Appendix A. Proof

A.1. Proof for Theorem 1

Proof. We denote the set $S_R = \{(X,Y) | \forall (X,Y) \sim D, \exists X' \in \mathcal{B}(X,\epsilon) \ s.t. \ f_\theta(X')Y \leq 0\}$ and $S_{CaliR} = \{(X,Y) | \forall (X,Y) \sim D, \exists X' \in \mathcal{B}(X,\epsilon) \ s.t. \ f_\theta(X') f_{oracle}(X') \leq 0\}$.

Since $S_R \subseteq S_{CaliR} \implies \mathcal{R}_{rob}(f) \leq \mathcal{R}_{cali}(f)$**, we only need to prove** $S_R \subseteq S_{CaliR}$.

$$\begin{aligned}
&\forall (\mathbf{X},\mathbf{Y}) \in \mathbf{S_R},\\
&(1) \ \textbf{if } \mathbf{f_\theta(X)Y \leq 0}, then \ f_\theta(X) f_{oracle}(X) \leq 0 \implies X \in S_{CaliR}.\\
&(2) \ \textbf{if } \mathbf{f_\theta(X)Y > 0}, then \ \exists X' \in \mathcal{B}(X,\epsilon) \ s.t \ f_\theta(X')Y \leq 0;\\
&\quad \mathbf{1)if \ f_{oracle}(X')f_\theta(X') \leq 0} \implies X \in S_{CaliR}\\
&\quad \mathbf{2)if \ f_{oracle}(X')f_\theta(X') > 0}, \ then \ it \ must \ have:\\
&\qquad \exists X'' \in \mathcal{B}(X,\epsilon) \ s.t. f_\theta(X'') f_{oracle}(X'') \leq 0; \qquad (15)
\end{aligned}$$

We prove Eq. 15 by the contradiction method. We assume:

$$\forall X'' \in \mathcal{B}(X,\epsilon) \ s.t. f_\theta(X'') f_{oracle}(X'') > 0 \ is \ True. \qquad (16)$$

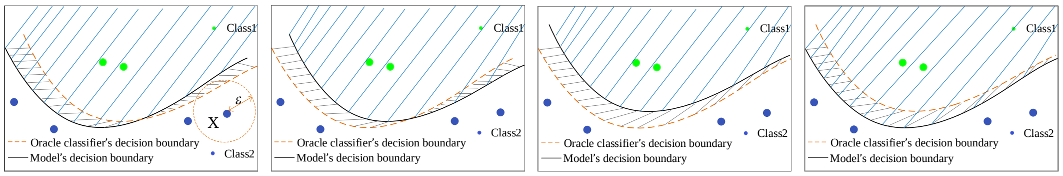
$$\begin{aligned}
&\mathbf{f_\theta(X)Y > 0, f_\theta(X')Y \leq 0} \implies the \ decision \ boundary \ of \ f_\theta\\
&\qquad\qquad crosses \ the \ \epsilon - norm \ ball \ of \ X.\\
&\mathbf{f_\theta(X')Y \leq 0, f_{oracle}(X')f_\theta(X') > 0} \implies the \ decision \ boundary\\
&\qquad\qquad of \ f_{oracle} \ crosses \ the \ \epsilon - norm \ ball \ of \ X.
\end{aligned}$$

If Eq.16 is true, *which implies that f_θ and f_{oracle} have the same prediction on any sample from the ϵ − ball of X.*
$\implies$ *the decision boundaries of f_θ and f_{oracle} will be completely overlapped in ϵ − ball of X,* **which contradicts** *the assumption of the Theorem 1 : the decision boundaries of f_θ and f_{oracle} are not overlapped.*
Therefore Eq.16 is False .
$\implies \exists X'' \in \mathcal{B}(X,\epsilon) \ s.t. f_\theta(X'') f_{oracle}(X'') \leq 0;$ *Eq. 15 is proved.*
$\implies X \in S_{CaliR}$.

By now, we proved $\forall X \in S_R \implies X \in S_{CaliR}$. Besides, $\exists X \in S_{CaliR} \implies X \notin S_R$, e.g. the sample X in Fig. 7(*a*)subfigure. Therefore $S_R \subseteq S_{CaliR}$ is proved. □

Besides going through a formal proof itself, we think it is useful to look into the provided visualization of the decision boundary for a more intuitive understanding. According the spatial relationship of decision boundaries of f_θ and f_{oracle}, it can be separated into intersection and non-intersection cases (no overlap case according to the assumption in

Theorem 1), which are showed in Fig. 3. From Fig. 3, for any sample (X,Y) from class 2, if $\exists X' \in \mathcal{B}(X,\epsilon)$ lies in the region filled with blue lines, it must have $\exists X'' \in \mathcal{B}(X,\epsilon)$ lies in the region filled with gray lines. However, if $\exists X' \in \mathcal{B}(X,\epsilon)$ lies in the region filled with gray lines, it is possible that $\forall X' \in \mathcal{B}(X,\epsilon)$ do not lie in the region filled with blue lines. Therefore $S_R \subseteq S_{CaliR}$.


Class1
Class2
X
ε
Oracle classifier's decision boundary
Model's decision boundary

(a) Intersect1 (b) Intersect2 (c) Non-Intersect1 (d) Non-Intersect2

Figure 7: Visualization of f_θ and f_{oracle} decision boundaries. Region filled with gray lines: $\{X'|f_\theta(X')f_{oracle}(X') \leq 0\}$. Region filled with blue lines: $\{X'|f_\theta(X')Y \leq 0, Y = class2\}$.

A.2. Proof for Theorem 2

Theorem 2 (Upper Bound) *Let ψ be a nondecreasing, continuous and convex function:$[0,1] \rightarrow [0,\infty]$. Let $\mathcal{R}_\phi(f) := \mathbf{E}\phi(f_\theta(X)Y)$ and $\mathcal{R}_\phi^* := \min_f \mathcal{R}_\phi(f)$, $\mathcal{R}(f) := \mathbf{E}(f_\theta(X)Y)$ and $\mathcal{R}^* = \min_f \mathcal{R}(f)$. For any non-negative loss function ϕ such that $\phi(0) \geq 1$, any measurable $f_\theta : \mathcal{X} \rightarrow \mathbf{R}$ and any probability distribution on $\mathcal{X} \times \{+1,-1\}$, we have:*

$$\mathcal{R}_{cali}(f) - \mathcal{R}^* \leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*) + \mathbf{E}\Big[\max_{\substack{X' \in \mathbf{B}(X,\epsilon) \\ f_{oracle}(X')=Y}} \phi(f_\theta(X')Y)\Big]. \tag{6}$$

Proof.

$$\begin{aligned}
&\mathcal{R}_{cali}(f) - \mathcal{R}^* = \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathcal{B}(X,\epsilon)\ s.t.\ f_\theta(X')f_{oracle}(X') \leq 0\} \\
&= \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathcal{B}(X,\epsilon)\ s.t.\ f_\theta(X')f_{oracle}(X') \leq 0, f_\theta(X)Y \leq 0\} \\
&\qquad + \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathcal{B}(X,\epsilon)\ s.t.\ f_\theta(X')f_{oracle}(X') \leq 0, f_\theta(X)Y > 0\} - \mathcal{R}^* \\
&= \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{f_\theta(X)Y \leq 0\} - \mathcal{R}^* \\
&\qquad + \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathcal{B}(X,\epsilon)\ s.t.\ f_\theta(X')f_{oracle}(X') \leq 0, f_\theta(X)Y > 0\} \\
&\leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*) + \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathcal{B}(X,\epsilon)\ s.t.\ f_\theta(X')f_{oracle}(X') \leq 0, f_\theta(X)Y > 0\} \\
&\leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*) + \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathcal{B}(X,\epsilon)\ s.t.\ f_\theta(X')f_{oracle}(X') \leq 0\} \\
&\leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*) + \mathbf{E}_{(X,Y)\sim D} \max_{X' \in \mathcal{B}(X,\epsilon)} \mathbf{1}\{f_\theta(X')f_{oracle}(X') \leq 0\} \\
&\leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*) + \mathbf{E}_{(X,Y)\sim D} \max_{X' \in \mathcal{B}(X,\epsilon)} \phi(f_\theta(X')f_{oracle}(X')) \\
&\textit{Let } f_{oracle}(X') = Y, \textit{ then,} \\
&\mathcal{R}_{cali}(f) - \mathcal{R}^* \leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*) + \mathbf{E}_{(X,Y)\sim D} \max_{\substack{X' \in \mathbf{B}(X,\epsilon) \\ f_{oracle}(X')=Y}} \phi(f_\theta(X')Y)
\end{aligned}$$

□

Algorithm 1 Calibrated adversarial training

```
Input: neural network $f_\theta$, neural network $g_\varphi$, training dataset $(X, Y) \in D$.
Output: adversarial robust network $f_\theta$
for $epoch = 1$ to $T$ do
  for mini-batch=1 to $M$ do
    Generate $X' = X + \delta$ using PGD or C&W$_\infty$ attack
    Obtain $X'_{cali}$ using Eq. 10
    Update $\theta$ by back-propagating Eq. 13
    Update $\varphi$ by back-propagating Eq. 14
  end for
end for
```

The first inequality holds when ϕ is a classification-calibrated loss Zhang et al. (2019); Bartlett et al. (2006). Classification-calibrated loss contains the cross-entropy loss, hinge loss, **KL** divergence and etc.

Appendix B. Training Strategy

There are two neural networks to be trained: f_θ and g_φ. f_θ is the neural network that we want to obtain and g_φ is the auxiliary neural network for generating soft mask M. In practice, we train these two neural networks in turn. Specifically, in each step, we firstly update f_θ using Eq. 13, then update g_φ using Eq. 14. More details can be found in Algorithm 1.

The pseudocode of our method is showed in Algorithm 1.

Appendix C. Implement Details

C.1. MNIST

We copy the model architecture for f_θ from https://adversarial-ml-tutorial.org/adversarial_training/.

Model architecture for f_θ and g_φ: The architecture of f_θ and g_φ are showed in Table 5.

Table 5: Model architecture (MINST)

f_θ		g_φ	
Layer name	Neurons	Layer name	Neurons
Conv layer	32	Conv layer	64
Conv layer	32	Conv layer	128
Conv layer	64	Conv layer	128
Conv layer	64	Up sampling	(28*28)
FC layer	(7*7*64)X100	Conv layer	1
FC layer	100X10	Sigmoid	-

Hyper-parameters settings for training our method: Epochs:40, optimizer:Adam, The initial learning rate is 1e-3 divided by 10 at 30-th epoch. we set $k = 150$ for CW_∞ loss in CAT$_{cw}$. Other hyper-parameters are described in Section 5.1.1.

C.2. CIFAR-10/CIFAR-100

Model architecture for g_φ: The architecture of g_φ is showed in Table 6.

Table 6: Model architecture g_φ (CIFAR-10/CIFAR-100)

Layer name	Neurons
ResNet-18 without FC layer	-
Up Sampling	(32*32)
Conv layer	3
Sigmoid	-

Hyper-parameters for training our method: For CIFAR-10, we use SGD optimizer with momentum 0.9, weight decay 5e-4 and an initial learning rate of 0.1, which divided by 10 at 100-th and 120-th epoch. Total epochs:140.

For CIFAR-100,we use SGD optimizer with momentum 0.9, weight decay 5e-4 and an initial learning rate of 0.1, which divided by 10 at 100-th and 110-th epoch. Total epochs:120.

we set $k = 50$ for CW_∞ loss in CAT$_{cw}$. Other hyper-parameters are described in Section 5.1.1.

Baselines We run the official code for baselines and all hyper-parameters are set to the values reported in their papers.

- **AT**, which is described in Section 3.2. We adopt the implementation in Rice et al. (2020) with early stop, which can achieve better performance. we use the implementation in Rice et al. (2020). https://github.com/locuslab/robust_overfitting.
- **Trades** Zhang et al. (2019).It separates loss function into cross-entropy loss for natural accuracy and a regularization terms for robust accuracy. The official code: https://github.com/yaodongyu/TRADES.
- **MART** Wang et al. (2020). It incorporates an explicit regularization into the loss function for misclassified examples. The official code: https://github.com/YisenWang/MART.
- **FAT** Zhang et al. (2020). It constructs the objective function based on adversarial examples that close to the model's decision boundary. We adopt *FAT for TRADES* as a baseline since it achieves better performance. The official code: https://github.com/zjfheart/Friendly-Adversarial-Training.

All trained models in our experiments are trained in a single Nvidia Tesla V100 GPU and selected at the best checkpoint where the sum of robust accuracy (under PGD-10) and natural accuracy is highest.

Appendix D. Experiments on CIFAR-100

This section shows the performance of our method on CIFAR-100. The test settings are the same as Table 2 and Table 3. Results are reported in Table 7. From Table 7, we can see that our method improves natural accuracy and robust accuracy compared with AT, which further shows the evidence that our method are effective in achieving natural and robust accuracy.

Table 7: Evaluation on CIFAR-100 (PreAct ResNet-18).

Models	Natural	FGSM	PGD-20	PGD-100	C&W$_\infty$	Avg
AT	55.13	29.77	27.51	27.01	25.91	33.06
CAT$_{cent}$($\beta_1 = 0.05$)	58.52	31.45	28.93	28.48	25.55	34.59
CAT$_{cent}$($\beta_1 = 0.1$)	59.77	30.33	27.16	26.62	24.17	33.61
CAT$_{cw}$($\beta_1 = 0.05$)	58.9	**31.51**	**29.11**	**28.5**	**26.25**	**34.85**
CAT$_{cw}$($\beta_1 = 0.1$)	**60.37**	31.04	28.31	27.88	25.67	34.65

Appendix E. Analysis for Hyper-parameter β

In this section, we conduct experiments for hyper-parameter β with varying from 1 to 5. Test settings are the same as Figure 6. Results are showed in Table 8. Besides, we also report results for hyper-parameter β_1 with varying from 0.05 to 0.3.

From Table 8, it can be observed that β also controls a trade-off between natural accuracy and robust accuracy. The larger β value leads to a larger robust accuracy but with a smaller natural accuracy. By comparing with β_1, we can see that adapting β_1 can achieve a better trade-off than adapting β. Therefore, for our method, we suggest to fix β to large value, e.g., $\beta = 5$, then adapt β_1 to achieve the trade-off that we want.

Table 8: Impact of hyper-parameter β.

CAT$_{cent}$						CAT$_{cw}$					
β^a	*Nat*	PGD-20	$\beta_1{}^b$	*Nat*	PGD-20	β^a	*Nat*	PGD-20	$\beta_1{}^b$	*Nat*	PGD-20
5	84.1	55.6	0.05	84.1	55.6	5	84.2	55.2	0.05	84.2	55.2
4	85.0	54.6	0.1	85.8	54.1	4	84.5	55.1	0.1	85.3	54.9
3	85.4	53.7	0.2	87.0	52.6	3	85.5	53.5	0.2	86.9	52.8
2	86.4	51.7	0.3	88.0	51.1	2	86.3	52.6	0.3	88.1	51.5
1	86.8	51.1	-	-	-	1	87.6	50.8	-	-	-

a : Model is trained with fixing β_1 : 0.05. b : Model is trained with fixing β : 5. *Nat* denotes Natural accuracy.

Appendix F. Difference Between Pixel-level Adapted and Instance-level Adapted Adversarial Examples

It is assumed that we want to find adversarial examples on the decision boundary. As showed in Figure 8, given the maximum perturbation bound is ε. Instance-level adapted adversarial examples will reduce ε to ε' and find the adapted adversarial examples x'_0 while pixel-level adapted adversarial examples may find x'_1 or x'_2 as long as it is on the decision boundary within the ϵ-ball. In other words, pixel-level adapted adversarial examples could lead to more diversified adversarial examples.

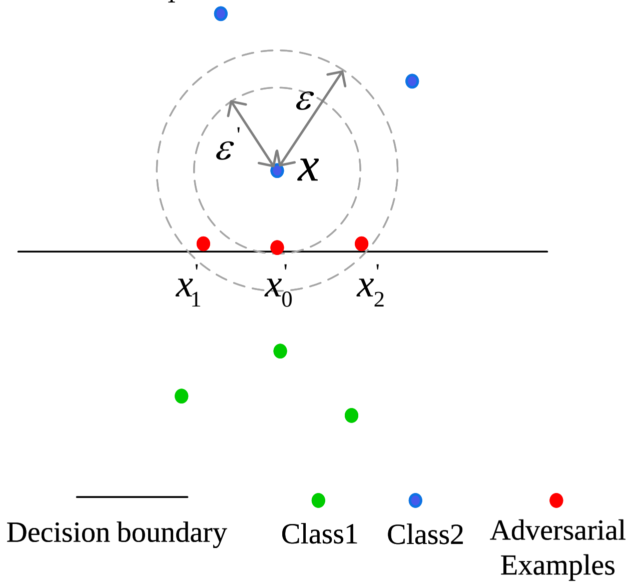


Figure 8: Illustration of the difference between pixel-level adapted and instance-level adapted adversarial examples.

Appendix G. Analysis for Differences Between the Proposed General Objective Function (Eq. 7) and the General Objective Function in Zhang et al. (2019)

The general objective function derived on the upper bound of robust error is expressed as follows Zhang et al. (2019):

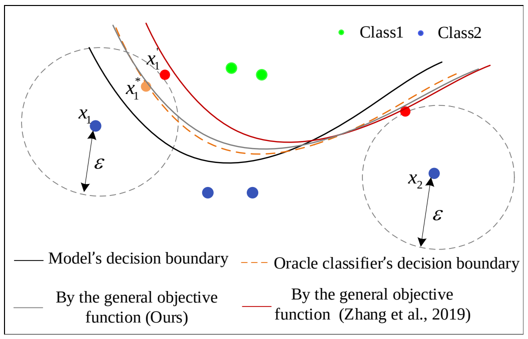
$$\min_{\theta} \mathbf{E}_{(X,Y)}[\phi(f_\theta(X)Y) + \max_{X' \in \mathbf{B}(X,\epsilon)} \phi(f_\theta(X')Y)/\lambda]. \tag{17}$$

By comparing Eq. 17 and Eq. 7, the main difference is that there is an extra constraint $f_{oracle}(X') = Y$ for the inner maximization in our proposed general objective function. Therefore, we heuristically analyze the decision boundaries learned by these two general objective functions according to whether the constraint $f_{oracle}(X') = Y$ is active or not:

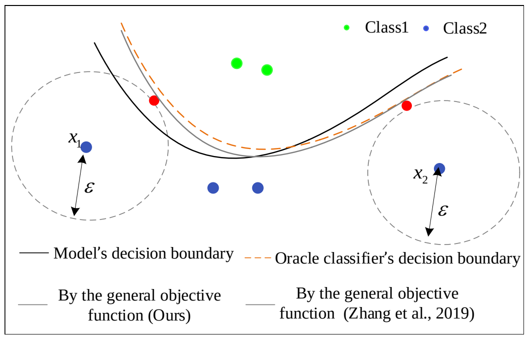
- Given the oracle classifier's decision boundary does not cross the ϵ-ball of input X, then the $f_{oracle}(X') = Y$ is an inactive constraint for the inner maximization and the proposed general objective function will be equivalent to Eq. 17. As showed in Figure 9(*b*)subfigure, by minimizing the general objective function, the decision boundary would be transformed from black solid line to gray solid line in order to classify adversarial examples (red points) as "Class 2".

- Given the oracle decision boundary crosses the ϵ-ball of input x, then the $f_{oracle}(X') = Y$ is an active constraint. As showed in Figure 9(a)subfigure, the constraint $f_{oracle}(X') = Y$ is active for input x_1. Therefore, the example generated by the inner maximization in the proposed general objective function will be x_1^* (orange point) while the example generated by the inner maximization in Eq. 17 will be x_1' (red point). By minimizing the general objective function, the decision boundary learned by the proposed general objective function could be the gray line since it try to classify X_1^* as "Class 2" while the decision boundary learned by Eq. 17 could be the red line since it try to classify x_1' as "Class 2".

Intuitively, our proposed general objective function will push the learned decision boundary to be near the oracle classifier's decision boundary while the general objective function in Zhang et al. (2019) will push the learned decision boundary to be near the boundary of ϵ-ball.


Class1
Class2
x_1'
x_1^*
x_1
ε
x_2
ε
Model's decision boundary
Oracle classifier's decision boundary
By the general objective function (Ours)
By the general objective function (Zhang et al., 2019)

(a) Active


Class1
Class2
x_1
ε
x_2
ε
Model's decision boundary
Oracle classifier's decision boundary
By the general objective function (Ours)
By the general objective function (Zhang et al., 2019)

(b) Inactive

Figure 9: Illustration for the decision boundaries learned by the proposed general objective function and the general objective function in Zhang et al. (2019). The gray line in Figure 9(a)subfigure denotes the decision boundary learned by our proposed general objective function. The red line in Figure 9(a)subfigure denotes the decision boundary learned by the general objective function in Zhang et al. (2019). The gray line in Figure 9(b)subfigure denotes the decision boundary learned by our proposed general objective function or by the general objective function in Zhang et al. (2019).

Appendix H. The Loss Functions for Other Variants of Adversarial Training

Table 9: Loss functions of other variants of adversarial training.

Methods	Loss Function
AT Madry et al. (2017)	$L(f_\theta(X'), Y)$
ALP Kannan et al. (2018)	$L(f_\theta(X'), Y) + \lambda \cdot \lVert f_\theta(X') - f_\theta(X) \rVert$
MMA Ding et al. (2019)	$L(f_\theta(X'), Y) \cdot \mathbf{1}(f_\theta(X) = Y) + L(f_\theta(X), Y) \cdot \mathbf{1}(f_\theta(X) \neq Y)$
Trades Zhang et al. (2019)	$L(f_\theta(X), Y) + \lambda \cdot \mathbf{KL}(P(Y\|X') \|\| P(Y\|X))$
MART Wang et al. (2020)	$BCE(f_\theta(X'), Y) + \lambda \cdot \mathbf{KL}(P(Y\|X') \|\| P(Y\|X)) \cdot (1 - P(Y = y\|X))$