After Catastrophic Overfitting

Figure 11: Feature attribution correlation matrices for **Fast adversarial training** before and after catastrophic overfitting happens.

E PROOFS FOR THEOREMS

E.1 PRELIMINARIES

First we present some preliminaries, and then review the data distribution, the hypothesis space and the optimization objective.

Notations Let $\mathcal{N}(\mu, \sigma)$ be the normal distribution with mean μ and variance σ^2. We denote $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ and $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \mathrm{d}t = \Pr.(\mathcal{N}(0,1) < x)$ as its probability density function and distribution function.

Data distribution For $i \in \{1,2,3\}$, the sample of the i-th class is

$$(x_{E,1}, x_{E,2}, x_{E,3}, x_{C,1}, x_{C,2}, x_{C,3}) \in \mathbb{R}^6, \tag{12}$$

follows a distribution

$$\begin{cases} x_{E,j}|(y_i = j) \sim \mathcal{N}(\mu, \sigma^2) \\ x_{E,j}|(y_i \neq j) = 0 \end{cases}, \quad \begin{cases} x_{C,j}|(y_i \neq j) \sim \mathcal{N}(\mu, \sigma^2) \\ x_{C,j}|(y_i = j) = 0 \end{cases}, \tag{13}$$

and $\mu, \sigma > 0$. We also assume $\sigma < \sqrt{\pi}\mu$ to control the variance.

Hypothesis space The hypothesis space is $\{f_{\boldsymbol{w}} : \boldsymbol{w} = (w_1, w_2), w_1, w_2 \geq 0\}$ and $f_{\boldsymbol{w}}(x)$ calculates its i-th logit by

$$f_{\boldsymbol{w}}(\boldsymbol{x})_i = w_1 x_{E,i} + w_2(x_{C,j_1} + x_{C,j_2}), \quad \text{where} \quad \{j_1, j_2\} = \{1,2,3\} \backslash \{i\}. \tag{14}$$

Optimization objective Consider adversarially training $f_{\boldsymbol{w}}$ with ℓ_∞-norm perturbation bound $\epsilon < \frac{\mu}{2}$. We hope that given sample $x \sim \mathcal{D}_i$, under any perturbation $\{\delta : \|\delta\|_\infty \leq \epsilon\}$, the $f(x+\delta)_i$ is larger than any $f(x+\delta)_j$ as much as possible. We also add a regularization term $\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$ to the loss function.

Overall, the loss function can be formulated as

$$\mathcal{L}(f_{\boldsymbol{w}}) = \mathbb{E}_i[\mathbb{E}_{x \sim \mathcal{D}_i} \max_{\|\delta\|_\infty \leq \epsilon} (\max_{j \neq i} f_{\boldsymbol{w}}(x+\delta)_j - f_{\boldsymbol{w}}(x+\delta)_i)] + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2. \tag{15}$$

E.2 PROOF FOR THEOREM 1

Theorem 1 *There exists a $\epsilon_0 \in (0, \frac{1}{2}\mu)$, for AT by optimizing the robust loss (15) with $\epsilon \in (0, \epsilon_0)$, the output function obtains $w_2 > 0$; for AT with $\epsilon \in (\epsilon_0, \frac{1}{2}\mu)$, the output function returns $w_2 = 0$. By contrast, AT with $\epsilon \in (0, \frac{1}{2}\mu)$ always obtains $w_1 > 0$.*

To prove Theorem 1, we need the following lemmas.

Lemma 1 *Suppose that $X, Y \sim \mathcal{N}(0, 1)$, and they are independent. Let $Z = \max\{X, Y\}$, then $\mathbb{E}[Z] = \frac{1}{\sqrt{\pi}}$.*

proof. Let $p(\cdot)$ and $F(\cdot)$ be the probability density function and distribution function of Z, respectively. Then, for any $z \in \mathbb{R}$,

$$F(z) = \Pr(Z < z) = \Pr(\max\{X, Y\} < z) = \Pr(X < z) \cdot \Pr(Y < z) = \Phi^2(z), \tag{16}$$

and we have

$$p(z) = F'(z) = [\Phi^2(z)]' = 2\phi(z)\Phi(z). \tag{17}$$

Thus,

$$\begin{aligned}
\mathbb{E}[Z] &= \int_{-\infty}^{+\infty} 2z\phi(z)\Phi(z)dz \\
&= 2\int_{-\infty}^{+\infty} z \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} \left(\int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt\right) dz \\
&= -\frac{1}{\pi}\int_{-\infty}^{+\infty} \left(\int_{-\infty}^{z} e^{-\frac{t^2}{2}} dt\right) d(e^{-\frac{z^2}{2}}) \\
&= -\frac{1}{\pi}[e^{-\frac{z^2}{2}} \int_{-\infty}^{z} e^{-\frac{t^2}{2}} dt]_{-\infty}^{+\infty} + \frac{1}{\pi}\int_{-\infty}^{+\infty} e^{-\frac{z^2}{2}} e^{-\frac{z^2}{2}} dz \\
&= 0 + \frac{1}{\pi}\int_{-\infty}^{+\infty} e^{-z^2} dz = \frac{1}{\sqrt{\pi}}.
\end{aligned} \tag{18}$$

Lemma 2 *Given $x = (x_{E,1}, x_{E,2}, x_{E,3}, x_{C,1}, x_{C,2}, x_{C,3}) \sim \mathcal{D}_1$, $\epsilon \in (0, \frac{\mu}{2})$ and $\boldsymbol{w} = (w_1, w_2)$, then $\delta = (-\epsilon, \epsilon, \epsilon, \epsilon, -\epsilon, -\epsilon)$ is a solution for $\delta = \arg\max_{\|\delta\|_\infty \le \epsilon} [\max_{j \neq 1} f_{\boldsymbol{w}}(x + \delta)_j - f_{\boldsymbol{w}}(x + \delta)_1]$.*

proof. Denote $\delta = (\delta_{E,1}, \delta_{E,2}, \delta_{E,3}, \delta_{C,1}, \delta_{C,2}, \delta_{C,3})$. Note that for $x \sim \mathcal{D}_1$, we have $x_{E,2} = x_{E,3} = x_{C,1} = 0$. Then,

$$\begin{aligned}
&\max_{j \neq 1} f_{\boldsymbol{w}}(x + \delta)_j - f_{\boldsymbol{w}}(x + \delta)_1 \\
=& \max_{j \in \{2,3\}} [w_1\delta_{E,2} + w_2\delta_{C,1} + w_2(x_{C,3} + \delta_{C,3}),\ w_1\delta_{E,3} + w_2\delta_{C,1} + w_2(x_{C,2} + \delta_{C,2})] \\
&- w_1(x_{E,1} + \delta_{E,1}) - w_2(x_{C,2} + \delta_{C,2} + x_{C,3} + \delta_{C,3}) \\
=& w_2\delta_{C,1} + \max_{j \in \{2,3\}} [w_1\delta_{E,2} + w_2(x_{C,3} + \delta_{C,3}),\ w_1\delta_{E,3} + w_2(x_{C,2} + \delta_{C,2})] \\
&- w_1(x_{E,1} + \delta_{E,1}) - w_2(x_{C,2} + \delta_{C,2} + x_{C,3} + \delta_{C,3}).
\end{aligned} \tag{19}$$

Since $w_1, w_2 \ge 0$, it is clear that $\delta_{E,1} = -\epsilon$, $\delta_{E,2} = \delta_{E,3} = \delta_{C,1} = \epsilon$ are the optimal values for maximizing (19). As for $\delta_{C,2}$ and $\delta_{C,3}$, to prove that $\delta_{C,2} = \delta_{C,2} = -\epsilon$ are the optimal values, by variable simplification ($a' = \delta_{C,2}, b' = \delta_{C,3}$) and dividing by w_2 we only need to show that

$$\max\{a + a', b + b'\} - a' - b' \le \max\{a - \epsilon, b - \epsilon\} + 2\epsilon \tag{20}$$

under the constraint $|a'| \le \epsilon$ and $|b'| \le \epsilon$. Note that (20) is equivalent to

$$\begin{aligned}
&\max\{a + a', b + b'\} - a' - b' \le \max\{a, b\} + \epsilon \\
\Leftrightarrow &\max\{a + a', b + b'\} \le \max\{a, b\} + a' + b' + \epsilon \\
\Leftrightarrow &\max\{a + a', b + b'\} \le \max\{a + a' + b' + \epsilon, b + a' + b' + \epsilon\}.
\end{aligned} \tag{21}$$

Since $|b'| \leq \epsilon$, we have $b'+\epsilon \geq 0$ and hence $a+a' \leq a+a'+b'+\epsilon \leq \max\{a+a'+b'+\epsilon, b+a'+b'+\epsilon\}$. Similarly, $b+b' \leq \max\{a+a'+b'+\epsilon, b+a'+b'+\epsilon\}$ and finally we have $\max\{a+a', b+b'\} \leq \max\{a+a'+b'+\epsilon, b+a'+b'+\epsilon\}$. Clearly when $a' = b' = -\epsilon$, the equal sign holds.

Proof for Theorem 1. First, due to symmetry, optimizing (15) is equivalent to optimize

$$\mathbb{E}_{x\sim\mathcal{D}_1}[\max_{\|\delta\|_\infty\leq\epsilon}(\max_{j\neq 1} f_{\boldsymbol{w}}(x+\delta)_j - f_{\boldsymbol{w}}(x+\delta)_1)] + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2. \tag{22}$$

Further, by Lemma 2 we can replace δ with its optimal value and transform the optimization objective above as

$$\mathbb{E}_{\hat{x}\sim\hat{\mathcal{D}}_1}(\max_{j\neq i} f_{\boldsymbol{w}}(\hat{x})_j - f_{\boldsymbol{w}}(\hat{x})_i)] + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2, \tag{23}$$

where $\hat{\mathcal{D}}_1$ is the **adversarial data distribution**:

$$\hat{x}_{E,j} \sim \begin{cases}\mathcal{N}(\mu-\epsilon,\sigma^2), & j=1\\ \epsilon, & j\neq 1\end{cases}, \quad \hat{x}_{C,j} \sim \begin{cases}\mathcal{N}(\mu-\epsilon,\sigma^2), & j\neq 1\\ \epsilon, & j=1\end{cases}. \tag{24}$$

Now we calculate the expectation in (23).

$$\begin{aligned}
&\mathbb{E}_{\hat{x}\sim\hat{\mathcal{D}}_1}[(\max f_{\boldsymbol{w}}(\hat{x})_j - f_{\boldsymbol{w}}(\hat{x})_i)] + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2\\
=&\mathbb{E}_{\hat{x}\sim\hat{\mathcal{D}}_1}[\max(w_1\epsilon + w_2\epsilon + w_2\hat{x}_{C,3},\ w_1\epsilon + w_2\epsilon + w_2\hat{x}_{C,2}) - w_1\hat{x}_{E,1} - w_2(\hat{x}_{C,2}+\hat{x}_{C,3})] + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2\\
=&\mathbb{E}_{\hat{x}\sim\hat{\mathcal{D}}_1}[w_1\epsilon + w_2\epsilon + w_2\max(\hat{x}_{C,3},\ \hat{x}_{C,2}) - w_1\hat{x}_{E,1} - w_2(\hat{x}_{C,2}+\hat{x}_{C,3})] + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2\\
=&w_1\epsilon + w_2\epsilon + w_2\mathbb{E}_{\hat{x}\sim\hat{\mathcal{D}}_1}[\max(\hat{x}_{C,3},\ \hat{x}_{C,2})] + \mathbb{E}_{\hat{x}\sim\hat{\mathcal{D}}_1}[-w_1\hat{x}_{E,1} - w_2(\hat{x}_{C,2}+\hat{x}_{C,3})] + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2\\
=&w_1\epsilon + w_2\epsilon + w_2\mathbb{E}_{\hat{x}\sim\hat{\mathcal{D}}_1}[\max(\hat{x}_{C,3},\ \hat{x}_{C,2})] + [-w_1(\mu-\epsilon) - 2w_2(\mu-\epsilon)] + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2.
\end{aligned} \tag{25}$$

Finally, since $\hat{x}_{C,3},\ \hat{x}_{C,2} \sim (\mu-\epsilon,\sigma^2)$ and they are independent, by Lemma 1 we have

$$\mathbb{E}[\max(\frac{\hat{x}_{C,3}-(\mu-\epsilon)}{\sigma}, \frac{\hat{x}_{C,2}-(\mu-\epsilon)}{\sigma})] = \frac{1}{\sqrt{\pi}}, \tag{26}$$

hence $\mathbb{E}_{\hat{x}\sim\hat{\mathcal{D}}_1}[\max(\hat{x}_{C,3},\ \hat{x}_{C,2})] = \mu - \epsilon + \frac{\sigma}{\sqrt{\pi}}$.

Therefore, the optimizing objective can be simplified as

$$\mathcal{L}(f_{\boldsymbol{w}}) = (-\mu+2\epsilon)w_1 + (-\mu+2\epsilon+\frac{\sigma}{\sqrt{\pi}})w_2 + \frac{\lambda}{2}(w_1^2+w_2^2). \tag{27}$$

For w_2, we have

$$\frac{\partial\mathcal{L}}{\partial w_2} = -\mu+2\epsilon+\frac{\sigma}{\sqrt{\pi}} + \lambda w_2. \tag{28}$$

Recall that $\sigma < \sqrt{\pi}\mu$. Let $\epsilon_0 = \frac{1}{2}(\mu - \frac{\sigma}{\sqrt{\pi}}) \in (0, \frac{\mu}{2})$. By analysing the sign of (28), it is clear that for $\epsilon \in (0,\epsilon_0)$, the optimal w_2 for minimizing the loss function (27) is

$$w_2 = \frac{\mu - 2\epsilon - \frac{\sigma}{\sqrt{\pi}}}{\lambda}. \tag{29}$$

However, for $\epsilon \in (\epsilon_0, \frac{\mu}{2})$, $\frac{\partial\mathcal{L}}{\partial w_2}$ is always negative, thus the returned w_2 by AT is $w_2 = 0$ under the constraint $w_2 \geq 0$.

By contrast,

$$\frac{\partial\mathcal{L}}{\partial w_1} = -\mu+2\epsilon+\lambda w_1, \tag{30}$$

and for $\epsilon \in (0, \frac{\mu}{2})$, the optimal w_1 for minimizing the loss function (27) is always positive:

$$w_1 = \frac{\mu - 2\epsilon}{\lambda} > 0. \tag{31}$$

This ends our proof.

E.3 Proof for Theorem 2

Theorem 2 *For any $w_1 > 0$ and $\epsilon \in (0, \frac{\mu}{2})$, if $w_2 \in [0, w_1]$, a larger w_2 increases the possibility of the model distinguishing the adversarial examples from any other given class.*

To prove Theorem 2, we need the following lemma.

Lemma 3 *Suppose that $X, Y \sim \mathcal{N}(1, \sigma_1^2)$ and they are independent, $\sigma_1 > 0$. Let $Z_t = X + tY$ where $t > 0$. Denote $u(t) = \Pr(Z_t > 0)$, then $u(t)$ is monotonically increasing at t for $t \in [0, 1]$.*

proof. Note that $Z_t = X + tY \sim \mathcal{N}(1 + t, (1 + t^2)\sigma_1^2)$. Thus, the distribution function of Z_t is $\Phi_t(z) = \Phi(\frac{z-1-t}{\sqrt{1+t^2}\sigma_1})$, and

$$\begin{aligned} u(t) &= 1 - \Phi_t(0) = 1 - \Phi(\frac{-1-t}{\sqrt{1+t^2}\sigma_1}) = \Phi(\frac{1+t}{\sqrt{1+t^2}\sigma_1}), \\ u'(t) &= p(\frac{1+t}{\sqrt{1+t^2}\sigma_1}) \frac{\sqrt{1+t^2}\sigma_1 - (1+t)\frac{t\sigma_1}{\sqrt{1+t^2}}}{(1+t^2)\sigma_1^2} = p(\frac{1+t}{\sqrt{1+t^2}\sigma_1}) \frac{(1+t^2) - (1+t)t}{(1+t^2)\sqrt{1+t^2}\sigma_1} \\ &= p(\frac{1+t}{\sqrt{1+t^2}\sigma_1}) \frac{1-t}{(1+t^2)\sqrt{1+t^2}\sigma_1}. \end{aligned} \tag{32}$$

Therefore, for $t \in (0, 1)$, $u'(t) > 0$ and $u(t)$ is monotonically increasing at t for $t \in [0, 1]$.

Proof for Theorem 2. Due to symmetry, it's suffice to show that given w_1, for $w_2 \in [0, w_1]$, the probability

$$\Pr(f_{\boldsymbol{w}}(\hat{x})_1 > f_{\boldsymbol{w}}(\hat{x})_2), \quad \hat{x} \sim \hat{\mathcal{D}}_1 \tag{33}$$

is monotonically increasing at w_2. Note that

$$\begin{aligned} &f_{\boldsymbol{w}}(\hat{x})_1 - f_{\boldsymbol{w}}(\hat{x})_2 = w_1(\hat{x}_{E,1} - \hat{x}_{E,2}) + w_2(\hat{x}_{C,2} - \hat{x}_{C,1}), \\ &\hat{x}_{E,1} - \hat{x}_{E,2} \sim \mathcal{N}(\mu - 2\epsilon, 2\sigma^2), \\ &\hat{x}_{C,2} - \hat{x}_{C,1} \sim \mathcal{N}(\mu - 2\epsilon, 2\sigma^2). \end{aligned} \tag{34}$$

By dividing $w_1 \cdot (\mu - 2\epsilon)$, and let $t = \frac{w_2}{w_1}$, $X = \frac{\hat{x}_{E,1} - \hat{x}_{E,2}}{\mu - 2\epsilon}$ and $Y = \frac{\hat{x}_{C,2} - \hat{x}_{C,1}}{\mu - 2\epsilon}$, from Lemma 3 we know that the probability

$$\Pr(f_{\boldsymbol{w}}(\hat{x})_1 - f_{\boldsymbol{w}}(\hat{x})_2 > 0) \tag{35}$$

is monotonically increasing at $t = \frac{w_2}{w_1}$, and hence increasing at w_2. This ends our proof.

E.4 Proof for Theorem 3 and Corollary 1

Simplification of knowledge distillation as label smoothing. In this context, the term 'symmetry' specifically refers to the symmetry of logits for the other two classes when taking the expectation in the loss function (equation 10). When considering data from class y, both the distribution of features $x_{E,i}$ and x_{C_i} for the other two classes, as well as their respective weights w_1 and w_2, exhibit symmetry respectively. Consequently, after applying knowledge distillation, the expectation for logits of the other two classes in the objective loss function (equation 10) becomes identical. To simplify this process, we can employ label smoothing.

We prove Theorem 3 and Corollary 1 in the following. Recall that we define the robust loss under knowledge distillation as

$$\mathcal{L}_{\text{LS}}(f_{\boldsymbol{w}}) = \mathbb{E}_i\{\mathbb{E}_{x \sim \mathcal{D}_i}(1-\beta)[\max_{\|\delta\|_\infty \le \epsilon}(\max_{j \neq i} f_{\boldsymbol{w}}(x+\delta)_j - f_{\boldsymbol{w}}(x+\delta)_i)] - \frac{\beta}{2}\sum_{j \neq i} f_{\boldsymbol{w}}(x+\delta)_j\} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2. \tag{36}$$

Theorem 3 Consider AT with knowledge distillation loss (36). There exists an $\epsilon_1 > \epsilon_0$, such that for $\epsilon \in (0, \epsilon_1)$, the output function obtains $w_2 > 0$; for $\epsilon \in (\epsilon_1, \frac{1}{2}\mu)$, the output function returns $w_2 = 0$.

Proof for Theorem 3. Similar to the proof for Theorem 1, the optimization objective (36) can be simplified as

$$\begin{aligned}\mathcal{L}_{\text{LS}}(f_{\boldsymbol{w}}) &= (1-\beta)[(-\mu+2\epsilon)w_1 + (-\mu+2\epsilon+\frac{\sigma}{\sqrt{\pi}})w_2] - \beta[\epsilon w_1 + \mu w_2] + \frac{\lambda}{2}(w_1^2+w_2^2) \\ &= [(1-\beta)\mu + (2-3\beta)\epsilon]w_1 + [(1-\beta)(2\epsilon+\frac{\sigma}{\sqrt{\pi}}) - \mu]w_2 + \frac{\lambda}{2}(w_1^2+w_2^2).\end{aligned} \tag{37}$$

Thus

$$\frac{\mathcal{L}_{\text{LS}}}{w_2} = (1-\beta)(2\epsilon+\frac{\sigma}{\sqrt{\pi}}) - \mu + \lambda w_2, \tag{38}$$

and let $\epsilon_1 = \frac{1}{2}(\frac{\mu}{1-\beta} - \frac{\sigma}{\sqrt{\pi}}) > \epsilon_0$, similar to the analysis for ϵ_0, we have for $\epsilon \in (0, \epsilon_1)$, the output function obtains $w_2 > 0$; for $\epsilon \in (\epsilon_1, \frac{1}{2}\mu)$, the output function returns $w_2 = 0$. This ends our proof.

Corollary 1 Let $w_2^*(\epsilon)$ be the value of w_2 returned by AT with (15), and $w_2^{\text{LS}}(\epsilon)$ be the value of w_2 returned by label smoothed loss (36). Then, for $\epsilon \in (0, \epsilon_1)$, we have $w_2^{\text{LS}}(\epsilon) > w_2^*(\epsilon)$.

Proof for Corollary 1. For $\epsilon \in (0, \epsilon_1)$, by analysing the sign of (38), we have

$$w_2^{\text{LS}}(\epsilon) = \frac{\mu - (1-\beta)(2\epsilon+\frac{\sigma}{\sqrt{\pi}})}{\lambda}, \tag{39}$$

and recall that in the proof for Theorem 1 we have

$$w_2^*(\epsilon) = \frac{\mu - (2\epsilon+\frac{\sigma}{\sqrt{\pi}})}{\lambda}, \tag{40}$$

thus it is clear that

$$w_2^{\text{LS}}(\epsilon) - w_2^*(\epsilon) = \frac{\beta(2\epsilon+\frac{\sigma}{\sqrt{\pi}})}{\lambda} > 0. \tag{41}$$

This ends our proof.

F More Experiments of WAKE

We also conduct experiments comparing WAKE and baselines in other settings to further demonstrate its effectiveness in terms of improving robustness and mitigating robust overfitting.

F.1 ℓ_2-norm AT

We conduct experiments on ℓ_2-norm AT with $\epsilon = 128/255$ on CIFAR-10 dataset. The settings are the same as those of CIFAR-10 in Section 5.2. The results are shown in Table 3.

Table 3: Comparison of WAKE with vanilla AT and AT+KDSWA on ℓ_2-norm.

Dataset	Method	Robust Acc. (%)		Clean Acc. (%)	
		Best	Last	Best	Last
CIFAR-10	AT	67.3	64.5	88.6	88.7
	AT + KDSWA	68.9	68.3	89.4	89.7
	AT + WAKE	**70.4**	**70.2**	**89.9**	**90.1**

Results clearly show the advantage of WAKE over vanilla AT and KDSWA in terms of mitigating robust overfitting for ℓ_2-norm AT.

F.2 COMBINATION WITH OTHER METHODS

We conduct experiments on TRADES (Zhang et al., 2019) on CIFAR-10 dataset. The settings are the same as those of CIFAR-10 in Section 5.2. We follow the hyperparameters of TRADES from its original papers. The results are shown in Table 4.

Table 4: Comparison of WAKE with vanilla AT and AT+KDSWA combined with TRADES.

Dataset	Method	Robust Acc. (%) Best	Robust Acc. (%) Last	Clean Acc. (%) Best	Clean Acc. (%) Last
CIFAR-10	TRADES	48.3	46.9	82.5	83.7
	TRADES + KDSWA	50.1	49.5	82.9	83.3
	TRADES + WAKE	**50.7**	**50.4**	**83.8**	**84.1**

Consistent with the results in the paper, these results clearly show that WAKE can also be combined with other advanced methods to further mitigate robust overfitting and improve adversarial robustness.

F.3 TRANSFORMER ARCHITECTURE

We conduct experiments on transformer architecture DeiT-Ti (Touvron et al., 2021) on CIFAR-10 dataset. The settings are the same as those of CIFAR-10 in Section 5.2, and the robustness is evaluated using PGD-20. The results are shown in Table 5.

Table 5: Comparison of WAKE with vanilla AT and AT+KDSWA on DeiT-Ti architecture.

Dataset	Method	Robust Acc. (%) Best	Robust Acc. (%) Last	Clean Acc. (%) Best	Clean Acc. (%) Last
CIFAR-10	AT	50.0	47.7	79.4	79.6
	AT + KDSWA	50.4	49.5	79.6	79.8
	AT + WAKE	**50.6**	**50.3**	**80.1**	**80.4**

Consistent with the results in the paper, these results clearly show that WAKE can also work on transformer architecture to further mitigate robust overfitting and improve adversarial robustness.

G DETAILS FOR WAKE IMPLEMENTATION

G.1 ALGORITHM FOR WAKE

We provide a detailed implementation algorithm of WAKE in Algrithm 2.

G.2 DETAILS OF THE KNOWLEDGE DISTILLATION FUNCTION

Following Chen et al. (2021), the knowledge distillation (Hinton et al., 2015) function can be defined as:

$$\mathcal{KD}(f(\boldsymbol{\theta}_{\text{student}}; x), f(\boldsymbol{\theta}_{\text{teacher}}; x)) = \text{KL}[\text{softmax}(\frac{f(\boldsymbol{\theta}_{\text{student}}; x)}{T}), \text{softmax}(\frac{f(\boldsymbol{\theta}_{\text{teacher}}; x)}{T})], \quad (42)$$

where $\text{KL}(\cdot, \cdot)$ is the Kullback-Leibler divergence and T is the distillation temperature.

H ADDITIONAL RELATED WORK

Algorithm 2: **W**eight **A**verage guided **K**nowledg**E** Distillation (WAKE)

Input: A DNN classifier $f_{\boldsymbol{\theta}}(\cdot)$ with parameter $\boldsymbol{\theta}$; Train dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$; Batch size m; Initial perturbation margin ϵ; Train epochs N; Learning rate η; Weight average decay rate α; Knowledge distillation warm-up start epoch t_s and end epoch t_e; hyper-parameter λ.

Output: A robust classifier $f_{\bar{\boldsymbol{\theta}}}$ with less overfitting

for $t \leftarrow 1, 2, \cdots, T$ **do**
  for *Every minibatch (x,y) in trainset* D **do**
    $\delta \leftarrow \max_{\|\delta\|_p \leq \epsilon} \ell_{\text{CE}}(f(\boldsymbol{\theta}, x+\delta), y)$;
    if $t > t_s$ **then**
      $\tilde{y} \leftarrow f(\bar{\boldsymbol{\theta}}, x+\delta)$(stop gradient);
      if $t < t_e$ **then**
        $\lambda_t \leftarrow \frac{t-t_s}{t_e-t_s} \cdot \lambda$;
      else
        $\lambda_t \leftarrow 1$;
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} [(1-\lambda_t)\ell_{\text{CE}}(f(\boldsymbol{\theta}, x+\delta), y) + \lambda_t \cdot \mathcal{KD}(f(\boldsymbol{\theta}; x+\delta), \tilde{y}))].$;
    else
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} [\ell_{\text{CE}}(f(\boldsymbol{\theta}, x+\delta), y)]$;
    if $t < t_e$ **then**
      $\bar{\boldsymbol{\theta}} \leftarrow \alpha \bar{\boldsymbol{\theta}} + (1-\alpha)\boldsymbol{\theta}$;

return $f_{\bar{\boldsymbol{\theta}}}$;

H.1 Adversarial Training and Adversarial Robustness

The adversarial robustness and adversarial training has become popular research topic since the discovery of adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014), which uncovers that DNNs can be easily fooled to make wrong decisions by adversarial examples that are crafted by adding small perturbations to normal examples. The malicious adversaries can conduct adversarial attacks by crafting adversarial examples, which cause serious safety concerns regarding the deployment of DNNs. Following this discovery, various types of adversarial attack methods have been proposed, including gradient-based (Carlini & Wagner, 2017b; Liu et al., 2022), query-based (Andriushchenko & Flammarion, 2020; Bai et al., 2019), decision-based (Brendel et al., 2017; Chen et al., 2020) and demonstration-based (Wang et al., 2023; Wei et al., 2023b) attacks on various models and tasks.

In response to such adversarial threats, numerous defense approaches have also been proposed, such as adversarial example detection (Grosse et al., 2017; Tian et al., 2018) and purification (Bai et al., 2019; Nie et al., 2022), parameter regularization (Jakubovitz & Giryes, 2018; Wei et al., 2023c), randomized smoothing (Cohen et al., 2019; Levine & Feizi, 2020), among which adversarial training methods (Madry et al., 2017; Wang et al., 2019) has been considered as the most promising defending method against adversarial attacks (Carlini & Wagner, 2017a; Athalye et al., 2018). Through this research thread, there are also other perspectives on improving adversarial training, including architecture design (Huang et al., 2021; Mo et al., 2022), data augmentation (Rebuffi et al., 2021b;a), optimization objective design (Wang et al., 2020; Pang et al., 2022).

H.2 Adversarial Robustness Distillation

Besides adversarial training, there are also several papers on distilling adversarial robustness from teacher models (Goldblum et al., 2020; Zhu et al., 2021; Zi et al., 2021; Huang et al., 2023; Yue et al., 2023). Similar to conventional knowledge distillation (Hinton et al., 2015; Gou et al., 2021), this thread works toward training an adversarially robust student model with a robust teacher model. By designing proper distilling objectives and algorithms, these works can enhance the robustness of the trained student model.

There are several differences between our proposed WAKE method and these adversarial robustness distillation methods. First, WAKE is designed to mitigate robust overfitting in adversarial training, which is different from existing work typically for improving adversarial robustness. To the best of

our knowledge, the KDSWA (Chen et al., 2021) is the only existing distillation method designed for the same purpose, thus we only include KDSWA and the vanilla adversarial training method as baselines in experiments. Moreover, WAKE uses the weight-averaged model as the teacher model, which does not require a given robust teacher model. Therefore, not only WAKE can save large amounts of computational resources, but also its robustness is not dependent on another teacher model.