Efficient Adversarial Training with Transferable Adversarial Examples

Haizhong Zheng, Ziqi Zhang, Juncheng Gu,
Honglak Lee, Atul Prakash

Presenters: Chengkai Su, Faryab Haye, Zuoyi Li

Outline

1. Introduction

- State of Adversarial Attacks Against Deep Learning
- Adversarial Training
  - Improves robustness
  - Costly in computation time
    - can be 100x natural training time
- This paper address the adversarial training speed problem with ATTA (Adversarial Training with Transferable Examples)

Contributions

- Reveal the high transferability between models of neighboring epochs in adversarial training. With this property, the authors verify that the attack strength can be accumulated across epochs by reusing adversarial perturbations from the previous epoch.
- Propose a novel method (ATTA) for iterative attack based adversarial training with the objectives of both efficiency and effectiveness. It can generate the similar (or even stronger) adversarial examples with much fewer attack iterations via accumulating adversarial perturbations through epochs.
- Evaluation result shows that, with comparable model robustness, ATTA is 12:2 (14:1) faster than traditional adversarial methods on MNIST (CIFAR10). ATTA can also enhance model adversarial accuracy by up to 7:2% for MAT on CIFAR10.

Background & Motivation

2. Adversarial training and Transferability

2. 1 Adversarial Training

- **PGD-k (k-step projective gradient descent)**
    - Iterative attack is commonly used to generate strong adversarial examples

$$x_{t+1} = \Pi_{x+\mathcal{S}}(x_t + \alpha \ \mathrm{sgn}(\Delta_{x_t} L(\theta, x_t, y)))$$

In the above, x_t is the adversarial example in the t-th attack iteration, α is the attack step size, and Π is the projection function to project adversarial examples back to the allowed perturbation space $\mathcal{S}$.

Iterative Attack (PGD-k) Training Time

Dataset	Natural Training	Adversarial training		
		Training	Attack	Total
MNIST	39.7 sec	210 sec	3723 sec	3933 sec
CIAFR10	55min	214 min	1813 min	2027 min

Table 1: Training time of natural training and adversarial training. *Attack* column shows the time consumed in adversarial example generation.

2.2 Transferability of Adversarial Examples

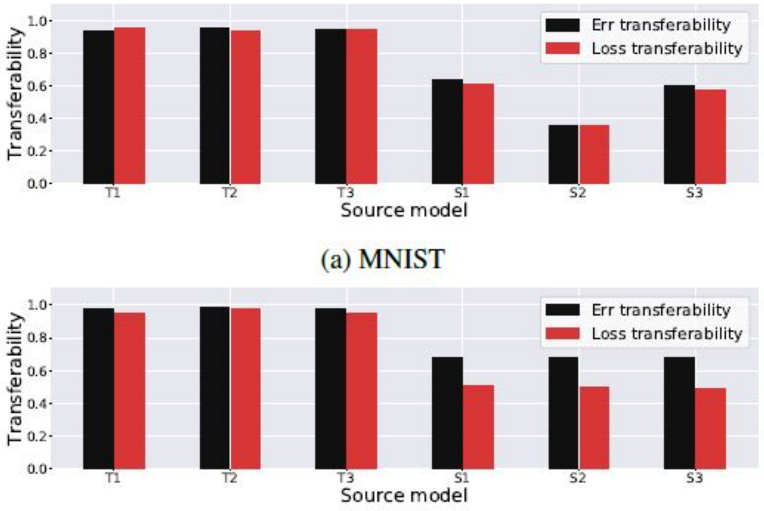
- **What is transferability in adversarial training?**
  - Adversarial examples generated for one model can stay adversarial for other models.
- **Substitute model training**
  - In order to train a source model $\boldsymbol{f}_s$(x), benchmark $\boldsymbol{f}_t$(x) instead of $\boldsymbol{y}$.
  - Used for black box attacks.

ATTA Design

3.1 Experiment on transferability and accumulate attack(Section 3)
3.2 ATTA Training design details(Section 4 in paper)

3.1 Experiment on transferability and accumulate attack

High Transferability between epochs



(a) MNIST

(b) CIFAR10

Figure 1: Error rate transferability and loss transferability with the different source models.

Definition:

- T -- Targeted Model
- S -- Model with another seed

Epoch n-3	Epoch n-2	Epoch n-1	Epoch n
Model T1	Model T2	Model T3	Model T
Model S1	Model S2	Model S3	Model S

Error Rate Transferability: number of adversarial examples being misclassified / number of adversarial examples being misclassified by model T

Loss Transferability: Loss Value caused by adversarial examples / Loss Value caused by adversarial examples on model T

Accumulative PGD-k

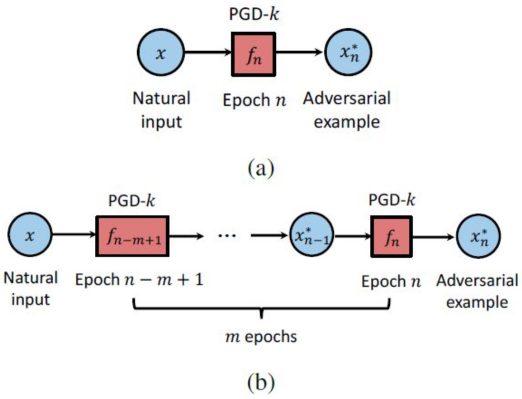


Figure 2: The traditional PGD-k attack (a) and the accumulative PGD-k attack through m epochs (b). The PGD-k attack is performed on the model in the red rectangle to get adversarial examples.

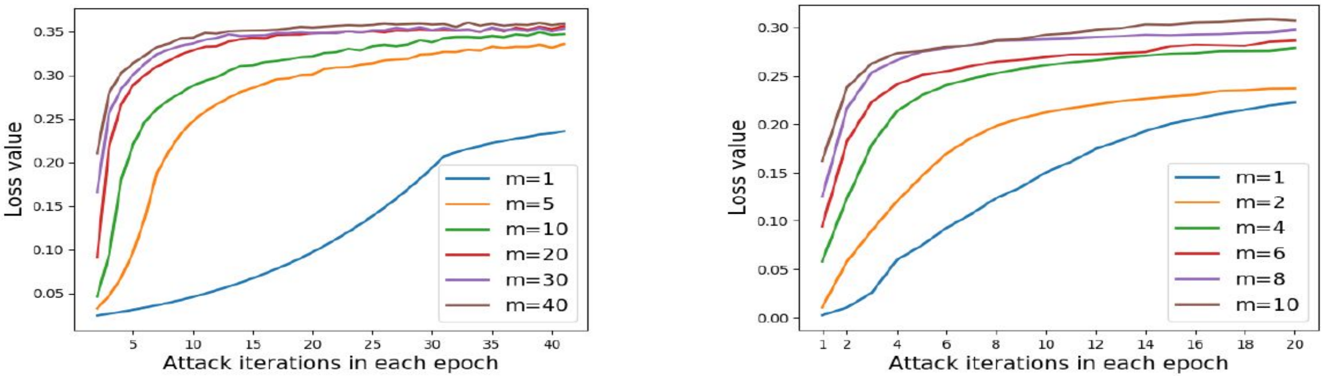
Hypothesis: Repeatedly reusing perturbations from previous epoch can accumulate attack strength epoch by epoch.

Goal: Design accumulative PGD-K attack to validate the hypothesis.

Accumulative PGD-k

Explanation: Number of Epochs being reused(m), Loss Value($\mathcal{L}(f_n(x_n^*), y)$) produced by two adversarial example.

Result: adversarial perturbations can be reused effectively. We could use fewer attack iterations(k) to generate same adversarial examples.



(a) MNIST

Figure 3: Given the number of epochs m, the relationship between the number of attack iterations k and the attack loss. $m = 1$ stands for the traditional PGD-k attack.

(b) CIFAR10

3.2. ATTA Training design details

Overview of training method

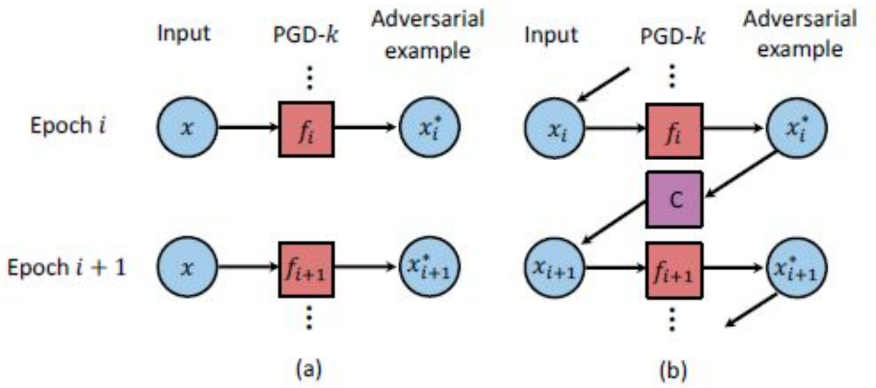


Figure 4: Traditional adversarial training (PGD-k) (a) and ATTA-k (b). C is the connection function which improves the transferability between epochs.

$$x_{i+1}^* = \mathcal{A}(f_{i+1}, x, y, C(x_i^*))$$

$\mathcal{A}$ is the attack algorithm.

C is a connection function

f_i is the model in the i-th epoch

x_i^* is the adversarial examples

Two challenges in Connection Function

Challenge 1: Data augmentation.

Challenge 2: Drastic model parameter change problem.

Challenge 1: Data Augmentation

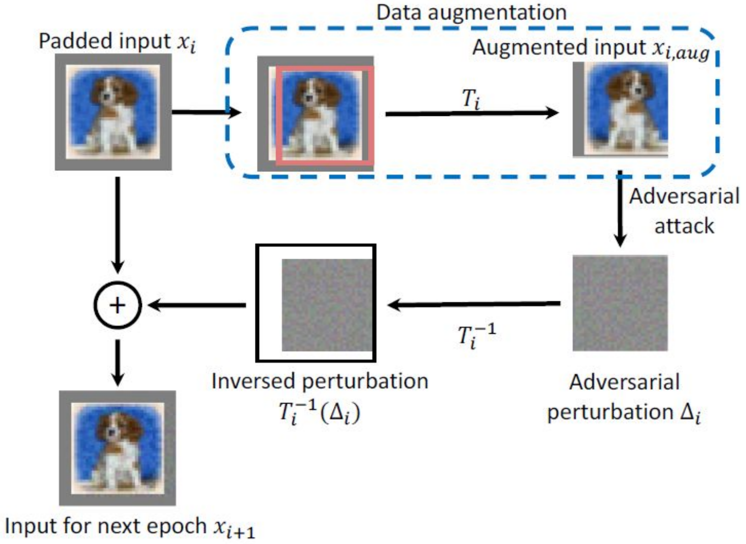


Figure 5: The workflow of inversed data augmentation.

Problem: Randomly transformation on original images cause mismatch of images between epochs.

Solution: Inverse data augmentation.

Challenge 2: Drastic parameter change

Problem: Model parameters tend to change drastically at the early stages of training

Solution: Periodically reset perturbation

MAT is the training method we studied the last time. It uses Cross Entropy Loss for the model predictions against the natural labels.

TRADES is another training algorithm which has a different “robustness” loss function.

tack algorithm $\mathcal{A}$, perturbation bound ϵ, the number of epochs to reset perturbation $reset$

Initialize θ

Initialize D by cloning D_{nat}

for $epoch = 1 \cdots N$ **do**

    for x_{nat}, y in D_{nat} and corresponding $x \in D$ **do**

        if $epoch \,\%\, reset = 0$ **then**

            $\beta \leftarrow$ a small random perturbation

            $x \leftarrow x_{nat} + \beta$

        end if

        Store the transformation T_{aug} for the inverse augmentation:

        $x_{aug}, x_{nat,aug}, T_{aug} \leftarrow \text{data\_aug}(x, x_{adv})$

        $x^* \leftarrow \mathcal{A}(f_\theta, x_{nat,aug}, y, x_{aug}, \epsilon)$

        $\theta \leftarrow \theta - \nabla_{f_\theta} \frac{\partial \mathcal{L}(f_\theta, x^*, y)}{\partial \theta}$

        $x \leftarrow \text{inverse\_aug}(x, x^*, T_{aug})$

    end for

end for

4. Evaluating Performance

Training Efficiency

Defense \ Attack		Natural	PGD-40	Time (sec)
MAT	PGD-1	99.52%	15.82%	226
	PGD-40	**99.37%**	**96.21%**	**3933**
	YOPO-5-10	99.15%	93.69%	789
	ATTA-1	**99.45%**	**96.31%**	**297**
	ATTA-40	99.23%	97.28%	4650
TRADES	PGD-1	99.41%	39.53%	583
	PGD-40	**98.89%**	**96.54%**	**6544**
	ATTA-1	**99.03%**	**96.10%**	**460**
	ATTA-40	98.21%	96.03%	4660

Table 3: The result of different attacks on MNIST dataset.

Training Efficiency (ctd.)

Defense \ Attack		Natural	PGD-20	Time (min)
MAT	PGD-1	93.18%	22.3%	435
	PGD-3	89.95%	41.38%	785
	PGD-10	**87.49%**	**47.07%**	**2027**
	Free($m = 8$)	85.54%	47.68%	640
	YOPO-5-3	86.43%	48.24%	335
	ATTA-1	**85.71%**	**50.96%**	**134**
	ATTA-3	85.44%	52.56%	267
	ATTA-10	83.80%	54.33%	690
TRADES	PGD-1	93.58%	35.52%	592
	PGD-3	88.52%	54.20%	861
	PGD-10	**84.13%**	**56.6%**	**2028**
	YOPO-3-4 [3]	87.82%	46.13%	-
	ATTA-1	85.04%	54.50%	199
	ATTA-3	**84.23%**	**56.36%**	**320**
	ATTA-10	83.67%	57.34%	752

Table 4: The result of different attacks on CIFAR10 dataset.

Training Efficiency (ctd.)

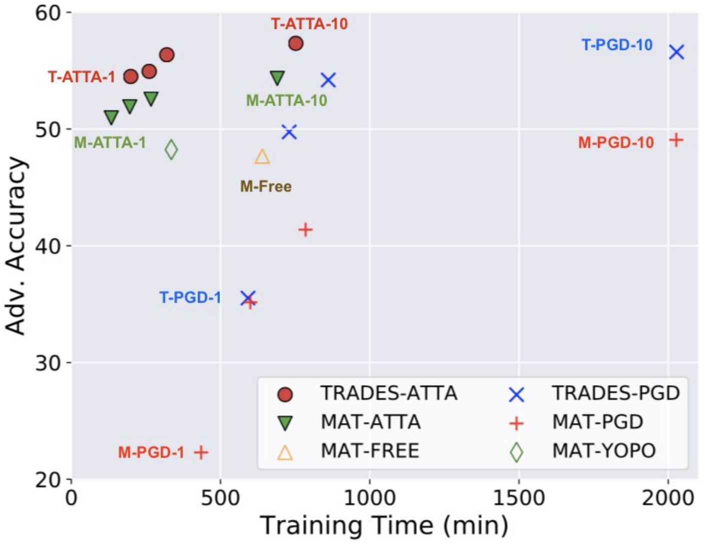


Figure 6: The scatter plot presents adversarial accuracy against PGD-20 attack and training time of different adversarial training methods on CIFAR10.

Defense \ Attack		Natural	PGD-20	Time (min)
MAT	PGD-1	93.18%	22.3%	435
	PGD-3	89.95%	41.38%	785
	PGD-10	**87.49%**	**47.07%**	**2027**
	Free($m = 8$)	85.54%	47.68%	640
	YOPO-5-3	86.43%	48.24%	335
	ATTA-1	**85.71%**	**50.96%**	**134**
	ATTA-3	85.44%	52.56%	267
	ATTA-10	83.80%	54.33%	690
TRADES	PGD-1	93.58%	35.52%	592
	PGD-3	88.52%	54.20%	861
	PGD-10	**84.13%**	**56.6%**	**2028**
	YOPO-3-4 [3]	87.82%	46.13%	-
	ATTA-1	85.04%	54.50%	199
	ATTA-3	**84.23%**	**56.36%**	**320**
	ATTA-10	83.67%	57.34%	752

Table 4: The result of different attacks on CIFAR10 dataset.

Defense Under Other Attacks

Defense	PGD100	FGSM	CW20
MNIST			
M-PGD-40	94.69%	97.37%	99.06%
M-ATTA-40	96.85%	98.55%	98.02%
CIFAR10			
M-PGD-10	46.77%	63.5%	56.7%
M-ATTA-10	52.6%	63.49%	75.37%
T-PGD-10	55.36%	63.02%	79.4%
T-ATTA-10	56.39%	64.1%	82.01%

Table 5: The robustness comparison between ATTA and PGD under other attacks. The first 'M' and 'T' stand for MAT and TRADES, respectively.

ATTA with Image Net

Defense	Natural	PGD10	PGD50	PGD100
Free($m = 4$)	64.44%	43.52%	43.39%	43.40%
ATTA-2	60.70%	44.57%	43.57%	43.51%

Table 6: Evaluation of ATTA on ImageNet

Ablation Study (Inverse Data Augmentation)

Defense \ Attack	Natural	PGD-20
MAT(w/o d.a., w/o i.d.a.)	82.53%	41.64%
MAT(w/ d.a., w/o i.d.a.)	91.65%	42.55%
MAT(w/ d.a., w/ i.d.a.)	85.71%	50.96%
TRADES(w/o d.a., w/o i.d.a.)	81.87%	41.65%
TRADES(w/ d.a., w/o i.d.a.)	90.46%	48.38%
TRADES(w/ d.a., w/ i.d.a.)	85.04%	54.50%

Table 7: The accuracy under PGD attack of models trained by ATTA-1 with or without d.a.(data augmentation) and i.d.a. (inverse data augmentation).

Ablation Study (Attack Loss)

Defense \ Attack	Natural	PGD-20
TRADES loss	85.95%	52.08%
MAT loss	85.04%	54.50%

Table 8: The accuracy under PGD attack of models trained by TRADES(ATTA-1) with MAT loss and TRADES loss.

5. Conclusion

Conclusion

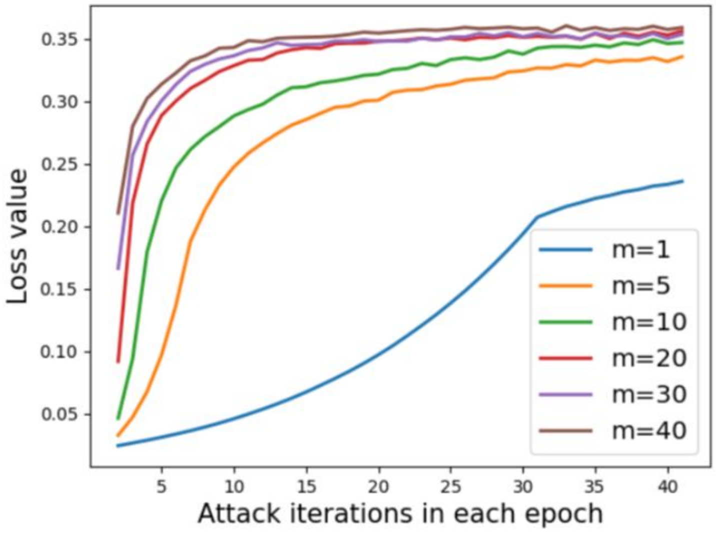
- ATTA is a new method for iterative attack based adversarial training
- More robust and faster than prior methods
- The key insight behind it is the high transferability between models from neighboring epochs
    - Based on this property, the attack strength in ATTA can be accumulated across epochs by repeatedly reusing adversarial perturbations from the previous epoch.
- ATTA is a generic method and can be applied to enhance the performance of other iterative attack based adversarial training methods.

6. Demo

Demo Continuation

Figure 3(a) comparison

Figure in Paper



(a) MNIST

Reproduction: