Automatic Discovery of Adaptive Attacks on Adversarial Defenses

SRILAB · ETH zürich

Chengyuan Yao, Pavol Bielik, Petar Tsankov, Martin Vechev

Introduction

Adversarial defenses are proposed to address the problem of adversarial examples. However, the authors of many defenses provide over-estimated robustness using fixed set of common techniques. These defenses are broken later with handcrafted adaptive attacks which are designed to reflect the defense mechanism. Yet this approach requires strong domain expertise.

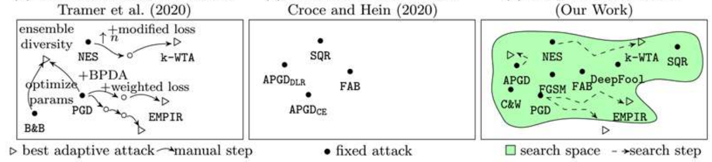
Our Work: We present an extensible tool A^3 that defines a search space over reusable blocks and automatically discovers an effective attack given the defense.

Motivation

Example Defenses	Robustness by authors		Handcrafted attacks (Tramer et al. 2020)
ME-Net (Yang et al. 2019)	53%	→	15%
Error Correcting Codes (Verma&Swami, 2019)	57%	→	5%
kWinner Takes All (Xiao et al. 2020)	51%	→	0.2%

Our work: automate this adaptive process

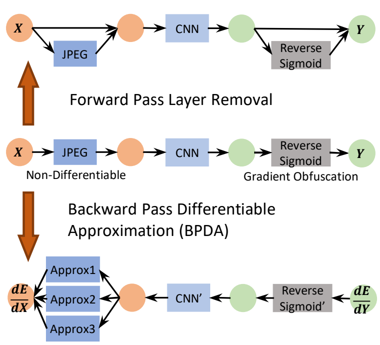
Robustness Evaluation Paradigms



Requires Manual Effort — Covers a Small Space — Automate the Manual Process

Network Transformation

X: Input, Y: Logits, E: Loss. Candidates: $4 \times 3 = 12$



Attack Algorithms & Parameters

Space Formulation:

```
(Attack Search Space)
  S ::=  S; S | randomize S | EOT S, n | repeat S, n |
         try S for n | Attack with params with loss ∈ L
```

High Level:

- 8 attacks in the search space - *FGSM, PGD, C&W, DeepFool, NES, APGD, FAB, SQR*
- Generic Parameters - *Randomize, Repeat, EOT*
- Attacks Specific Parameters
- Sequence of Attacks - *Evaluate attacks sequentially and return the first adversarial examples found*
- Try S for n - *set the runtime constraint for the attack to be n seconds*

Loss Functions

Space Formulation:

```
(Loss Function Search Space)
   L ::=  targeted Loss, n with Z | untargeted Loss with Z |
          targeted Loss, n - untargeted Loss with Z
   Z ::=  logits | probs
Loss ::=  CrossEntropy | HingeLoss | L1 | DLR | LogitMatching
```

- Difference between targeted and untargeted loss is the sign
- Logits/Probs means whether to add a softmax to logits

Loss Functions

$$\ell_{\texttt{CrossEntropy}} = -\sum_{i=1}^{K} y_i \log(Z(x)_i)$$

$$\ell_{\texttt{HingeLoss}} = \max(-Z(x)_y + \max_{i \neq y} Z(x)_i, -\kappa)$$

(Carlini & Wagner, 2017)

$$\ell_{\texttt{L1}} = -Z(x)_y$$

$$\ell_{\texttt{DLR}} = -\frac{Z(x)_y - \max_{i \neq y} Z(x)_i}{Z(x)_{\pi_1} - Z(x)_{\pi_3}}$$

(Croce & Hein, 2020b)

$$\ell_{\texttt{LogitMatching}} = \|Z(x') - Z(x)\|_2^2$$

Result

A^3 is evaluated on 24 defenses and compared with AutoAttack (AA)

- 10 cases: **3.0%-50.8%** additional adversarial examples.
- 13 cases: Typically **2x** faster attack time.

	CIFAR-10, l_∞	AA	A³	Δ	AA	A³	Speed-up	A³
A1	Madry et al. (2018)	44.78	**44.69**	-0.09	25	20	1.25×	88
A2†	Buckman et al. (2018)	2.29	**1.96**	-0.33	9	7	1.29×	116
A3†	Das et al. (2017)	0.59	**0.11**	-0.48	6	2	3.00×	40
A4	Metzen et al. (2017)	6.17	**3.04**	-3.13	21	13	1.62×	80
A5	Guo et al. (2018)	22.30	**12.14**	-10.16	19	17	1.12×	99
A6†	Pang et al. (2019)	4.14	**3.94**	-0.20	28	24	1.17×	237
A7	Papernot et al. (2015)	2.85	**2.71**	-0.14	4	4	1.00×	84
A8	Xiao et al. (2020)	19.82	**11.11**	-8.71	49	22	2.23×	189
A9	Xiao et al. (2020)ADV	64.91	**63.56**	-1.35	157	100	1.57×	179
A9'	Xiao et al. (2020)ADV	64.91	**17.70**	-47.21	157	2,280	0.07×	1,548
B10*	Gowal et al. (2021)	62.80	**62.79**	-0.01	818	226	3.62×	761
B11*	Wu et al. (2020)RTS	60.04	**60.01**	-0.03	706	255	2.77×	690
B12*	Zhang et al. (2021)	59.64	**59.56**	-0.08	604	261	2.31×	565
B13*	Carmon et al. (2019)	59.53	**59.51**	-0.02	638	282	2.26×	575
B14*	Sehwag et al. (2020)	**57.14**	57.16	0.02	671	429	1.56×	691
C15*	Stutz et al. (2020)	77.64	**39.54**	-38.10	101	108	0.94×	296
C15'	Stutz et al. (2020)	77.64	**26.87**	-50.77	101	205	0.49×	659
C16*	Zhang & Wang (2019)	**36.74**	37.11	0.37	381	302	1.26×	726
C17	Grathwohl et al. (2020)	**5.15**	5.16	0.01	107	114	0.94×	749
C18	Xiao et al. (2020)ADV	5.40	**2.31**	-3.09	95	146	0.65×	828
C19	Wang et al. (2019)	50.84	**50.81**	-0.03	734	372	1.97×	755
C20†	B11 + Defense in A3	60.72	**60.04**	-0.68	621	210	2.96×	585
C21†	C17 + Defense in A3	15.27	**5.24**	-10.03	261	79	3.30×	746
C22	B11 + Random Rotation	49.53	**41.99**	-7.54	255	462	0.55×	900
C23	C17 + Random Rotation	22.29	**13.45**	-8.84	114	374	0.30×	1,023
C24	Hu et al. (2019)	6.25	**3.07**	-3.18	110	56	1.96×	502

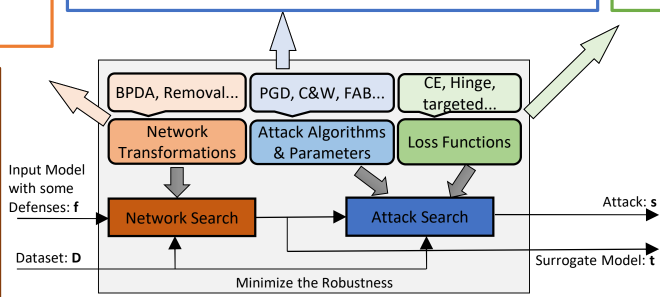
In addition, the attacks found by A^3 can reflect the defense mechanism. (Analysis for C15, C18, C24 are shown in the paper)

Network Search

Goal: Find the best surrogate model **t** to attack with. We use **t** to generate adversarial images but use **f** to evaluate

Search: Exhaustive search. Use PGD as the test attack to evaluate each candidate.

Complexity: Cheap to perform



Overview of A^3

Attack Search

Goal: Find the best sequence of attacks **s**

Search: For number of attacks in the **s**, repeat 1-3 (Greedy):

1. Get a set of samples from **D** for attack evaluation
2. Use Tree Parzen Estimation to select attacks
3. Use Successive Halving to select the best attack

Complexity: We constrained the per sample attack runtime. The search time bound is 4/3 of the attack runtime bound.