LAS-AT: Adversarial Training with Learnable Attack Strategy Supplementary Material

Xiaojun Jia[1,2,†*], Yong Zhang[3,*], Baoyuan Wu[4,5,‡], Ke Ma[6], Jue Wang[3], Xiaochun Cao[1,2,‡]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyberspace Security, University of Chinese Academy of Sciences, Beijing, China
[3] Tencent, AI Lab, Shenzhen, China
[4]School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
[5] Secure Computing Lab of Big Data, Shenzhen Research Institute of Big Data, Shenzhen, China
[6] School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China
jiaxiaojun@iie.ac.cn; zhangyong201303@gmail.com; wubaoyuan@cuhk.edu.cn;
make@ucas.ac.cn; arphid@gmail.com; caoxiaochun@iie.ac.cn

In the manuscript, we report extensive experimental results on three benchmark databases. In this supplementary material, more additional studies are provided as follows:

- We present the architecture of the strategy network (see Sec. 1).
- We present the proof of theorem 1 (see Sec. 2).
- We introduce the details of these databases (see Sec. 3).
- We introduce the detailed settings for the proposed LAT-AT (see Sec. 4).
- We introduce the training and evaluation settings for the comparison methods (see Sec. 5).
- We present comparisons with AWP trained with more iterations and a larger perturbation strength (see Sec. 6).
- We study how the maximal perturbation strength, the number of iteration, and the step size affect the robustness and the clean accuracy of the target network in adversarial training. We use PGD-AT [6] as an illustration (see Sec. 7).
- We present the selection of the two trade-off hyper-parameters (*i.e.,* α and β) in the objective function (see Sec. 8).
- We illustrate the evolution of the generated perturbation strength of several images during the training process (see Sec. 9).
- We conduct experiments on more image databases (see Sec. 10).
- We introduce the selection of the strategy $\hat{a}$ (see Sec. 11).
- We discuss the training efficiency of the proposed method (see Sec. 12).

*The first two authors contribute equally to this work. † Work done during an internship at Tencent AI Lab. ‡ Correspondence to: Baoyuan Wu (wubaoyuan@cuhk.edu.cn) and Xiaochun Cao (caoxiaochun@iie.ac.cn).

1. Architecture of the Strategy Network

The architecture of the strategy network is illustrated in Fig. 1. We exploit ResNet18 [3] as the backbone. Given an image, the strategy network outputs an attack strategy, *i.e.,* the configuration of how to perform the adversarial attack. Specifically, the strategy network outputs a set of attack parameters for AE generation.

2. Proof of Theorem 1

First we introduce some notations. Let $\mathcal{L}_0 : \mathcal{X} \times \mathcal{Y} \times \mathcal{W} \times \mathbf{\Theta} \to \mathbb{R}^+$ be the objective function in (7) of submission (Line 400) as

$$\mathcal{L}_0 = \mathcal{L}_1 + \alpha\mathcal{L}_2 + \beta\mathcal{L}_3. \tag{1}$$

We define $\boldsymbol{x}^*_{\text{adv}}(\boldsymbol{x}, \mathbf{w})$ as the optimal adversarial example generated by the strategy network

$$\begin{aligned} \boldsymbol{x}^*_{\text{adv}}(\boldsymbol{x}, \mathbf{w}) &= \arg\max_{\boldsymbol{\theta}} g(\boldsymbol{x}, \boldsymbol{a}(\boldsymbol{\theta}), \mathbf{w}) \\ &= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{a}\sim p(\boldsymbol{a}|\boldsymbol{x},\boldsymbol{\theta})}[\mathcal{L}_0], \end{aligned} \tag{2}$$

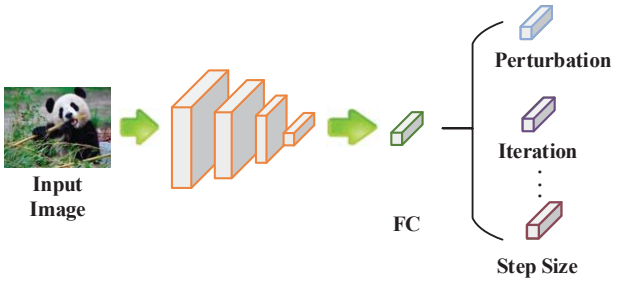

Figure 1. The overview of the strategy network. The outputs are a set of attack parameters for generating adversarial examples. Each attack parameter is encoded by a one-hot vector.

and $\hat{\boldsymbol{x}}_{\text{adv}}(\boldsymbol{x}, \mathbf{w})$ is a δ-approximate solution to $\boldsymbol{x}^*_{\text{adv}}(\boldsymbol{x}, \mathbf{w})$. In addition, the full gradient of $\mathcal{L}_0$ w.r.t $\mathbf{w}$ is

$$\begin{aligned} \nabla_{\mathbf{w}}\mathcal{L}_0(\mathbf{w}) &= \frac{1}{N}\sum_{i=n}^{N}\nabla_{\mathbf{w}}\mathcal{L}_0^n \\ &= \frac{1}{N}\sum_{n=1}^{N}\nabla_{\mathbf{w}}\mathcal{L}_0(\boldsymbol{x}^*_{\text{adv}}(\boldsymbol{x}_n, \mathbf{w}), \mathbf{w}), \end{aligned} \tag{3}$$

where $\boldsymbol{x}^*_{\text{adv}}(\boldsymbol{x}_n)$ is the optimal adversarial example for $\boldsymbol{x}_n$. The stochastic gradient of $\mathcal{L}_0$ w.r.t $\mathbf{w}$ is

$$\begin{aligned} \nabla_{\mathbf{w}}\ell(\mathbf{w}) &= \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\nabla_{\mathbf{w}}\mathcal{L}_0^i \\ &= \frac{1}{|\mathcal{B}|}\sum_{n=1}^{N}\nabla_{\mathbf{w}}\mathcal{L}_0(\boldsymbol{x}^*_{\text{adv}}(\boldsymbol{x}_i, \mathbf{w}), \mathbf{w}). \end{aligned} \tag{4}$$

Then $\nabla_{\boldsymbol{\theta}}\mathcal{L}_0$ and $\nabla_{\boldsymbol{\theta}}\ell$ correspond to the full and stochastic gradients of $\mathcal{L}_0$ w.r.t $\boldsymbol{\theta}$. Without lose of generality, we assume that

$$\mathbb{E}[\nabla_{\mathbf{w}}\ell(\mathbf{w})] = \nabla_{\mathbf{w}}\mathcal{L}_0(\mathbf{w}). \tag{5}$$

We note the approximate stochastic gradient as $\nabla_{\mathbf{w}}\hat{\ell}$:

$$\begin{aligned} \nabla_{\mathbf{w}}\hat{\ell}\,(\mathbf{w}) &= \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\nabla_{\mathbf{w}}\hat{\mathcal{L}}_0^i \\ &= \frac{1}{|\mathcal{B}|}\sum_{n=1}^{N}\nabla_{\mathbf{w}}\mathcal{L}_0(\hat{\boldsymbol{x}}_{\text{adv}}(\boldsymbol{x}_i, \mathbf{w}), \mathbf{w}). \end{aligned} \tag{6}$$

Moreover, the adversarial example $\boldsymbol{x}_{\text{adv}}(\boldsymbol{x}, \mathbf{w})$ can be identified by a parameter $\boldsymbol{\theta}$ of the strategy network and the gradients like (3), (4), (6) would be

$$\begin{aligned} \nabla_{\mathbf{w}}\mathcal{L}_0(\boldsymbol{\theta}, \mathbf{w}) &:= \nabla_{\mathbf{w}}\mathcal{L}_0(\mathbf{w}) \\ \nabla_{\mathbf{w}}\ell(\boldsymbol{\theta}, \mathbf{w}) &:= \nabla_{\mathbf{w}}\ell(\mathbf{w}) \\ \nabla_{\mathbf{w}}\hat{\ell}(\boldsymbol{\theta}, \mathbf{w}) &:= \nabla_{\mathbf{w}}\hat{\ell}\,(\mathbf{w}). \end{aligned} \tag{7}$$

The corresponding gradients w.r.t $\boldsymbol{\theta}$ will be $\nabla_{\boldsymbol{\theta}}\mathcal{L}_0$, $\nabla_{\boldsymbol{\theta}}\ell$ and $\nabla_{\boldsymbol{\theta}}\hat{\ell}$. As the $\mathcal{L}_0$ in (7) of submission (Line 400) satisfies the Lipschitz gradient conditions, given $\boldsymbol{x}_n \in \mathcal{X}$, it holds that

$$\begin{aligned} &\sup_{\boldsymbol{\theta}}\|\nabla_{\mathbf{w}}\mathcal{L}_0^n(\boldsymbol{\theta}, \mathbf{w}) - \nabla_{\mathbf{w}}\mathcal{L}_0^n(\boldsymbol{\theta}, \mathbf{w}')\|_2 \\ \le\ & L_{\mathbf{ww}}\|\mathbf{w} - \mathbf{w}'\|_2 \\ &\sup_{\mathbf{w}}\|\nabla_{\mathbf{w}}\mathcal{L}_0^n(\boldsymbol{\theta}, \mathbf{w}) - \nabla_{\mathbf{w}}\mathcal{L}_0^n(\boldsymbol{\theta}', \mathbf{w})\|_2 \\ \le\ & L_{\mathbf{w}\boldsymbol{\theta}}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2 \\ &\sup_{\boldsymbol{\theta}}\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_0^n(\boldsymbol{\theta}, \mathbf{w}) - \nabla_{\boldsymbol{\theta}}\mathcal{L}_0^n(\boldsymbol{\theta}, \mathbf{w}')\|_2 \\ \le\ & L_{\boldsymbol{\theta}\mathbf{w}}\|\mathbf{w} - \mathbf{w}'\|_2, \end{aligned} \tag{8}$$

where $L_{\mathbf{ww}}$, $L_{\mathbf{w}\boldsymbol{\theta}}$ and $L_{\boldsymbol{\theta}\mathbf{w}}$ are positive constants. Furthermore, by the strongly-concavity of $\mathcal{L}_0$ and given $\boldsymbol{x}_n \in \mathcal{X}$, we know that for any $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2 \in \boldsymbol{\Theta}$,

$$\begin{aligned} &\mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}) - \mathcal{L}_0^n(\boldsymbol{\theta}_2, \mathbf{w}) \\ \le\ & \langle\nabla_{\boldsymbol{\theta}}\mathcal{L}_0^n(\boldsymbol{\theta}, \mathbf{w}), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle - \frac{\mu}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2. \end{aligned} \tag{9}$$

As the variance of the stochastic gradient is bounded by $\sigma^2 > 0$, it means that

$$\mathbb{E}\big[\|\nabla_{\mathbf{w}}\ell(\mathbf{w}) - \nabla_{\mathbf{w}}\mathcal{L}_0(\mathbf{w})\|_2^2\big] \le \sigma^2. \tag{10}$$

To prove the main result, we need the following two important lemmas.

Lemma 1. *Suppose that $\mathcal{L}_0$ in (7) of submission (Line 400) satisfies the Lipschitz gradient conditions as (8) and $\mathcal{L}_0$ is μ-strongly concave in $\boldsymbol{\Theta}$, we have $\mathcal{L}_0$ is Lipschitz smooth with L_0*

$$L_0 = \frac{L_{\mathbf{w}\boldsymbol{\theta}}L_{\boldsymbol{\theta}\mathbf{w}}}{\mu} + L_{\mathbf{ww}}. \tag{11}$$

It holds that

$$\begin{aligned} \mathcal{L}_0(\mathbf{w}_1) \le \mathcal{L}_0(\mathbf{w}_2) &+ \langle\nabla_{\mathbf{w}}\mathcal{L}_0(\mathbf{w}_2), \mathbf{w}_1 - \mathbf{w}_2\rangle \\ &+ \frac{L_0}{2}\|\mathbf{w}_1 - \mathbf{w}_2\|_2^2, \end{aligned} \tag{12}$$

and

$$\|\nabla_{\mathbf{w}}\mathcal{L}_0 - \nabla_{\mathbf{w}}\mathcal{L}_0(\mathbf{w}_2)\|_2 \le L_0\|\mathbf{w}_1 - \mathbf{w}_2\|_2. \tag{13}$$

Proof. By the strongly-concavity of $\mathcal{L}_0$ and given $\boldsymbol{x}_n \in \mathcal{X}$, for any $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$ and the corresponding $\mathbf{w}_1$, $\mathbf{w}_2$, we have

$$\begin{aligned} &\mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_2) - \mathcal{L}_0^n(\boldsymbol{\theta}_2, \mathbf{w}_2) \\ \le\ & \langle\nabla_{\boldsymbol{\theta}}\mathcal{L}_0^n(\boldsymbol{\theta}_2, \mathbf{w}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle - \frac{\mu}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2 \\ \le\ & -\frac{\mu}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2. \end{aligned} \tag{14}$$

The second inequality is true as

$$\langle\nabla_{\boldsymbol{\theta}}\mathcal{L}_0^n(\boldsymbol{\theta}_2, \mathbf{w}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle \le 0.$$

In addition, we have

$$\begin{aligned}
&\mathcal{L}_0^n(\boldsymbol{\theta}_2, \mathbf{w}_2) - \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_2) \\
\leq\ & \left\langle \nabla_{\boldsymbol{\theta}} \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_2), \boldsymbol{\theta}_2 - \boldsymbol{\theta}_1 \right\rangle - \frac{\mu}{2} \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2 \\
\leq\ & -\frac{\mu}{2} \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2.
\end{aligned} \tag{15}$$

Combining (14) and (15), we have

$$\begin{aligned}
&\mu \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2 \\
\leq\ & \langle \nabla_{\boldsymbol{\theta}} \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_2), \boldsymbol{\theta}_2 - \boldsymbol{\theta}_1 \rangle \\
\leq\ & \langle \nabla_{\boldsymbol{\theta}} \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_2) - \nabla_{\boldsymbol{\theta}} \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_1), \boldsymbol{\theta}_2 - \boldsymbol{\theta}_1 \rangle \\
\leq\ & \|\nabla_{\boldsymbol{\theta}} \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_2) - \nabla_{\boldsymbol{\theta}} \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_1)\|_2 \|\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1\|_2 \\
\leq\ & L_{\boldsymbol{\theta}\mathbf{w}} \|\mathbf{w}_2 - \mathbf{w}_1\|_2 \|\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1\|_2,
\end{aligned} \tag{16}$$

where the second inequality holds as

$$\langle \nabla_{\boldsymbol{\theta}} \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_1), \boldsymbol{\theta}_2 - \boldsymbol{\theta}_1 \rangle \leq 0,$$

the third inequality follows from the Cauchy-Schwarz inequality, and the last one holds by the Lipschitz smoothness of the gradients of $\mathcal{L}_0$ (8).

For any $n \in [N]$, we have

$$\begin{aligned}
&\|\nabla_{\mathbf{w}} \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_1) - \nabla_{\mathbf{w}} \mathcal{L}_0^n(\boldsymbol{\theta}_2, \mathbf{w}_2)\|_2 \\
\leq\ & \|\nabla_{\mathbf{w}} \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_1) - \nabla_{\mathbf{w}} \mathcal{L}_0^n(\boldsymbol{\theta}_2, \mathbf{w}_1)\|_2 \\
&+ \|\nabla_{\mathbf{w}} \mathcal{L}_0^n(\boldsymbol{\theta}_2, \mathbf{w}_1) - \nabla_{\mathbf{w}} \mathcal{L}_0^n(\boldsymbol{\theta}_2, \mathbf{w}_2)\|_2 \\
\leq\ & L_{\mathbf{w}\boldsymbol{\theta}} \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2 + L_{\mathbf{ww}} \|\mathbf{w}_1 - \mathbf{w}_2\|_2 \\
=\ & \left(\frac{L_{\mathbf{w}\boldsymbol{\theta}} L_{\boldsymbol{\theta}\mathbf{w}}}{\mu} + L_{\mathbf{ww}} \right) \|\mathbf{w}_1 - \mathbf{w}_2\|_2,
\end{aligned} \tag{17}$$

where the first inequality follows from the triangle inequality, and the second inequality holds due to (16) and the Lipschitz smoothness of the gradients of $\mathcal{L}_0$ (8). By the definition of $\mathcal{L}$, it holds that

$$\begin{aligned}
&\|\nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}_1) - \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}_2)\|_2 \\
=\ & \left\| \frac{1}{N} \sum_{n=1}^{N} (\nabla_{\mathbf{w}} \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_1) - \nabla_{\mathbf{w}} \mathcal{L}_0^n(\boldsymbol{\theta}_2, \mathbf{w}_2)) \right\|_2 \\
\leq\ & \frac{1}{N} \sum_{n=1}^{N} \|\nabla_{\mathbf{w}} \mathcal{L}_0^n(\boldsymbol{\theta}_1, \mathbf{w}_1) - \nabla_{\mathbf{w}} \mathcal{L}_0^n(\boldsymbol{\theta}_2, \mathbf{w}_2)\|_2 \\
\leq\ & \left(\frac{L_{\mathbf{w}\boldsymbol{\theta}} L_{\boldsymbol{\theta}\mathbf{w}}}{\mu} + L_{\mathbf{ww}} \right) \|\mathbf{w}_1 - \mathbf{w}_2\|_2.
\end{aligned} \tag{18}$$

With the definition of the Lipschitz smoothness, we complete the proof. □

Lemma 2. *Suppose that $\mathcal{L}_0$ in (7) of submission (Line 400) satisfies the Lipschitz gradient conditions as (8) and $\mathcal{L}_0$ is μ-strongly concave in $\boldsymbol{\Theta}$, the approximate stochastic gradient $\nabla_{\mathbf{w}} \hat{\ell}(\mathbf{w})$ (6) satisfies*

$$\|\nabla_{\mathbf{w}} \hat{\ell}\,(\mathbf{w}) - \nabla_{\mathbf{w}} \ell(\mathbf{w})\|_2 \leq L_{\mathbf{w}\boldsymbol{\theta}} \sqrt{\frac{\delta}{\mu}}, \tag{19}$$

where $\hat{\boldsymbol{x}}_{adv}(\boldsymbol{x}, \mathbf{w})$ is a δ-approximate solution to $\boldsymbol{x}_{adv}^*(\boldsymbol{x}, \mathbf{w})$ with given $\boldsymbol{x} \in \mathcal{X}$.

Proof. By the definitions of $\nabla_{\mathbf{w}} \hat{\ell}$ and $\nabla_{\mathbf{w}} \ell$, we have

$$\begin{aligned}
&\|\nabla_{\mathbf{w}} \hat{\ell}\,(\mathbf{w}) - \nabla_{\mathbf{w}} \ell(\mathbf{w})\|_2 \\
=\ & \Bigg\| \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} (\nabla_{\mathbf{w}} \mathcal{L}_0(\hat{\boldsymbol{x}}_{\text{adv}}(\boldsymbol{x}_i, \mathbf{w}), \mathbf{w}) \\
&\qquad - \nabla_{\mathbf{w}} \mathcal{L}_0(\boldsymbol{x}_{\text{adv}}^*(\boldsymbol{x}_i, \mathbf{w}), \mathbf{w})) \Bigg\|_2 \\
\leq\ & \frac{1}{|\mathcal{B}|} \sum_{n=1}^{N} \|\nabla_{\mathbf{w}} \mathcal{L}_0(\hat{\boldsymbol{x}}_{\text{adv}}(\boldsymbol{x}_i, \mathbf{w}), \mathbf{w}) \\
&\qquad - \nabla_{\mathbf{w}} \mathcal{L}_0(\boldsymbol{x}_{\text{adv}}^*(\boldsymbol{x}_i, \mathbf{w}), \mathbf{w})\|_2 \\
\leq\ & \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} L_{\mathbf{w}\boldsymbol{\theta}} \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2,
\end{aligned} \tag{20}$$

where the second inequality follows from the triangle inequality, the third inequality holds due to the gradient Lipschitz condition, and $\hat{\boldsymbol{\theta}}$ is the parameter of strategy network corresponding to $\hat{\boldsymbol{x}}_{\text{adv}}(\boldsymbol{x}_i, \mathbf{w})$, $\boldsymbol{\theta}^*$ is similar.

Since $\hat{\boldsymbol{x}}_{\text{adv}}(\boldsymbol{x}_i, \mathbf{w})$ is a δ-approximate adversarial example generated by the strategy network, we have

$$\left\langle \boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}, \nabla_{\boldsymbol{\theta}} \mathcal{L}_0(\hat{\boldsymbol{\theta}}, \mathbf{w}) \right\rangle \leq \delta. \tag{21}$$

In addition, it holds that

$$\left\langle \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*, \nabla_{\boldsymbol{\theta}} \mathcal{L}_0(\boldsymbol{\theta}^*, \mathbf{w}) \right\rangle \leq 0. \tag{22}$$

Putting (21) and (22) together gives birth to

$$\left\langle \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*, \nabla_{\boldsymbol{\theta}} \mathcal{L}_0(\boldsymbol{\theta}^*, \mathbf{w}) - \nabla_{\boldsymbol{\theta}} \mathcal{L}_0(\hat{\boldsymbol{\theta}}, \mathbf{w}) \right\rangle \leq \delta. \tag{23}$$

Moreover, by the strongly concavity of $\mathcal{L}_0$ and (16), we have

$$\begin{aligned}
&\mu \|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}\|_2^2 \\
\leq\ & \langle \nabla_{\boldsymbol{\theta}} \mathcal{L}_0^n(\boldsymbol{\theta}^*, \mathbf{w}) - \nabla_{\boldsymbol{\theta}} \mathcal{L}_0^n(\hat{\boldsymbol{\theta}}, \mathbf{w}), \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^* \rangle \\
\leq\ & \delta.
\end{aligned} \tag{24}$$

Consequently, it immediately yields

$$\|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}\|_2 \leq \sqrt{\frac{\delta}{\mu}}. \tag{25}$$

Substituting (25) into (20), we complete the proof. □

Proof. By Lemma 1, we have

$$\begin{aligned}\mathcal{L}_0(\mathbf{w}^{t+1}) \;\leq\; & \mathcal{L}_0(\mathbf{w}^t) + \frac{L_0}{2}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|_2^2 \\ & + \left\langle \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t), \mathbf{w}^{t+1} - \mathbf{w}^t \right\rangle.\end{aligned}$$

Due to

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta_t \nabla_{\mathbf{w}} \hat{\ell}(\mathbf{w}^t),$$

it holds that

$$\begin{aligned}
& \mathcal{L}_0(\mathbf{w}^{t+1}) \\
\leq\; & \mathcal{L}_0(\mathbf{w}^t) - \eta_t \|\nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t)\|_2^2 + \frac{L_0 \eta_t^2}{2} \|\nabla_{\mathbf{w}} \hat{\ell}(\mathbf{w}^t)\|_2^2 \\
& + \eta_t \langle \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t), \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t) - \nabla_{\mathbf{w}} \hat{\ell}(\mathbf{w}^t) \rangle \\
=\; & \mathcal{L}_0(\mathbf{w}^t) - \eta_t \left(1 - \frac{L_0 \eta_t}{2}\right) \|\nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t)\|_2^2 \\
& + \eta_t \left(1 - \frac{L_0 \eta_t}{2}\right) \langle \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t), \\
& \qquad \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t) - \nabla_{\mathbf{w}} \hat{\ell}(\mathbf{w}^t) \rangle \\
& + \frac{L_0 \eta_t^2}{2} \|\nabla_{\mathbf{w}} \hat{\ell}(\mathbf{w}^t) - \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t)\|_2^2 \\
=\; & \mathcal{L}_0(\mathbf{w}^t) - \eta_t \left(1 - \frac{L_0 \eta_t}{2}\right) \|\nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t)\|_2^2 \\
& + \eta_t \left(1 - \frac{L_0 \eta_t}{2}\right) \langle \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t), \\
& \qquad \nabla_{\mathbf{w}} \ell(\mathbf{w}^t) - \nabla_{\mathbf{w}} \hat{\ell}(\mathbf{w}^t) \rangle \\
& + \eta_t \left(1 - \frac{L_0 \eta_t}{2}\right) \langle \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t), \\
& \qquad \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t) - \nabla_{\mathbf{w}} \ell(\mathbf{w}^t) \rangle \\
& + \frac{L_0 \eta_t^2}{2} \|\nabla_{\mathbf{w}} \hat{\ell}(\mathbf{w}^t) - \nabla_{\mathbf{w}} \ell(\mathbf{w}^t) \\
& \qquad + \nabla_{\mathbf{w}} \ell(\mathbf{w}^t) - \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t)\|_2^2 \\
\leq\; & \mathcal{L}_0(\mathbf{w}^t) - \frac{\eta_t}{2} \left(1 - \frac{L_0 \eta_t}{2}\right) \|\nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t)\|_2^2 \\
& + \frac{\eta_t}{2} \left(1 - \frac{L_0 \eta_t}{2}\right) \|\nabla_{\mathbf{w}} \ell(\mathbf{w}^t) - \nabla_{\mathbf{w}} \hat{\ell}(\mathbf{w}^t)\|_2^2 \\
& + \eta_t \left(1 + \frac{L_0 \eta_t}{2}\right) \langle \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t), \\
& \qquad \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t) - \nabla_{\mathbf{w}} \ell(\mathbf{w}^t) \rangle \\
& + L_0 \eta_t^2 \|\nabla_{\mathbf{w}} \hat{\ell}(\mathbf{w}^t) - \nabla_{\mathbf{w}} \ell(\mathbf{w}^t)\|_2^2 \\
& + L_0 \eta_t^2 \|\nabla_{\mathbf{w}} \ell(\mathbf{w}^t) - \nabla_{\mathbf{w}} \mathcal{L}_0(\mathbf{w}^t)\|_2^2
\end{aligned} \tag{26}$$

Taking expectation on both sides of the above inequality

conditioned on $\mathbf{w}^t$, then we have

$$\begin{aligned}
&\mathbb{E}[\mathcal{L}_0(\mathbf{w}^{t+1}) - \mathcal{L}_0(\mathbf{w}^t)|\mathbf{w}^t] \\
\leq\ & -\frac{\eta_t}{2}\left(1-\frac{L_0\eta_t}{2}\right)\|\nabla_{\mathbf{w}}\mathcal{L}_0(\mathbf{w}^t)\|_2^2 \\
&+\frac{\eta}{2}\left(1+\frac{3\eta_t L_0}{2}\right)\frac{\delta L_{\mathbf{w}\boldsymbol{\theta}}^2}{\mu}+L_0\eta_t^2\sigma^2.
\end{aligned} \tag{27}$$

Then we do the telescope sum over $t = 0, \ldots, T-1$, we obtain

$$\begin{aligned}
&\sum_{t=0}^{T-1}\frac{\eta_t}{2}\left(1-\frac{L_0\eta_t}{2}\right)\mathbb{E}[\|\mathcal{L}_0(\mathbf{w}^t)\|_2^2] \\
\leq\ & \mathbb{E}[\mathcal{L}_0(\mathbf{w}^0) - \mathcal{L}_0(\mathbf{w}^T)] + L_0\sum_{t=0}^{T-1}\eta_t^2\sigma^2 \\
&+\sum_{t=0}^{T-1}\frac{\eta}{2}\left(1+\frac{3\eta_t L_0}{2}\right)\frac{\delta L_{\mathbf{w}\boldsymbol{\theta}}^2}{\mu}.
\end{aligned} \tag{28}$$

Choosing $\eta_t = \eta_1$ as

$$\eta_1 = \min\left(\frac{1}{L_0}, \sqrt{\frac{\mathcal{L}_0(\mathbf{w}^0) - \min_{\mathbf{w}}\mathcal{L}_0(\mathbf{w})}{\sigma^2 T L_0}}\right), \tag{29}$$

it holds that

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big[\|\nabla\mathcal{L}_0(\mathbf{w}^t)\|_2^2\big] \leq 4\sigma\sqrt{\frac{\Delta L_0}{T}} + \frac{5\delta L_{\mathbf{w}\boldsymbol{\theta}}^2}{\mu}, \tag{30}$$

where $\Delta = \mathcal{L}_0(\mathbf{w}^0) - \min_{\mathbf{w}}\mathcal{L}_0(\mathbf{w})$. □

3. Details of the databases

CIFAR-10, CIFAR-100, and Tiny ImageNet are the most widely used databases for the evaluation of adversarial robustness. The CIFAR-10 dataset contains 50,000 training images and 10,000 test images, which covers 10 classes of images in the size of 32×32. The CIFAR-100 dataset also contains 50,000 training images and 10,000 test images in the size of 32×32, but it covers 100 classes. The Tiny ImageNet database is a subset collected from the ImageNet database, which covers 200 classes with 600 images in the size of 64×64 for each class. As there are no labels for test images of Tiny ImageNet, following [4], we evaluate methods on the validation set.

4. Detailed Settings for LAS-AT

LAS-PGD-AT: The proposed LAS-PGD-AT is implemented based on PGD-AT [6], which performs the early stopping adversarial training. PGD-AT is trained on the adversarial examples generated by a fixed PGD attack strategy. It has three stationary attack parameters that depict how to attack, *i.e.,* the maximal perturbation strength, the attack step, and the attack iteration. Hence, the strategy network of our LAS-PGD-AT has three parallel softmax layers to predict the three attack parameters. Following the default setting in [6], we adopt SGD with momentum 0.9, weight decay 5×10^{-4}, and batch size 128. LAS-PGD-AT and PGD-AT [6] are trained for 110 epochs. The learning rate decays with a factor of 0.1 at the 100 and 105 epochs, respectively.

LAS-TRADES: The proposed LAS-TRADES is implemented based on TRADES [10] which exploits a regularized surrogate loss to perform adversarial training. Since TRADES uses the same three attack parameters as PGD-AT to depict how to attack, the strategy network of LAS-TRADES shares the same output configuration as LAS-PGD-AT. Different from LAS-PGD-AT, LAS-TRADES is trained with the regularized surrogate loss rather than the cross-entropy loss. Following the default setting in [10], we adopt SGD with momentum 0.9, weight decay 2×10^{-4}, and batch size 128. LAS-TRADES and TRADES [10] are trained for 100 epochs. The learning rate decays with a factor of 0.1 at the 75 and 90 epochs, respectively.

LAS-AWP: The proposed LAS-AWP is implemented based on AWP [9]. AWP has 3 stationary attack parameters, *i.e.,* the maximal perturbation strength, the attack step, and the attack iteration. Hence, the strategy network of LAS-AWP has 3 parallel softmax layers to predict these attack parameters. Following the default setting in [9], we adopt SGD with momentum 0.9, weight decay 5×10^{-4}, and batch size 128. LAS-AWP and AWP [9] are trained for 200 epochs. The learning rate decays with a factor of 0.1 at the 100 and 150 epochs, respectively.

5. Detailed Training And Evaluation Settings

In Table 7 of the manuscript, we compared our LAS-Madry-AT with CAT [1], DART [7], and FAT [11]. We adopt the default setting in their original paper to train them. As for LAS-Madry-AT, it is trained for 200 epochs. The learning rate decays with a factor of 0.1 at the 100 and 150 epochs, respectively. We report the performance at the last epoch. In Fig. 3 of the manuscript, we compared our LAS-PGD-AT with OHL [5] and AdvHP [12]. OHL [5] and AdvHP [12] search the hyper-parameter of data augmentation for image classification. For a fair comparison, we use the same hyper-parameters and search range for them as ours. Specifically, the range of the maximal perturbation strength is set from 3 to 15, the range of the attack step is set from 1 to 6, and the range of the attack iteration is set from 3 to 15. We also adopt the same strategy network and training setting for them. Specifically, for the target network, we adopt

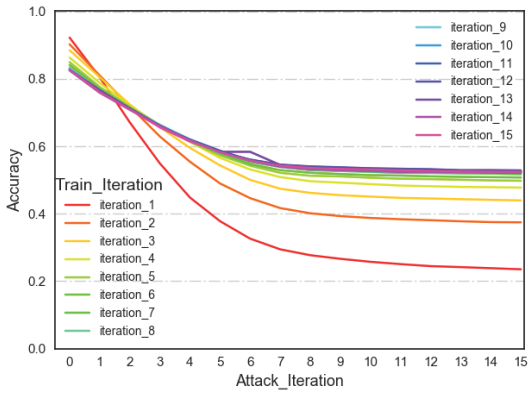

Figure 2. The influence of the number of iteration in PGD-AT [6]. Curves represent performance of PGD-AT [6] with different numbers of iteration for training. X-axis represents the number of iteration of the PGD attack. Y-axis represents the accuracy of adversarial examples generated by the PGD attack method.

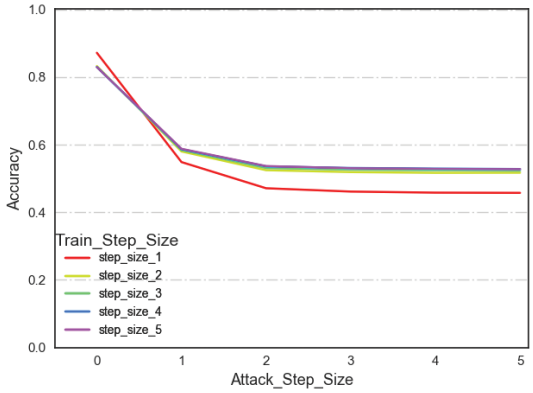

Figure 3. The influence of the step size in PGD-AT [6]. Curves represent performance of PGD-AT [6] with different step sizes for training. X-axis represents the step size of the PGD attack. Y-axis represents the accuracy of adversarial examples generated by the PGD attack method.

SGD momentum optimizer with a learning rate of 0.1, and a weight decay of 5×10^{-4}. ResNet18 is used as the backbone of the strategy network. The training epoch is set to 110. The learning rate decays with a factor of 0.1 at the 105 and 110 epochs, respectively. In this way, the difference between our proposed method and them is the loss term which guides the learning of the strategy network.

6. More Comparisons with AWP

Comparisons with AWP trained with more iterations and a larger perturbation strength. To evaluate the proposed method's effectiveness in improving the model robustness, we compare our LAS-AWP with the AWP [9] trained with more iterations ($I_{\text{train}} = 15$) and a larger perturbation strength ($\epsilon_{\text{train}} = 15$). The results are shown in Table 1. Our LAS-AWP not only achieves a higher robust accuracy under all attack scenarios but also achieves a higher clean accuracy on clean images. We can attribute the improvements to using automatically generated attack strategies instead of more iterations and a larger perturbation strength.

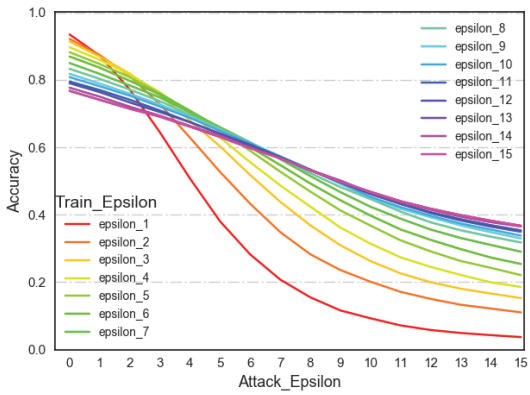

Figure 4. The influence of the maximal perturbation strength in PGD-AT [6]. Curves represent performance of PGD-AT [6] with different maximal perturbation strengths for training. X-axis represents the perturbation strength of the PGD attack. Y-axis represents the accuracy of adversarial examples generated by the PGD attack method.

7. Influence of Attack Parameters

In order to explore the influence of the maximal perturbation strength, the number of iteration, and the step size in adversarial training, we conduct an experiment on CIFAR10 by using the PGD-AT [6] with different maximal perturbation strengths, different numbers of iteration, or different step sizes to perform adversarial training. During testing, we use PGD attack with different perturbation strengths, different numbers of iteration, or different step sizes to evaluate the robustness of the trained target model. We use ResNet18 as the target model.

To explore the influence of the maximal perturbation strength, we use the PGD-AT [6] with different maximal perturbation strengths to train the target model. We then use PGD attack with different perturbation strengths to attack the target model. The number of iteration of the PGD attack is set to 10 and the step size is set to 2. Note that most works use the PGD attack method with the perturbation strength of 8, iteration of 10, and step size of 2 (PGD(8,10,2)) to evaluate model robustness. The results are shown in Fig. 4. '`Train_Epsilon`' represents the maximal perturbation strength used in adversarial training while '`Attack_Epsilon`' represents the perturbation strength of the PGD attack for evaluation.

Table 1. Test robustness (%) on the CIFAR-10 database using ResNet18. Number in bold indicates the best.

Method	Clean	PGD-10	PGD-20	PGD-50	C&W	AA
AWP($I_{\text{train}} = 10, \epsilon_{\text{train}} = 8$)	80.72	55.33	54.78	54.28	51.67	49.44
AWP($I_{\text{train}} = 10, \epsilon_{\text{train}} = 15$)	66.73	52.24	52.14	52.06	48.1	47.03
AWP($I_{\text{train}} = 15, \epsilon_{\text{train}} = 8$)	80.13	55.82	55.24	55.13	51.53	49.62
LAS-AWP(ours)	**83.03**	**56.45**	**55.76**	**55.43**	**53.06**	**50.77**

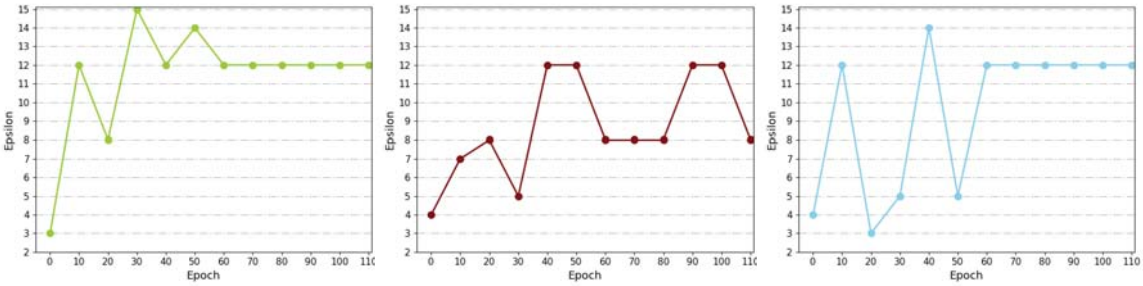

Figure 5. The evolution of the generated perturbation strength of several samples during the whole training process. X-axis represents the training epoch. Y-axis represents the perturbation strength.

Table 2. Hyperparameter selection. Test robustness (%) on the CIFAR-10 database using ResNet18. Number in bold indicates the best.

Method		Clean	PGD-10	AA
$\alpha = 2$	$\beta = 2$	**82.32**	54.27	49.46
	$\beta = 4$	82.05	**54.29**	**49.89**
	$\beta = 8$	82.01	54.28	49.65
$\alpha = 4$	$\beta = 2$	82.02	54.36	49.57
	$\beta = 4$	81.64	54.05	49.57
	$\beta = 8$	81.92	54.19	49.58
$\alpha = 8$	$\beta = 2$	81.73	54.10	49.28
	$\beta = 4$	82.18	54.30	49.38
	$\beta = 8$	82.10	54.35	49.52

Observations are summarized as follows. **First**, though using a large maximal perturbation strength for training improves the robustness, it decreases the clean accuracy. For example, when `Train_Epsilon` = 15, the clean accuracy (*i.e.,* `Attack_Epsilon` = 0) drops to 76.7% which is much worse than that of `Train_Epsilon` $\leq$ 8. **Second**, using a large `Train_Epsilon` can achieve better performance than using a small one only when the `Attack_Epsilon` is large. Comparing the performance of `Train_Epsilon` = 8 and `Train_Epsilon` = 15, the accuracy of `Train_Epsilon` = 15 is lower when `Attack_Epsilon` $<$ 8, while the accuracy is higher when `Attack_Epsilon` $>$ 8. **Third**, when `Attack_Epsilon` = 8, increasing `Train_Epsilon` could slightly improve the robustness, but gets sharp clean accuracy drop.

As mentioned above, when the perturbation strength of PGD attack is 8 for evaluation (*i.e.,* `Attack_Epsilon` = 8), using a large maximal perturbation strength can only slightly improve the robustness, but will hurt the clean accuracy a lot. **Hence, in the manuscript, the maximal perturbation strengths of other state-of-the-art adversarial training methods are set to the same as their original papers, *i.e.,* `Train_Epsilon` = 8.** The range of the maximal perturbation strength is set to [3,15] in our method.

To explore the influence of the number of iteration, we use the PGD-AT [6] with different numbers of iteration to train the target model. We then use PGD attack with different numbers of iteration to attack the target model. The perturbation strength of the PGD attack is set to 8 and the step size is set to 2. The results are shown in Fig. 2. '`Train_Iteration`' represents the number of iteration used in adversarial training while '`Attack_Iteration`' represents the number of iteration of the PGD attack for evaluation. As shown in Fig. 2, we can observe a similar phenomenon as in Fig. 4. Increasing the number of iteration could improve the robustness, but it will hurt the clean accuracy. When `Train_Iteration` is larger than 10, we can only slightly improve the robustness by increas-

ing `Train_Iteration`, but gets a slight drop in clean accuracy.

To explore the influence of the step size, we use the PGD-AT [6] with different step sizes to train the target model. We then use PGD attack with different step sizes to attack the target model. The maximal perturbation strength of the PGD attack is set to 8 and the number of iteration is set to 10. The results are shown in Fig. 3. '`Train_Step_Size`' represents the step size used in adversarial training while '`Attack_Step_Size`' represents the step size of the PGD attack for evaluation. As shown in Fig. 3, when `Train_Step_Size` is larger than 2, the clean accuracy is nearly the same, while the robustness has marginal changes.

8. Selection of Hyper-parameters

There are two trade-off hyper-parameters in the objective function of the proposed method, *i.e.,* α and β in Eq. (8) in the manuscript. In this section, we present the performance of our proposed LAS-PGD-AT [6] with different α and β pairs on CIFAR-10. The range of α and β is set to $\{2^1, 2^2, 2^3\}$. The results are shown in Table 2.

When $\alpha = 2$ and $\beta = 2$, LAS-PGD-AT [6] achieves the best performance in clean accuracy and PGD-10 attack. And when $\alpha = 2$ and $\beta = 4$, the proposed method achieves the best performance in APGD-T, FAB, SQUARE, and AA attacks. And it also achieves the competitive performance in clean accuracy and PGD-10 attack. Moreover, our method is not sensitive to the two hyper-parameters as the robustness and the clean accuracy does not change in a large range. Hence, in this paper, we set α to 2 and β to 4.

9. Illustration of the Evolution of Generated Perturbation Strength

Given a sample, our strategy network generates an attack strategy. As the maximal perturbation strength affects the performance the most, we illustrate how the generated perturbation strength changes during adversarial training. Fig. 5 presents the evolution of the generated perturbation strength of three randomly selected images during the whole training process.

It can be observed that the perturbation strength of the same sample changes dynamically during the training process. At the beginning of adversarial training, the perturbation strength is small. As the training process goes on and the robustness of the target network improves, the perturbation strength becomes larger.

10. Experiments on more databases

We conduct experiments under the same $L_\infty = 8$ as the manuscript. **1)** For GTSRB, following [2], we adopt ResNet18 as backbone. As shown in Table 3, LAS-AT improves the robust accuracy of three base models under all attack scenarios and also improves their clean accuracy. **1)** For ImageNet, following [8], we compare with PGD-AT using ResNet50 as backbone. As shown in Table 4, LAS-AT achieves the better clean and robust accuracy.

Table 3. Test robustness (%) on GTSRB using ResNet18.

Method	Clean	PGD-50	C&W	AA
Clean	**98.36**	15.60	16.31	13.07
PGD-AT [6]	90.49	62.11	62.83	60.80
TRADES [10]	88.17	63.02	62.86	61.35
AWP [9]	92.68	63.85	64.65	61.66
LAS-AT(ours)	92.27	64.98	64.48	62.64
LAS-TRADES(ours)	90.26	64.62	64.14	62.59
LAS-AWP(ours)	93.79	**66.68**	**67.56**	**64.44**

Table 4. Test robustness (%) on ImageNet using ResNet50.

Method	Clean	PGD-50	C&W	AA
PGD-AT	47.86	23.63	23.04	15.23
LAT-AT(ours)	**48.03**	**25.97**	**24.21**	**16.56**

Table 5. Test robustness (%) on CIFAR-10 using WRN34-10.

$\hat{a}$	Clean	PGD-50	C&W	AA
PGD-10	86.23	56.12	55.73	53.58
PGD-20	86.19	56.14	**55.76**	53.68
PGD-50	86.2	**56.21**	55.68	**53.75**
C&W	**86.29**	55.94	55.48	53.62

11. Selection of the Strategy

The strategy $\hat{a}$ is used to evaluate the robustness of the model updated with the current strategy, which can be any attack strategy. In our experiments, $\hat{a}$ is set to the widely used attack strategy, *i.e.*, PGD-10. Here we conduct experiments with different attack strategies $\hat{a}$ on CIFAR-10 with WRN34-10 as backbone, and find that our method is not affected too much by $\hat{a}$, as shown in Table. 5.

12. Discussion about the Training Efficiency

Compared to ordinary training, the most time-consuming part of AT is adversarial example (AE) generation. **1)** As claimed in Sec. 3.4 , our method calls AE generation to **update the target network** (T-net) with from 3 to 15 attack iterations (Line 588), and to **update the strategy network** (S-net) with 10 attack iterations. The S-net is updated once after k-times updates of the T-net. Thus, for each update of T-net, our LAS-AT takes $15 + 10/k$ attack iterations at most, while PGD-AT takes 10 iterations. **2)** In experiments, we set $k = 40$ (Line 640). Thus, in the worst case, LAS-AT takes 52.5% more than PGD-AT. As shown in Tab. 1

of the manuscript, the actual additional training cost of our method is about 40% of PGD-AT's cost. Our method can be scalable to large datasets.

References

[1] Qi-Zhi Cai, Min Du, Chang Liu, and Dawn Song. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018. 5

[2] Jiefeng Chen, Xi Wu, Yang Guo, Yingyu Liang, and Somesh Jha. Towards evaluating the robustness of neural networks learned by transduction. *arXiv preprint arXiv:2110.14735*, 2021. 8

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[4] Saehyung Lee, Hyungyu Lee, and Sungroh Yoon. Adversarial vertex mixup: Toward better adversarially robust generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 272–281, 2020. 5

[5] Chen Lin, Minghao Guo, Chuming Li, Xin Yuan, Wei Wu, Junjie Yan, Dahua Lin, and Wanli Ouyang. Online hyper-parameter learning for auto-augmentation strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6579–6588, 2019. 5

[6] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020. 1, 5, 6, 7, 8

[7] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, volume 1, page 2, 2019. 5

[8] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020. 8

[9] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33, 2020. 5, 6, 8

[10] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019. 5, 8

[11] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning*, pages 11278–11287. PMLR, 2020. 5

[12] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 5