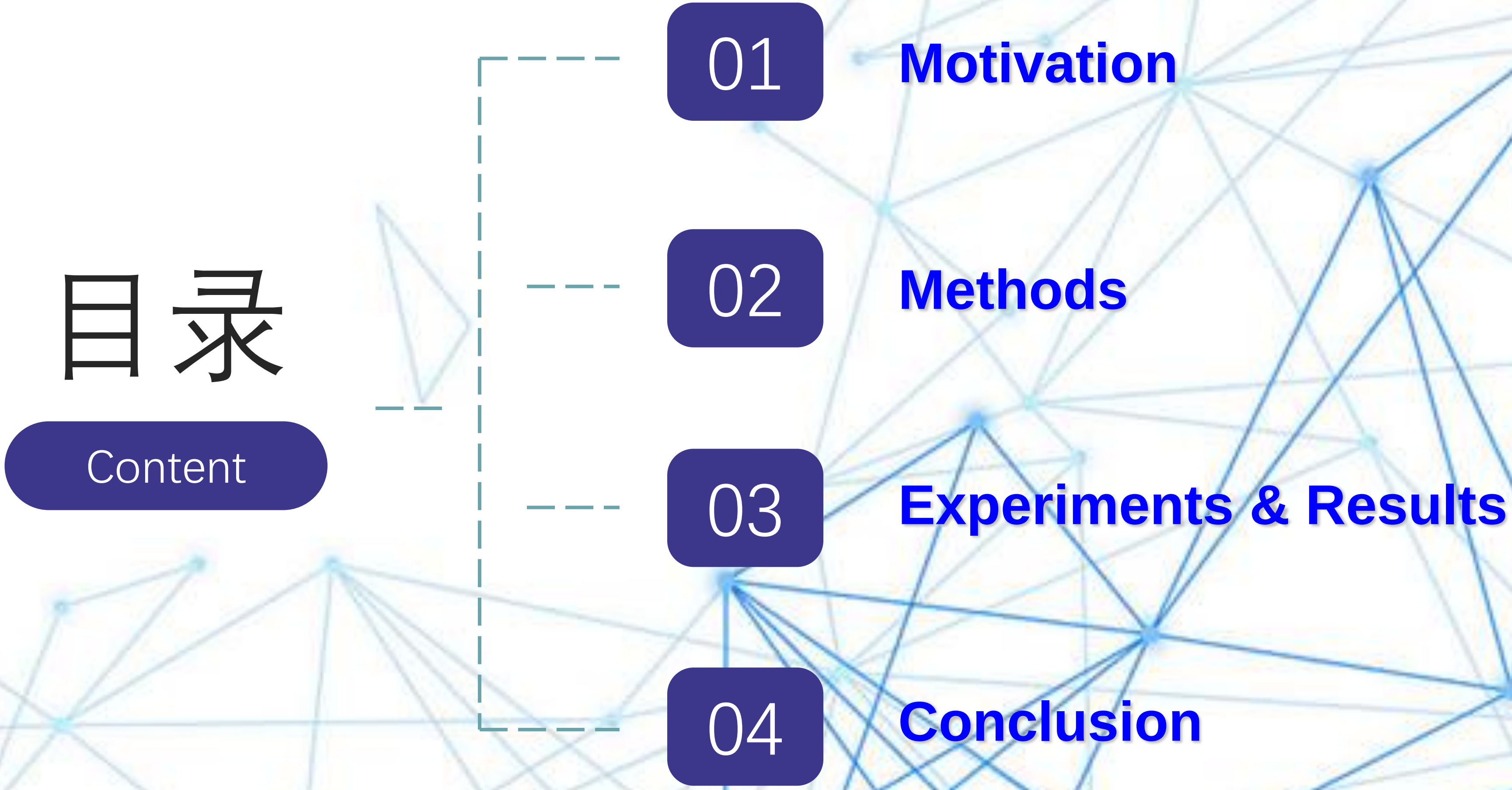
CVPR 2022

LAS-AT: Adversarial Training with Learnable Attack Strategy

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
2. School of Cyberspace Security, University of Chinese Academy of Sciences
3. Tencent, AI Lab 4. School of Data Science, The Chinese University of Hong Kong, Shenzhen
5. Secure Computing Lab of Big Data, Shenzhen Research Institute of Big Data, Shenzhen
6. School of Computer Science and Technology, University of Chinese Academy of Sciences

CVPR JUNE 19-24 2022 NEW ORLEANS • LOUISIANA



目录

Content

01 **Motivation**

02 **Methods**

03 **Experiments & Results**

04 **Conclusion**

01

Motivation

Motivation

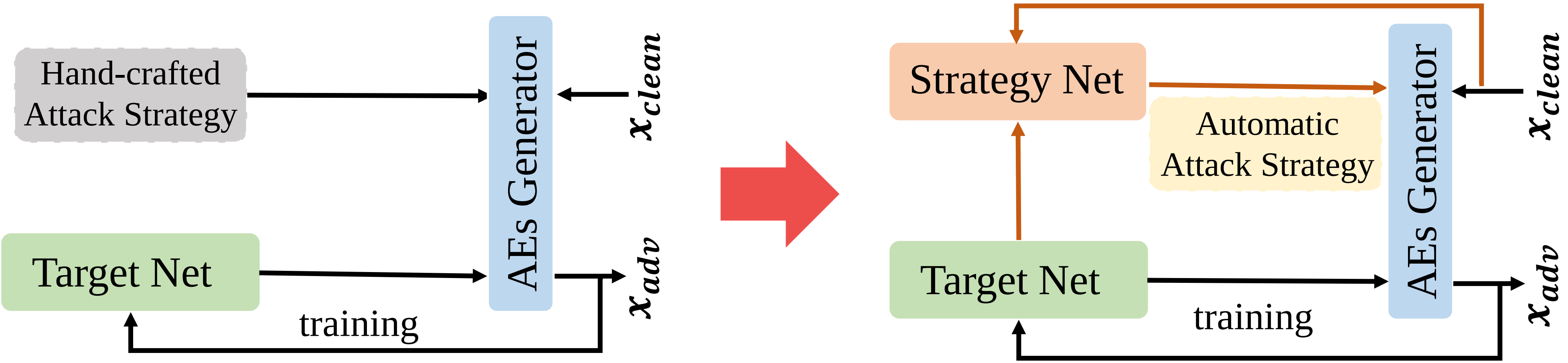
$$\min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\max_{\boldsymbol{\delta}\in\Omega} \mathcal{L}(f_{\mathbf{w}}(\mathbf{x}+\boldsymbol{\delta}), y)]$$

1. The inner maximization problem of standard AT is to generate adversarial examples by maximizing the classification loss.
2. The inner maximization problem of standard AT is to find model parameters by minimizing the classification loss on adversarial examples.
3. The inner maximization problem can be regarded as the attack strategy that guides the creation of AEs, which is the core to improve the model robustness. A training strategy is designed accordingly, which significantly improves the network’s robustness.

Motivation

$$\mathbf{x}_{adv} := \mathbf{x} + \boldsymbol{\delta} \leftarrow g(\mathbf{x}, \mathbf{a}, \mathbf{w})$$

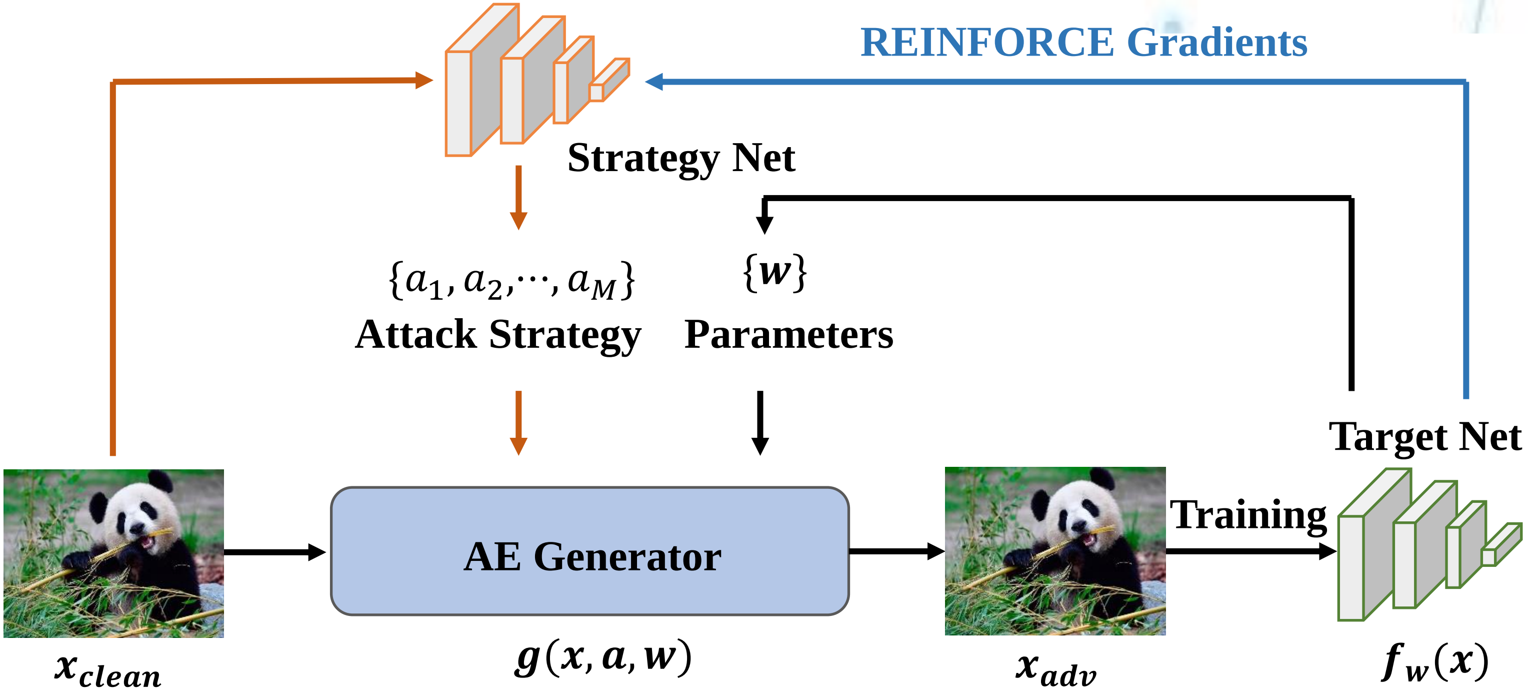
a is an attack strategy, i.e., the configuration of how to perform the adversarial attack. For example, PGD attack has three attack parameters, i.e., the attack step size, the attack iteration, and the maximal perturbation strength.



Conventional AT

LAS-AT

Contribution



Our main contributions are as follows:

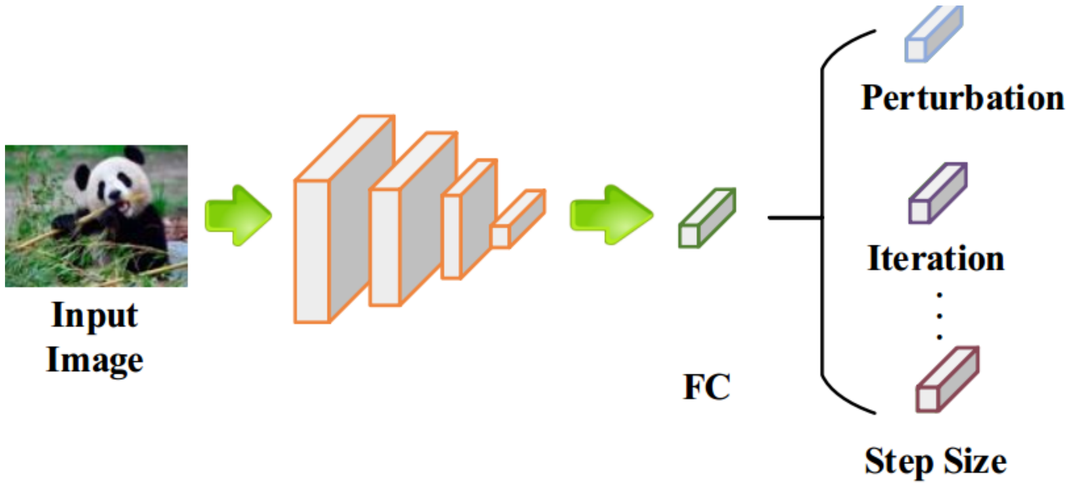
1. We propose a novel adversarial training framework by introducing the concept of “learnable attack strategy”, which learns to automatically produce sample-dependent attack strategies to generate AEs. Our framework can be combined with other state-of-the-art methods as a plug-and-play component.
2. We propose two loss terms to guide the learning of the strategy network, which involve explicitly evaluating the robustness of the target model and the accuracy of clean samples.
3. We conduct experiments and analyses on three databases to demonstrate the effectiveness of the proposed method, and the proposed method outperforms state-of-the-art adversarial training methods.

02

Method

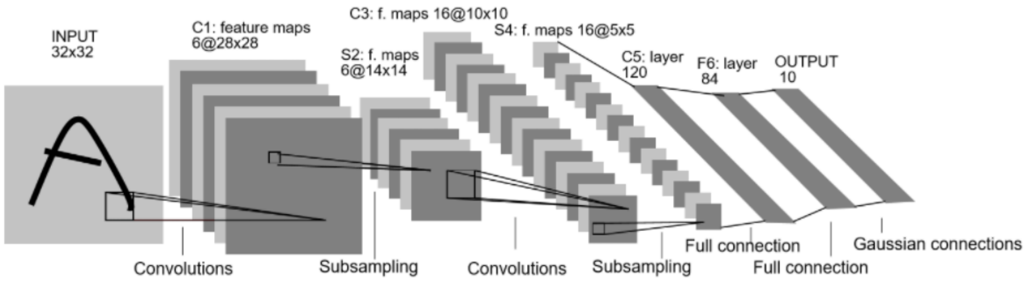
Method

Strategy Net



Given an image, the strategy network outputs an attack strategy, i.e., the configuration of how to perform the adversarial attack. A combination of the selected values for these attack parameters is an attack strategy. The strategy network captures the conditional distribution of a given x and θ.

Target Net



The target network is a convolutional network for image classification.

Adversarial Example Generator

$$\mathbf{x}_{adv} := \mathbf{x} + \boldsymbol{\delta} \leftarrow g(\mathbf{x}, \mathbf{a}, \mathbf{w})$$

g(·) is the PGD attack. The process is equivalent to solving the inner optimization problem, given an attack strategy a, i.e., finding the optimal perturbation to maximize the loss.

Method

Original Formulation of Adversarial Training:

$$\min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \mathcal{L}(f_{\mathbf{w}}(\mathbf{x}_{adv}), y)$$

Our Formulation of Adversarial Training:

$$\min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{a}\sim p(\mathbf{a}|\mathbf{x};\boldsymbol{\theta})} \mathcal{L}(f_{\mathbf{w}}(\mathbf{x}_{adv}), y)\right]$$

It can be observed that the two networks compete with each other in minimizing or maximizing the same objective. learns to improve attack strategies according to the given samples to attack the target network. At the beginning of the training phase, the target network is vulnerable, which a weak attack can fool. Hence, the strategy network can easily generate effective attack strategies. The strategies could be diverse because both weak and strong attacks can succeed. As the training process goes on, the target network becomes more robust. The strategy network has to learn to generate attack strategies that create stronger AEs. Therefore, the gaming mechanism could boost the robustness of the target network gradually along with the improvement of the strategy network

Method

Loss of adversarial training:

$$\mathcal{L}_1(\mathbf{w}, \boldsymbol{\theta}) := \mathcal{L}(f(\mathbf{x}_{adv}, \mathbf{w}), y)$$

Loss of Evaluating Robustness:

$$\mathcal{L}_2(\boldsymbol{\theta}) = -\mathcal{L}(f(\mathbf{x}^{\hat{\mathbf{a}}}_{adv}, \hat{\mathbf{w}}), y)$$

Loss of Predicting Clean Samples:

$$\mathcal{L}_3(\boldsymbol{\theta}) = -\mathcal{L}(f(\mathbf{x}, \hat{\mathbf{w}}), y)$$

Formal Formulation:

$$\min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\mathbf{a}\sim p(\mathbf{a}|\mathbf{x};\boldsymbol{\theta})} \left[\mathcal{L}_1(\mathbf{w}, \boldsymbol{\theta}) + \alpha\mathcal{L}_2(\boldsymbol{\theta}) + \beta\mathcal{L}_3(\boldsymbol{\theta}) \right] \right]$$

Method

Optimization of target network:

$$\min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \mathbb{E}_{\mathbf{a}\sim p(\mathbf{a}|\mathbf{x};\boldsymbol{\theta})}[\mathcal{L}_1(\mathbf{w},\boldsymbol{\theta})].$$

⬇

$$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \eta_1 \frac{1}{N} \sum_{n=1}^{N} \nabla_{\mathbf{w}} \mathcal{L}\left(f(\mathbf{x}_{adv}^{n}, \mathbf{w}^{t}), y_n\right)$$

Optimization of strategy network:

$$\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}),$$

$$\text{where } J(\boldsymbol{\theta}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\ \mathbb{E}_{\mathbf{a}\sim p(\mathbf{a}|\mathbf{x};\boldsymbol{\theta})}\ [\mathcal{L}_1 + \alpha\mathcal{L}_2 + \beta\mathcal{L}_3].$$

The biggest challenge of this optimization problem is that the process of AE generation is not differentiable, namely, the gradient can not be backpropagated to the attack strategy through the AEs. Moreover, there are some non-differentiable operations (e.g. choosing the iteration times) related to attack , which sets an obstacle to backpropagate the gradient to the strategy network.

Method

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \mathbb{E}_{\mathbf{a}\sim p(\mathbf{a}|\mathbf{x};\boldsymbol{\theta})} \left[\mathcal{L}_0\right] \\
&= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \int_{\mathbf{a}} \mathcal{L}_0 \cdot \nabla_{\boldsymbol{\theta}} p(\mathbf{a}|\mathbf{x};\boldsymbol{\theta}) d\mathbf{a} \\
&= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \int_{\mathbf{a}} \mathcal{L}_0 \cdot p(\mathbf{a}|\mathbf{x};\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log p(\mathbf{a}|\mathbf{x};\boldsymbol{\theta}) d\mathbf{a} \\
&= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \mathbb{E}_{\mathbf{a}\sim p(\mathbf{a}|\mathbf{x};\boldsymbol{\theta})} [\mathcal{L}_0 \cdot \nabla_{\boldsymbol{\theta}} \log p(\mathbf{a}|\mathbf{x};\boldsymbol{\theta})],
\end{aligned}$$

$$\Downarrow$$

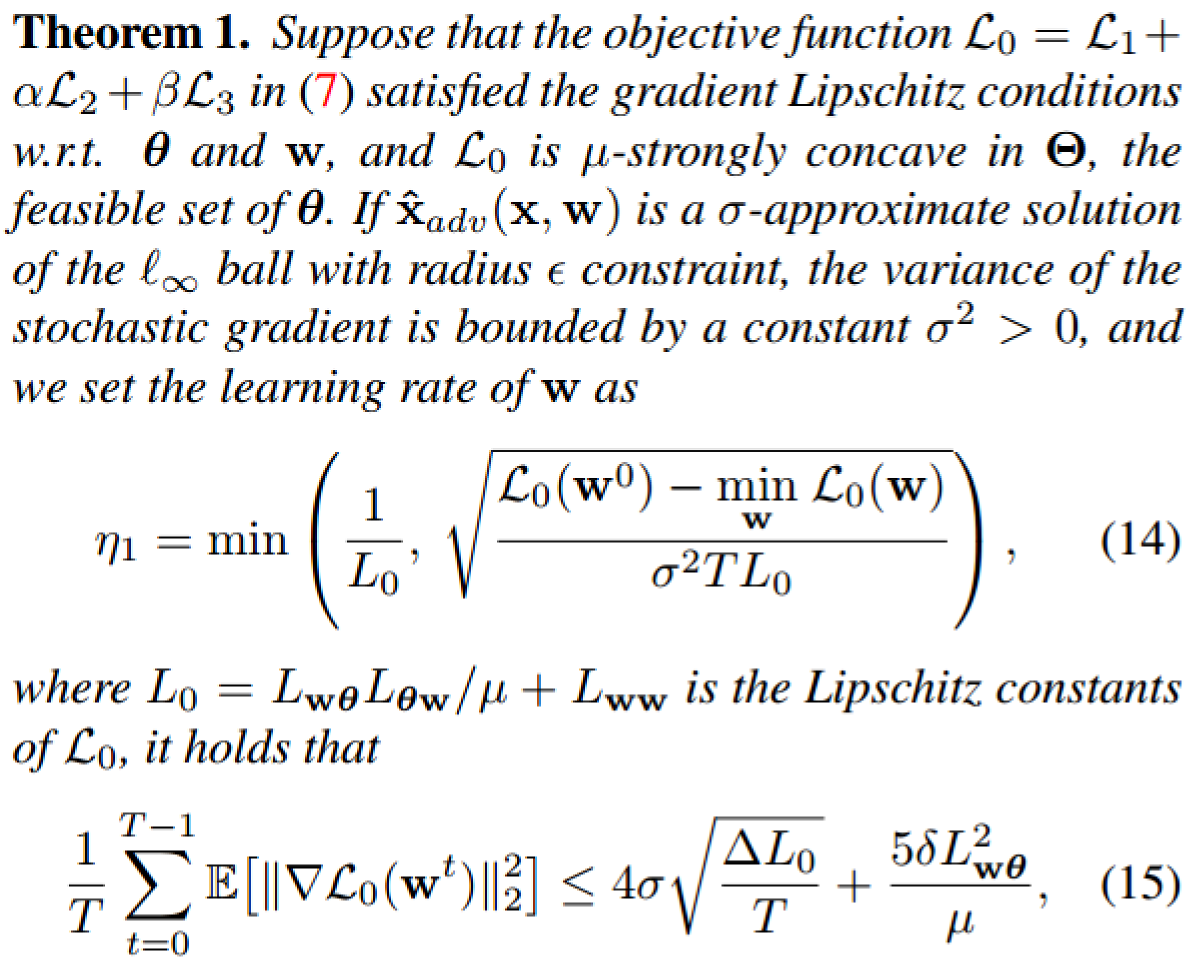
$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \approx \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_0(\mathbf{x}^n;\boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{a}^n|\mathbf{x}^n).$$

$$\Downarrow$$

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} + \eta_2 \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{t}),$$

Method

Convergence Analysis



Theorem 1. *Suppose that the objective function $\mathcal{L}_0 = \mathcal{L}_1 + \alpha\mathcal{L}_2 + \beta\mathcal{L}_3$ in (7) satisfied the gradient Lipschitz conditions w.r.t. $\boldsymbol{\theta}$ and $\mathbf{w}$, and $\mathcal{L}_0$ is μ-strongly concave in $\boldsymbol{\Theta}$, the feasible set of $\boldsymbol{\theta}$. If $\hat{\mathbf{x}}_{adv}(\mathbf{x}, \mathbf{w})$ is a σ-approximate solution of the ℓ_∞ ball with radius ϵ constraint, the variance of the stochastic gradient is bounded by a constant $\sigma^2 > 0$, and we set the learning rate of $\mathbf{w}$ as*

$$\eta_1 = \min\left(\frac{1}{L_0}, \sqrt{\frac{\mathcal{L}_0(\mathbf{w}^0) - \min_{\mathbf{w}} \mathcal{L}_0(\mathbf{w})}{\sigma^2 T L_0}}\right), \tag{14}$$

where $L_0 = L_{\mathbf{w}\boldsymbol{\theta}} L_{\boldsymbol{\theta}\mathbf{w}}/\mu + L_{\mathbf{w}\mathbf{w}}$ is the Lipschitz constants of $\mathcal{L}_0$, it holds that

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla\mathcal{L}_0(\mathbf{w}^t)\|_2^2] \leq 4\sigma\sqrt{\frac{\Delta L_0}{T}} + \frac{5\delta L_{\mathbf{w}\boldsymbol{\theta}}^2}{\mu}, \tag{15}$$

03

Experiments

Experiments

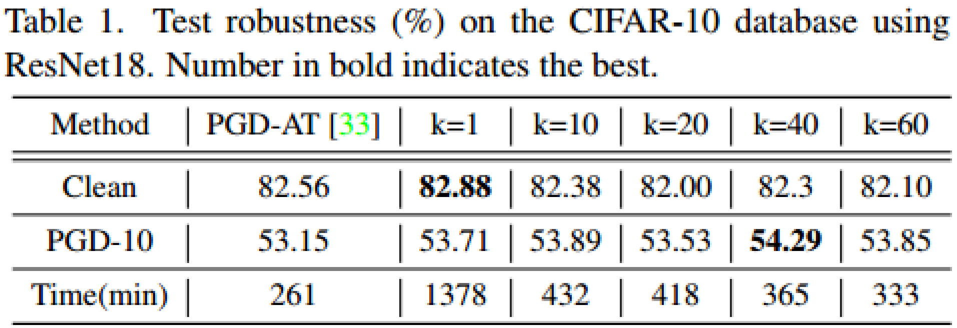


Table 1. Test robustness (%) on the CIFAR-10 database using ResNet18. Number in bold indicates the best.

Method	PGD-AT [33]	k=1	k=10	k=20	k=40	k=60
Clean	82.56	**82.88**	82.38	82.00	82.3	82.10
PGD-10	53.15	53.71	53.89	53.53	**54.29**	53.85
Time(min)	261	1378	432	418	365	333

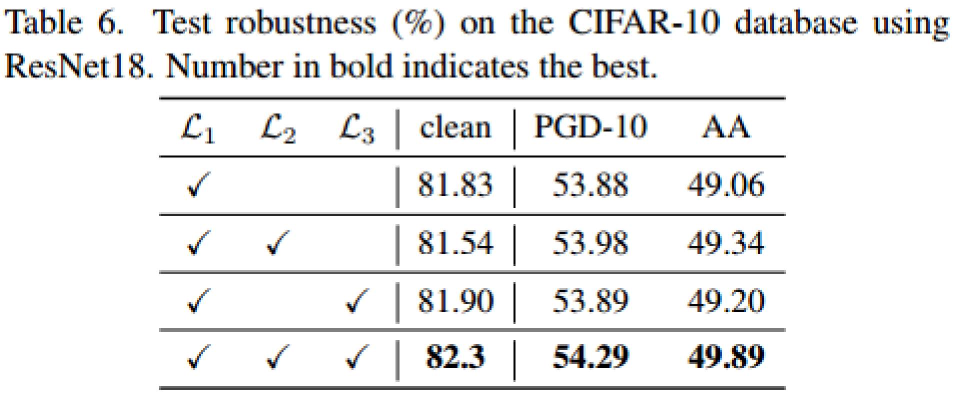


Table 6. Test robustness (%) on the CIFAR-10 database using ResNet18. Number in bold indicates the best.

$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$	clean	PGD-10	AA
✓			81.83	53.88	49.06
✓	✓		81.54	53.98	49.34
✓		✓	81.90	53.89	49.20
✓	✓	✓	**82.3**	**54.29**	**49.89**

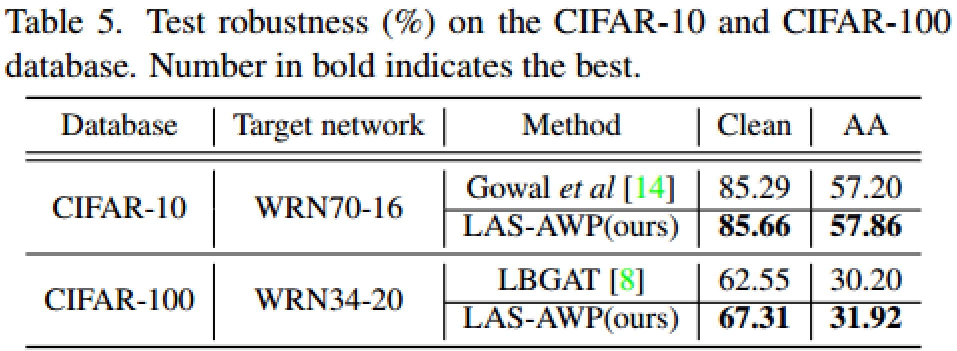


Table 5. Test robustness (%) on the CIFAR-10 and CIFAR-100 database. Number in bold indicates the best.

Database	Target network	Method	Clean	AA
CIFAR-10	WRN70-16	Gowal *et al* [14]	85.29	57.20
		LAS-AWP(ours)	**85.66**	**57.86**
CIFAR-100	WRN34-20	LBGAT [8]	62.55	30.20
		LAS-AWP(ours)	**67.31**	**31.92**

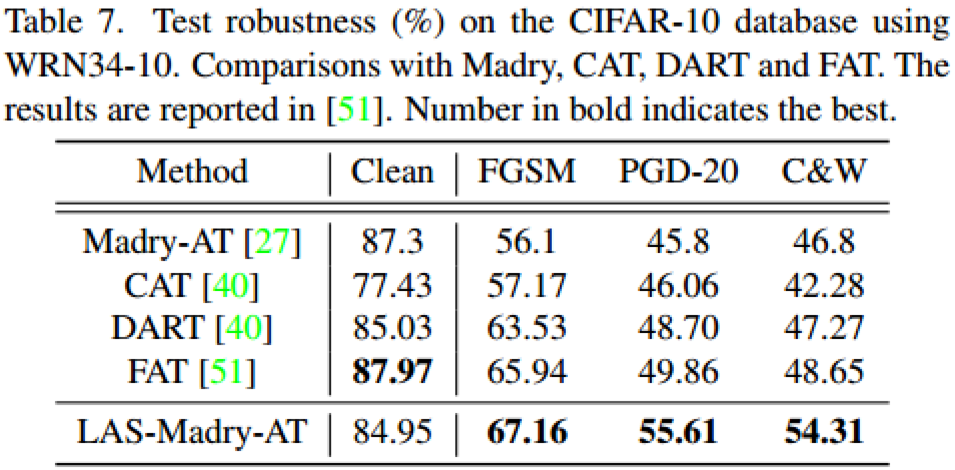


Table 7. Test robustness (%) on the CIFAR-10 database using WRN34-10. Comparisons with Madry, CAT, DART and FAT. The results are reported in [51]. Number in bold indicates the best.

Method	Clean	FGSM	PGD-20	C&W
Madry-AT [27]	87.3	56.1	45.8	46.8
CAT [40]	77.43	57.17	46.06	42.28
DART [40]	85.03	63.53	48.70	47.27
FAT [51]	**87.97**	65.94	49.86	48.65
LAS-Madry-AT	84.95	**67.16**	**55.61**	**54.31**

Experiments

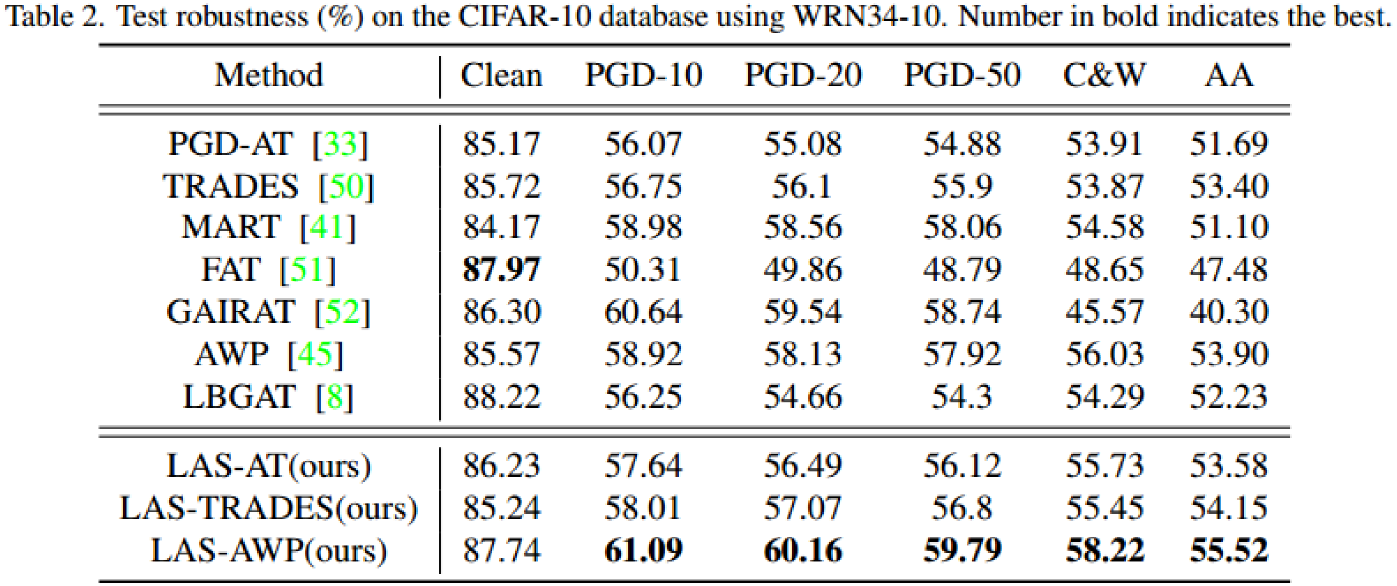


Table 2. Test robustness (%) on the CIFAR-10 database using WRN34-10. Number in bold indicates the best.

Method	Clean	PGD-10	PGD-20	PGD-50	C&W	AA
PGD-AT [33]	85.17	56.07	55.08	54.88	53.91	51.69
TRADES [50]	85.72	56.75	56.1	55.9	53.87	53.40
MART [41]	84.17	58.98	58.56	58.06	54.58	51.10
FAT [51]	**87.97**	50.31	49.86	48.79	48.65	47.48
GAIRAT [52]	86.30	60.64	59.54	58.74	45.57	40.30
AWP [45]	85.57	58.92	58.13	57.92	56.03	53.90
LBGAT [8]	88.22	56.25	54.66	54.3	54.29	52.23
LAS-AT(ours)	86.23	57.64	56.49	56.12	55.73	53.58
LAS-TRADES(ours)	85.24	58.01	57.07	56.8	55.45	54.15
LAS-AWP(ours)	87.74	**61.09**	**60.16**	**59.79**	**58.22**	**55.52**

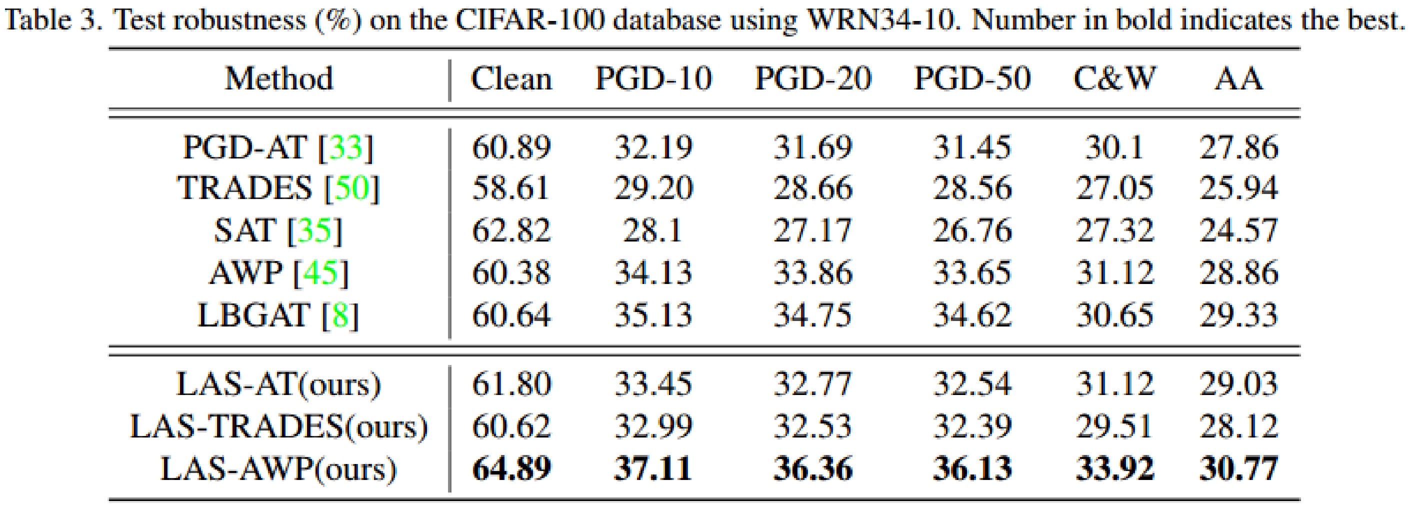


Table 3. Test robustness (%) on the CIFAR-100 database using WRN34-10. Number in bold indicates the best.

Method	Clean	PGD-10	PGD-20	PGD-50	C&W	AA
PGD-AT [33]	60.89	32.19	31.69	31.45	30.1	27.86
TRADES [50]	58.61	29.20	28.66	28.56	27.05	25.94
SAT [35]	62.82	28.1	27.17	26.76	27.32	24.57
AWP [45]	60.38	34.13	33.86	33.65	31.12	28.86
LBGAT [8]	60.64	35.13	34.75	34.62	30.65	29.33
LAS-AT(ours)	61.80	33.45	32.77	32.54	31.12	29.03
LAS-TRADES(ours)	60.62	32.99	32.53	32.39	29.51	28.12
LAS-AWP(ours)	**64.89**	**37.11**	**36.36**	**36.13**	**33.92**	**30.77**

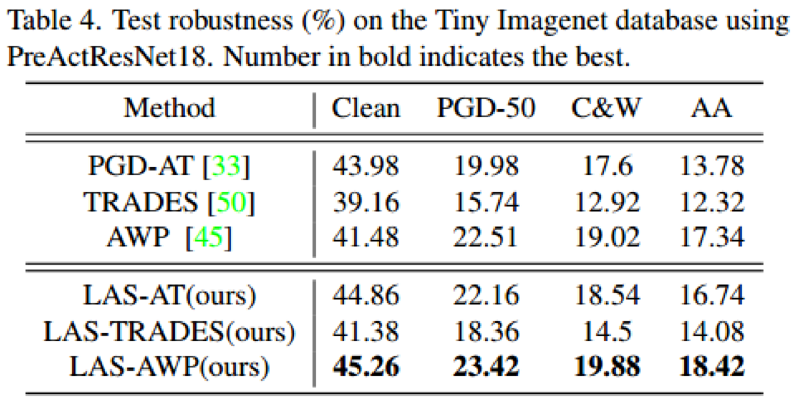
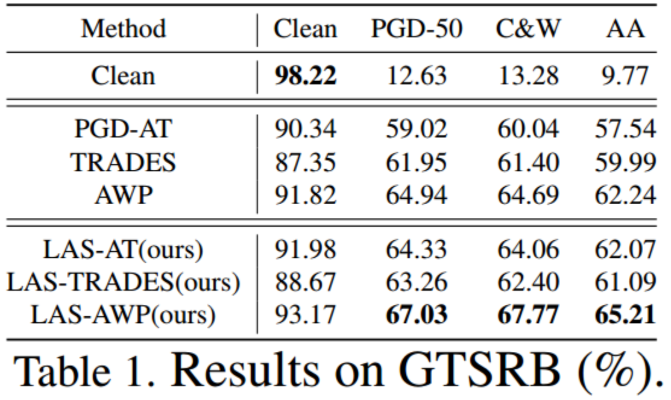


Table 4. Test robustness (%) on the Tiny Imagenet database using PreActResNet18. Number in bold indicates the best.

Method	Clean	PGD-50	C&W	AA
PGD-AT [33]	43.98	19.98	17.6	13.78
TRADES [50]	39.16	15.74	12.92	12.32
AWP [45]	41.48	22.51	19.02	17.34
LAS-AT(ours)	44.86	22.16	18.54	16.74
LAS-TRADES(ours)	41.38	18.36	14.5	14.08
LAS-AWP(ours)	**45.26**	**23.42**	**19.88**	**18.42**



Method	Clean	PGD-50	C&W	AA
Clean	**98.22**	12.63	13.28	9.77
PGD-AT	90.34	59.02	60.04	57.54
TRADES	87.35	61.95	61.40	59.99
AWP	91.82	64.94	64.69	62.24
LAS-AT(ours)	91.98	64.33	64.06	62.07
LAS-TRADES(ours)	88.67	63.26	62.40	61.09
LAS-AWP(ours)	93.17	**67.03**	**67.77**	**65.21**

Table 1. Results on GTSRB (%).

Experiments

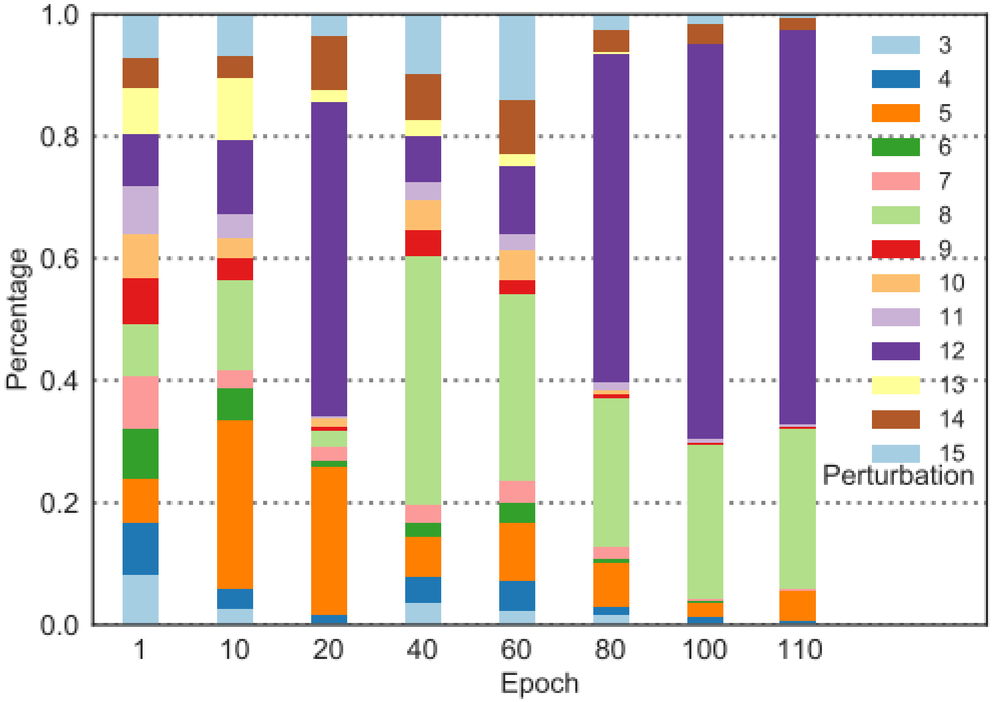


Figure 4. The distribution evolution of the maximal perturbation strength in LAS-PGD-AT during training.

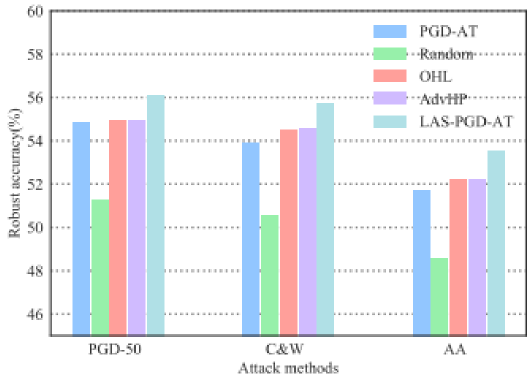


Figure 3. Comparisons with the hyper-parameter search methods using WRN34-10 on the CIFAR-10 database. x-axis represents the attack methods. y-axis represents the robust accuracy.

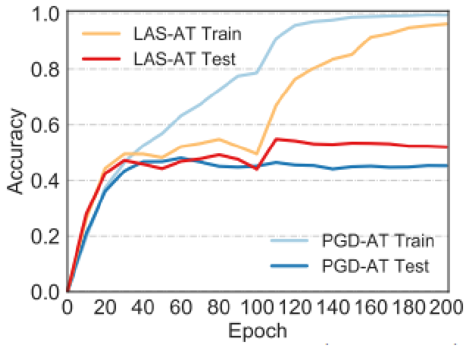
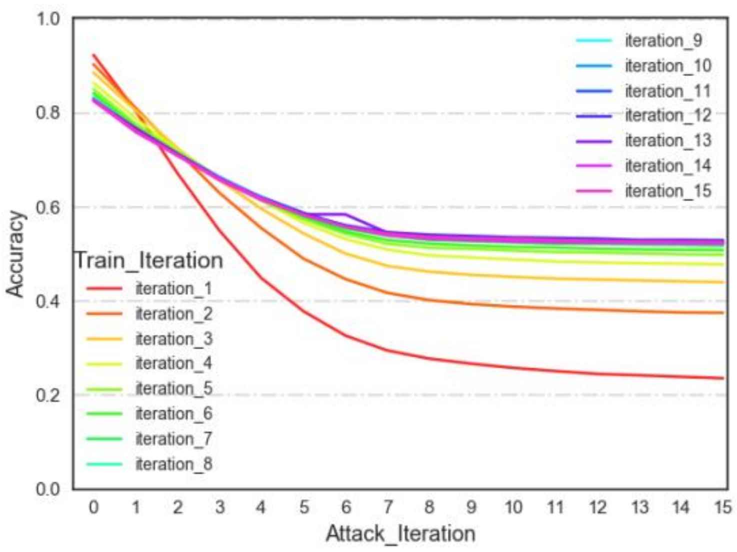
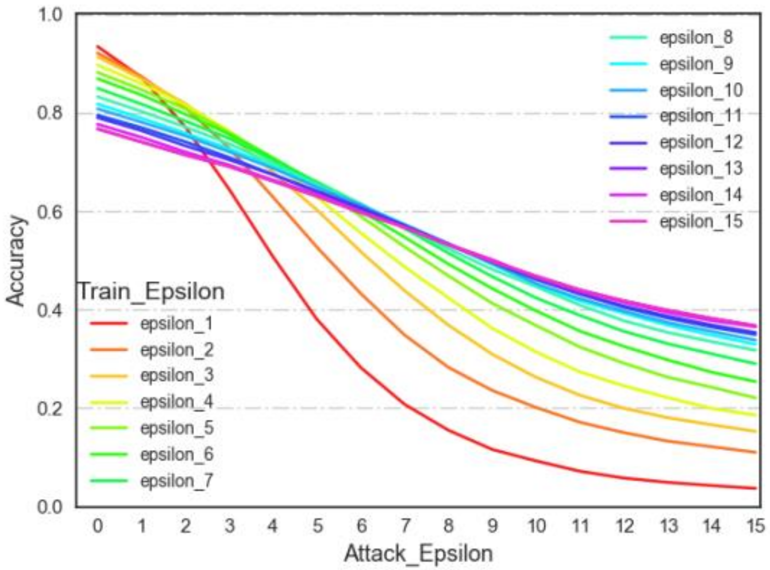


Figure 1. Robustness accuracy curves under PGD-10 attack on the training and test data of CIFAR-10.

Experiments





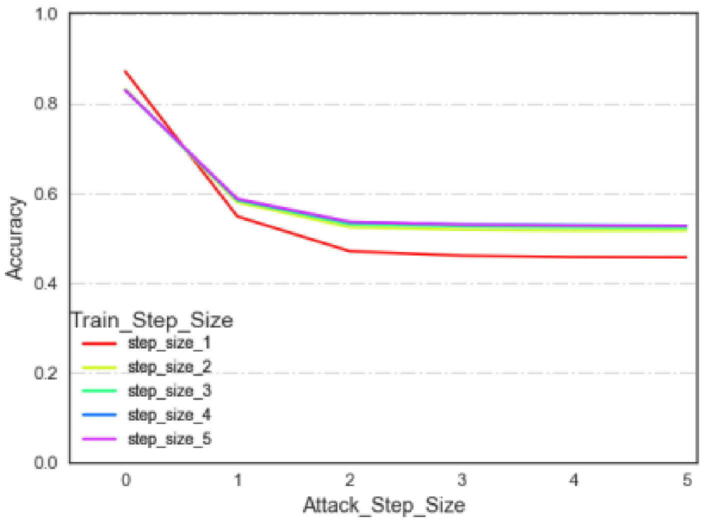


Table 1. Test robustness (%) on the CIFAR-10 database using ResNet18. Number in bold indicates the best.

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**

Experiments

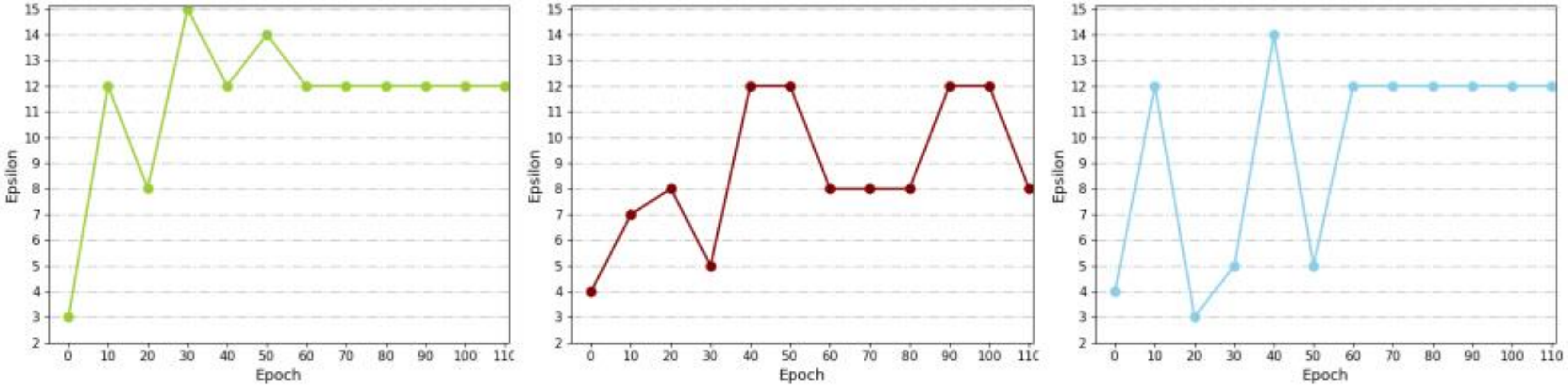
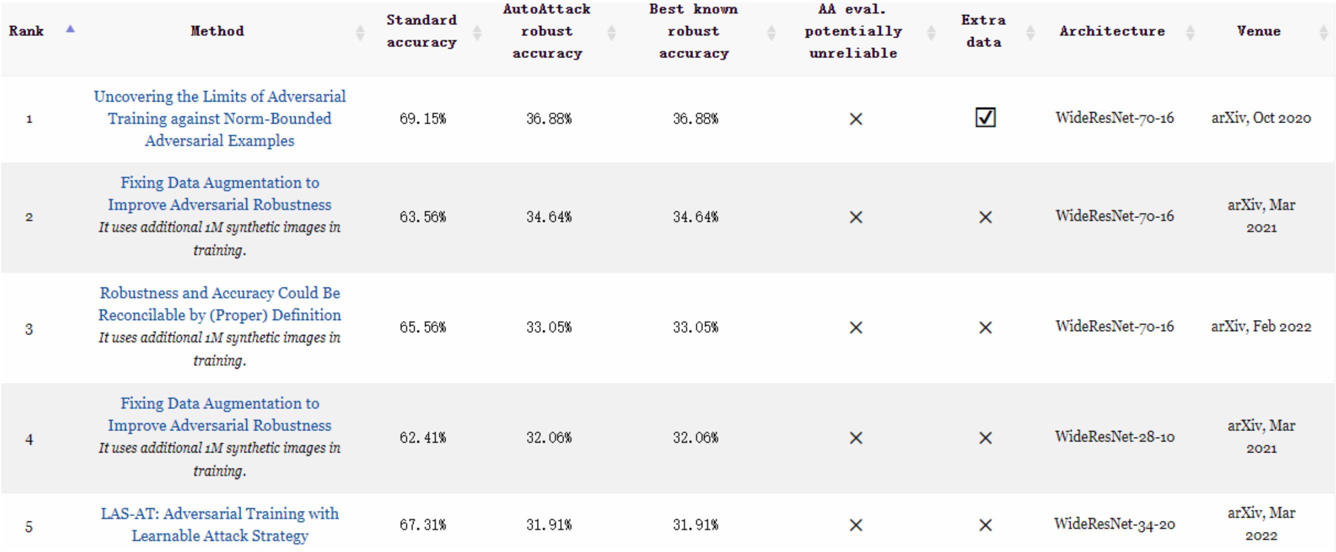
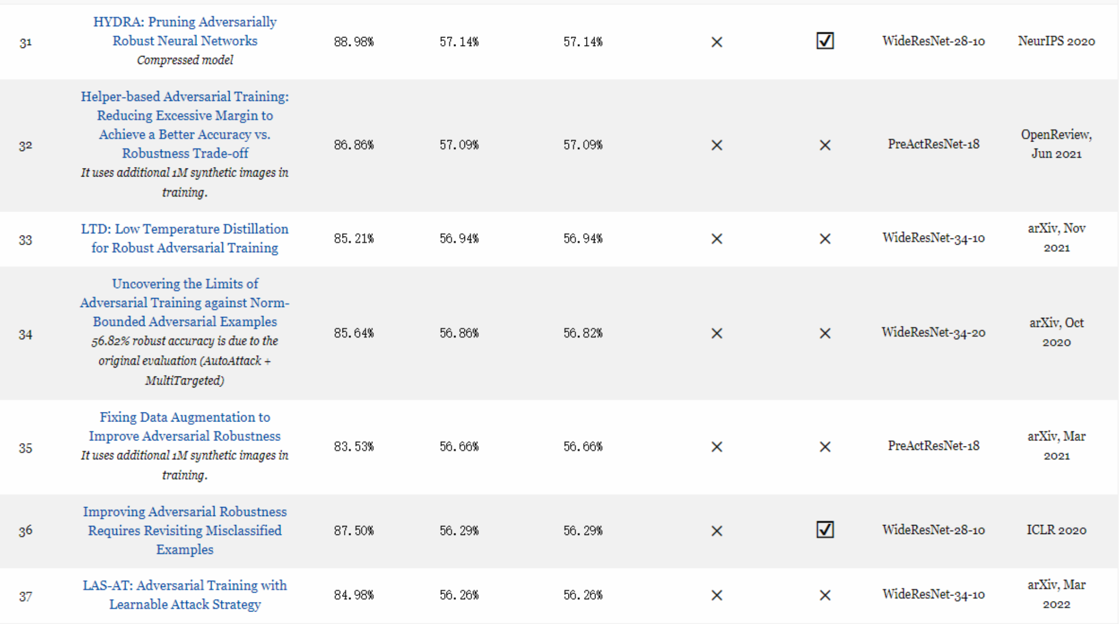


Figure 5. The evolution of the generated perturbation strength of several samples during the whole training process. X-axis represents the training epoch. Y-axis represents the perturbation strength.

Experiments



Rank	Method	Standard accuracy	AutoAttack robust accuracy	Best known robust accuracy	AA eval. potentially unreliable	Extra data	Architecture	Venue
1	Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples	69.15%	36.88%	36.88%	×	☑	WideResNet-70-16	arXiv, Oct 2020
2	Fixing Data Augmentation to Improve Adversarial Robustness *It uses additional 1M synthetic images in training.*	63.56%	34.64%	34.64%	×	×	WideResNet-70-16	arXiv, Mar 2021
3	Robustness and Accuracy Could Be Reconcilable by (Proper) Definition *It uses additional 1M synthetic images in training.*	65.56%	33.05%	33.05%	×	×	WideResNet-70-16	arXiv, Feb 2022
4	Fixing Data Augmentation to Improve Adversarial Robustness *It uses additional 1M synthetic images in training.*	62.41%	32.06%	32.06%	×	×	WideResNet-28-10	arXiv, Mar 2021
5	LAS-AT: Adversarial Training with Learnable Attack Strategy	67.31%	31.91%	31.91%	×	×	WideResNet-34-20	arXiv, Mar 2022



31	HYDRA: Pruning Adversarially Robust Neural Networks *Compressed model*	88.98%	57.14%	57.14%	×	☑	WideResNet-28-10	NeurIPS 2020
32	Helper-based Adversarial Training: Reducing Excessive Margin to Achieve a Better Accuracy vs. Robustness Trade-off *It uses additional 1M synthetic images in training.*	86.86%	57.09%	57.09%	×	×	PreActResNet-18	OpenReview, Jun 2021
33	LTD: Low Temperature Distillation for Robust Adversarial Training	85.21%	56.94%	56.94%	×	×	WideResNet-34-10	arXiv, Nov 2021
34	Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples *56.82% robust accuracy is due to the original evaluation (AutoAttack + MultiTargeted)*	85.64%	56.86%	56.82%	×	×	WideResNet-34-20	arXiv, Oct 2020
35	Fixing Data Augmentation to Improve Adversarial Robustness *It uses additional 1M synthetic images in training.*	83.53%	56.66%	56.66%	×	×	PreActResNet-18	arXiv, Mar 2021
36	Improving Adversarial Robustness Requires Revisiting Misclassified Examples	87.50%	56.29%	56.29%	×	☑	WideResNet-28-10	ICLR 2020
37	LAS-AT: Adversarial Training with Learnable Attack Strategy	84.98%	56.26%	56.26%	×	×	WideResNet-34-10	arXiv, Mar 2022

04

Conclusion

Conclusion

- **Learnable attack strategy:** we propose a novel adversarial training framework by introducing the concept of “learnable attack strategy”.
- **Two loss terms:** we also propose two loss terms that involve evaluating the robustness of the target network and predicting clean samples.
- **Superiority:** extensive experimental evaluations are performed on three benchmark databases to demonstrate the superiority of the proposed method.
- The code is released at *https://github.com/jiaxiaojunQAQ/LAS-AT* .