A2: Efficient Automated Attacker for Boosting Adversarial Training

Zhuoer Xu[1,2], Guanghui Zhu[1],
Changhua Meng[2], Shiwen Cui[2], Zhenzhe Ying[2],
Weiqiang Wang[2], Ming Gu[2], Yihua Huang[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]Tiansuan Lab, Ant Group

Adversarial Training

- In adversarial training, a defense model is trained on the worst-case adversarial perturbations generated by an attacker, which formulates a saddle point problem:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\in D} \overbrace{\left[\max_{\delta\in\mathbb{S}} l\left(f_\theta\left(\mathbf{x}+\delta\right), y\right)\right]}^{\textit{inner maximization}}$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\textit{outer minimization}}$$

- **Related Work**
  - Loss functions l: TRADES [Zhang et al., 2019] and MART [Wang et al., 2019];
  - Unlabeled data $D \bigcup D_{unlabel}$: RST [Carmon et al., 2019];
  - More perturbations $f_{\theta+\delta_\theta}(\mathrm{x}+\delta_{\mathrm{x}})$: AWP [Wu et al., 2020];
- **Our Work:** stronger perturbations yield more robust models.

Methodology

- **Adversarial Training (defense):** stronger perturbations yield more robust defense model.

$$\min_{\theta} \underbrace{\mathbb{E}_{(\mathbf{x},y)\in D} \overbrace{\left[\max_{\delta\in\mathbb{S}} l\left(f_\theta\left(\mathbf{x}+\delta\right), y\right)\right]}^{\text{inner maximization}}}_{\text{outer minimization}}$$

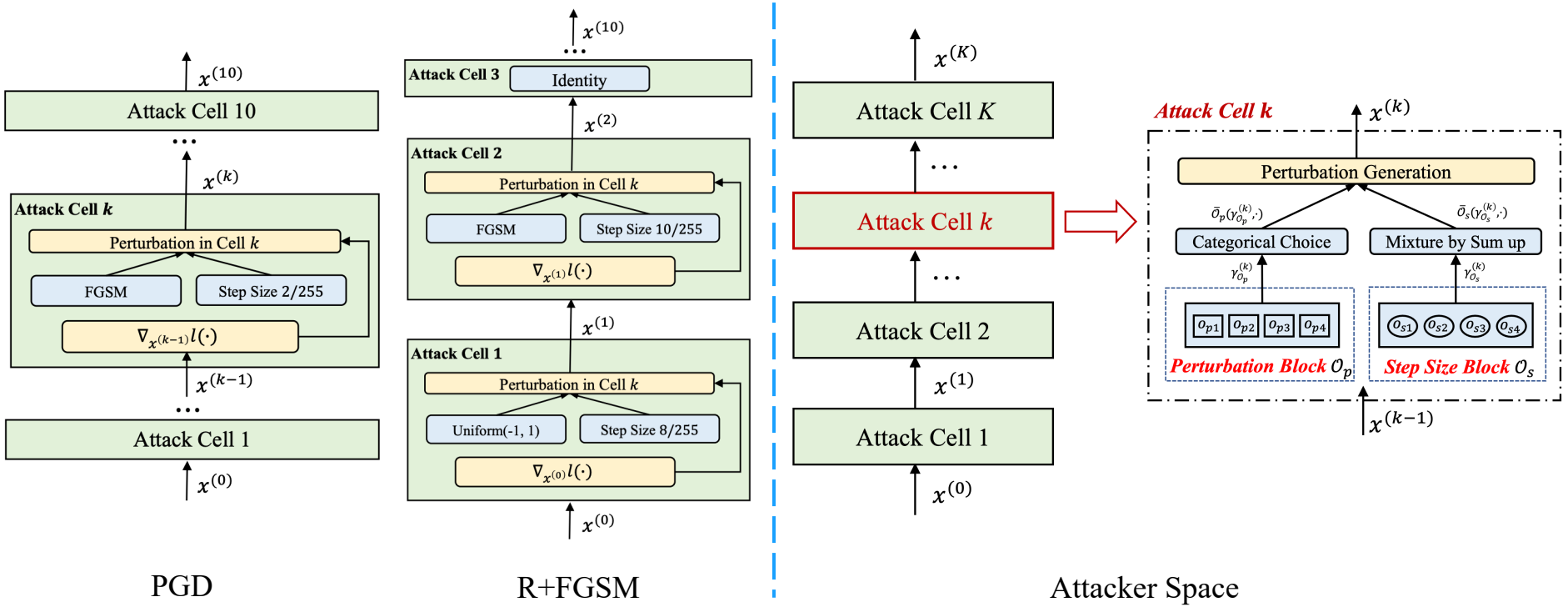
$$\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\in D}\left[l(f_\theta(\mathbf{x}+\delta_{\alpha^*}), y)\right]$$
$$\text{s.t. } \alpha^* = \arg\max_{\alpha} \mathbb{E}_{(\mathbf{x},y)\in D}\left[l\left(f_\theta\left(\mathbf{x}+\delta_\alpha\right), y\right)\right]$$

- **A^2 (attack):** a parameterized Automated Attacker to search in the attacker space for the best attacker against the defense model and examples.
- **Compositions:**
  - Attacker space: general enough to cover the existing attackers;
  - Architecture of A^2: leverage the information model and example to search for the best attacker;
  - Training and Inference of A^2: efficient to be used on-the-flying during training.

Search Space

- **Hierarchical attack space** by referring to existing attackers.

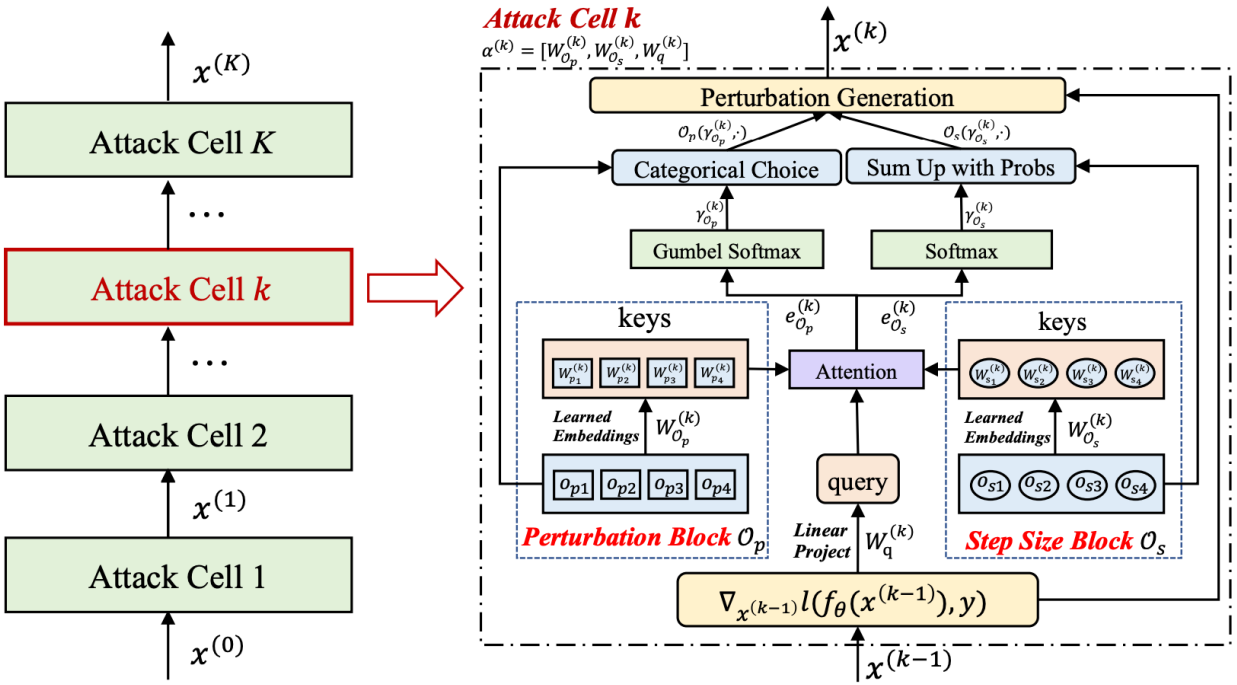
Cell	Block	Operation
One-Step Attacker	Perturbation & Step Size	Corresponding op



PGD R+FGSM Attacker Space

Automated Attacker A^2

- **Overview:**



- **Training & Inference:**

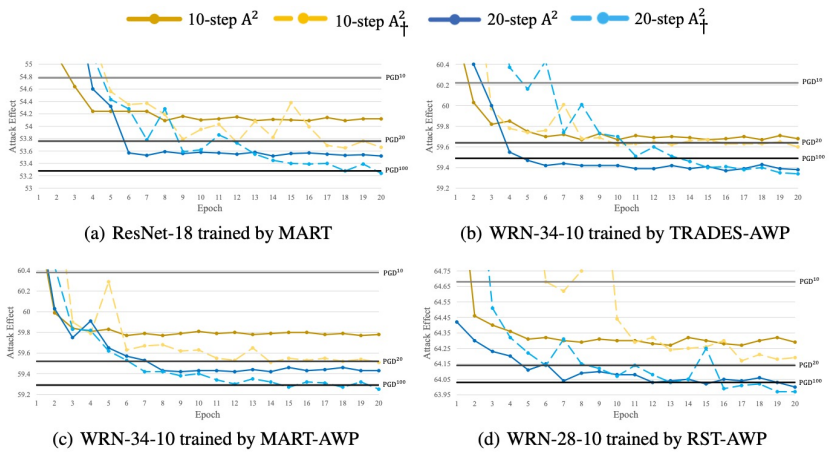
$$\frac{1}{M}\sum_{m=1}^{M} l\left(f_\theta\left(\mathbf{x} + \bar{C}([\phi(\kappa^{(m)}, e_{O_p}), \gamma_{O_s}], \nabla_{\mathbf{x}})\right), y\right)$$

Experiments

- **Attack Effect**

Defense	Natural	10-step PGD	10-step A^2	10-step $A^2_\dagger$	20-step PGD	20-step A^2	20-step $A^2_\dagger$	PGD^{100}
$MART^0$	83.07	54.78	54.09	53.65	53.76	53.52	**53.24**	53.28
$TRADES\text{-}AWP^1$	85.36	60.22	59.67	59.60	59.64	59.38	**59.34**	59.49
$MART\text{-}AWP^1$	85.60	60.38	59.76	59.51	59.52	59.42	**59.25**	59.29
$RST\text{-}AWP^1$	88.25	64.68	64.27	64.17	64.14	64.02	**63.97**	64.03

- **Attack Overhead**



(a) ResNet-18 trained by MART (b) WRN-34-10 trained by TRADES-AWP

(c) WRN-34-10 trained by MART-AWP (d) WRN-28-10 trained by RST-AWP

Attack Effect with training epoch

Step	PGD	A^2
1	19.75	20.03
10	147.09	157.61
20	287.76	302.51

Overhead in terms of clock time

Experiments

- **Robustness on Benchmark**

Defense	SVHN		CIFAR-10		CIFAR-100	
	Best	Last	Best	Last	Best	Last
AT	53.36	44.49	52.79	44.44	27.22	**20.82**
AT-A^2	**56.76**	**44.75**	**52.96**	**44.59**	**28.14**	20.28
AWP	59.12	55.87	55.39	54.73	30.71	30.28
AWP-A^2	**61.42**	**58.45**	**55.71**	**55.31**	**31.36**	**30.73**

- **Robustness on WideResNet**

Defense	Natural	FGSM	PGD^{20}	CW_∞	AutoAttack
AT	**87.30**	56.10	52.68	50.73	47.04
AT-A^2	84.54	**63.72**	**54.68**	**51.17**	**48.36**
TRADES	84.65	61.32	56.33	54.20	53.08
TRADES-A^2	**85.54**	**65.93**	**59.84**	**56.61**	**55.03**
MART	84.17	61.61	57.88	54.58	51.10
MART-A^2	**84.53**	**63.73**	**59.57**	**54.66**	**52.38**
AWP	85.57	62.90	58.14	55.96	54.04
AWP-A^2	**87.54**	**64.70**	**59.50**	**57.42**	**54.86**


ANT
GROUP


NJU-PASA Lab
Parallel Algorithm, System & Application for Big Data

Thanks