Rethinking the Backward Propagation for Adversarial Transferability

Xiaosen Wang[1*], Kangheng Tong[2*], Kun He[2†]

[1] Singular Security Lab

[2] School of Computer Science and Technology, Huazhong University of Science and Technology

Contact: xiaosen@hust.edu.cn

Homepage: http://xiaosenwang.com/

11/2/2023

Adversarial Example

NEURAL INFORMATION PROCESSING SYSTEMS

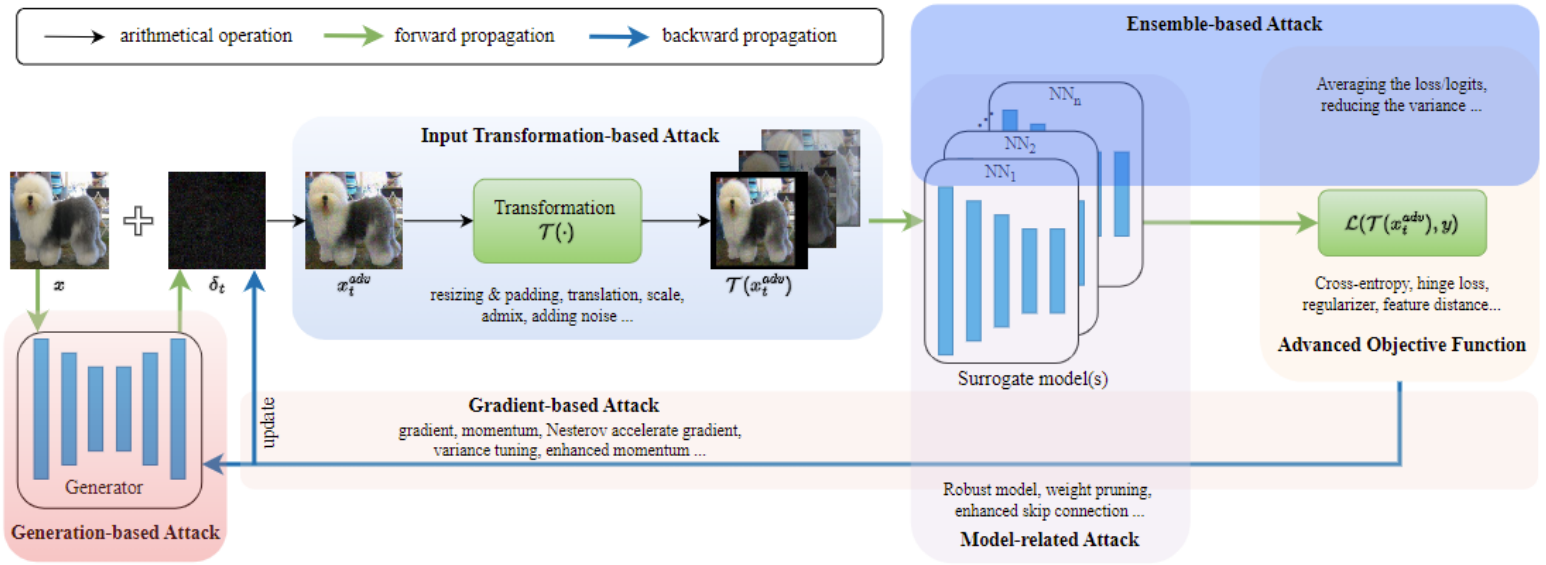
Adversarial examples are **indistinguishable** from legitimate ones by adding small perturbations, but lead to **incorrect model prediction**.

In the **black-box** setting, the attacker access limited or no information about the target model, making it applicable in the physical world. **Transfer-based attacks** generate adversarial examples on the **surrogate model** to fool the **target models**.

Adversarial Example

NEURAL INFORMATION PROCESSING SYSTEMS

Transfer-based attacks



- **Ghost** [Li et al., 2020] attacks a set of ghost networks generated by densely applying dropout at intermediate features.
- **SGM** [Wu et al., 2020] adjusts the decay factor to incorporate more gradients of the skip connections of ResNet to generate more transferable adversarial examples.
- **LinBP** [Guo et al., 2020] performs backward propagation in a more linear fashion by setting the gradient of ReLU as 1 and scaling the gradient of residual blocks.

Backward Propagation Attack

NEURAL INFORMATION PROCESSING SYSTEMS

Attacker's Goal: Adversarial attack optimizes the perturbation by maximizing the objective function:

$$x^{adv} = \underset{\|x'-x\|_p \leq \epsilon}{\text{argmax}} \; J(x', y; \theta)$$

White-box attacks often calculate the gradient to address such issue:

$$\nabla_x J(x, y; \theta) = \frac{\partial J(x, y; \theta)}{\partial f_{l+1}(z_l)} \left(\prod_{i=k+1}^{l} \frac{\partial f_{i+1}(z_i)}{z_i} \right) \frac{\partial z_{k+1}}{\partial z_k} \frac{\partial z_k}{\partial x}$$

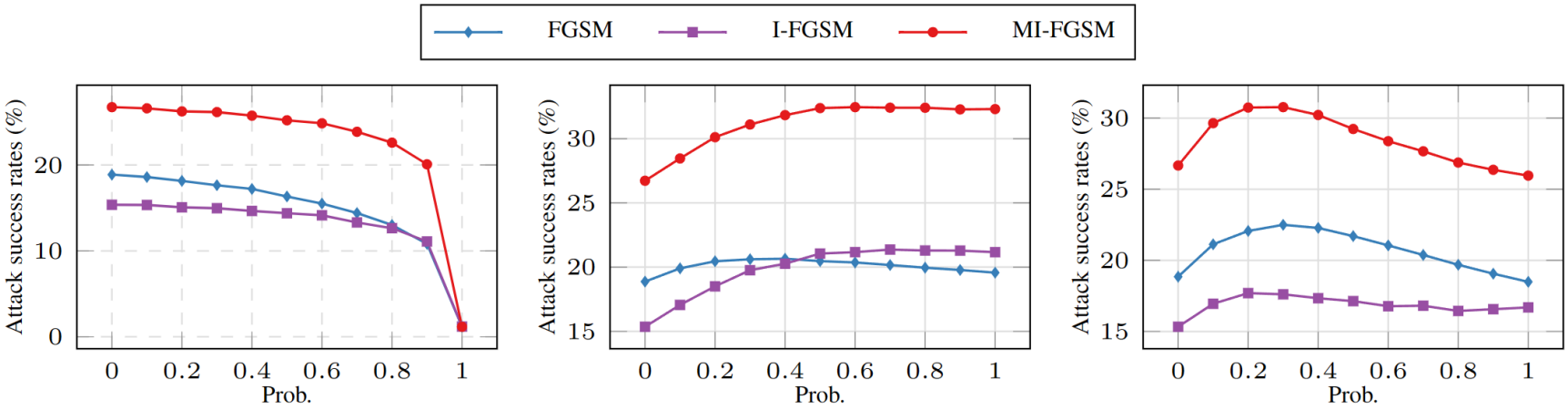
Assumption: The truncation of gradient $\nabla_x J(x, y; \theta)$ introduced by **non-linear layers** in the backward propagation process decays the adversarial transferability.

Backward Propagation Attack

NEURAL INFORMATION PROCESSING SYSTEMS

Verification:

- Randomly masking the gradient decays adversarial transferability
- Recovering the gradient of non-linear layers enhances adversarial transferability.



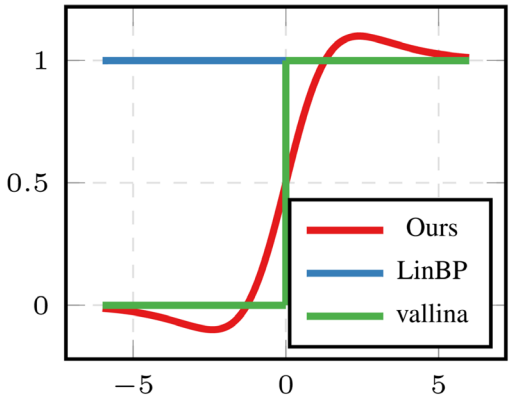
(a) Randomly mask the gradient (b) Recover the gradient of ReLU (c) Recover the gradient of max-pooling

Backward Propagation Attack

NEURAL INFORMATION PROCESSING SYSTEMS

ReLU: Use the derivative of **SiLU** to calculate the gradient of ReLU during the backward propagation process:

$$\frac{\partial z_{i+1}}{\partial z_i} = \sigma(z_i) \cdot \Big(1 + z_i \cdot \big(1 - \sigma(z_i)\big)\Big).$$



Max-pooling: Use the **softmax** function to calculate the gradient within each window w of the max-pooling operation:

$$\left[\frac{\partial z_{k+1}}{\partial z_k}\right]_{i,j,w} = \frac{e^{t \cdot z_{k,i,j}}}{\sum_{v \in w} e^{t \cdot v}}$$

0.1	-0.2	1.9	1.4
0.0	-0.5	2.3	0.7
-0.4	0.9	1.0	-2.0
0.7	0.6	0.5	1.7

Experimental Results

NEURAL INFORMATION PROCESSING SYSTEMS

Attacker	Method	Inc-v3	IncRes-v2	DenseNet	MobileNet	PNASNet	SENet	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
PGD	N/A	16.34	13.38	36.86	36.12	13.46	17.14	10.24	9.46	5.52
	SGM	23.68	19.82	51.66	55.44	22.12	30.34	13.78	12.38	7.90
	LinBP	27.22	23.04	59.34	59.74	22.68	33.72	16.24	13.58	7.88
	Ghost	17.74	13.68	42.36	41.06	13.92	19.10	11.60	10.34	6.04
	BPA	**35.36**	**30.12**	**70.70**	**68.90**	**32.52**	**42.02**	**22.72**	**19.28**	**12.40**
MI-FGSM	N/A	26.20	21.50	51.50	49.68	22.92	30.12	16.22	14.58	9.00
	SGM	33.78	28.84	63.06	65.84	31.90	41.54	19.56	17.48	10.98
	LinBP	35.92	29.82	68.66	69.72	30.24	41.68	19.98	16.58	9.94
	Ghost	29.76	23.68	57.28	56.10	25.00	34.76	17.10	14.76	9.50
	BPA	**47.58**	**41.22**	**80.54**	**79.40**	**44.70**	**54.28**	**32.06**	**25.98**	**17.46**
VMI-FGSM	N/A	42.68	36.86	68.82	66.68	40.78	46.34	27.36	24.20	17.18
	SGM	50.04	44.28	77.56	79.34	48.58	56.86	32.22	27.72	19.66
	LinBP	47.70	40.40	77.44	78.76	41.48	52.10	28.58	24.06	16.60
	Ghost	47.82	41.42	75.98	73.40	44.84	52.78	30.84	27.18	19.08
	BPA	**55.00**	**48.72**	**85.44**	**83.64**	**52.02**	**60.88**	**38.76**	**33.70**	**23.78**
ILA	N/A	29.10	26.08	58.02	59.10	27.60	39.16	15.12	12.30	7.86
	SGM	35.64	32.34	65.20	71.22	34.20	46.72	17.10	13.86	9.08
	LinBP	37.36	34.24	71.98	72.84	35.12	48.80	19.38	14.10	9.28
	Ghost	30.06	26.50	60.52	61.74	28.68	40.46	14.84	12.54	7.90
	BPA	**47.62**	**43.50**	**81.74**	**80.88**	**47.88**	**60.64**	**27.94**	**20.64**	**14.76**
SSA	N/A	35.78	29.58	60.46	64.70	25.66	34.18	20.64	17.30	11.44
	SGM	45.22	38.98	70.22	78.44	35.30	46.06	26.28	21.64	14.50
	LinBP	48.48	41.90	75.02	78.30	36.66	49.58	28.76	23.64	15.46
	Ghost	36.44	28.62	61.12	66.80	24.90	33.98	20.58	16.84	10.82
	BPA	**51.36**	**44.70**	**76.24**	**79.66**	**39.38**	**50.00**	**32.10**	**26.44**	**18.20**

Table 1: Untargeted attack success rates (%) of various adversarial attacks on nine models when generating the adversarial examples on ResNet-50 w/wo various model-related methods.

Experimental Results

NEURAL INFORMATION PROCESSING SYSTEMS

Attacker	Method	Inc-v3	IncRes-v2	DenseNet	MobileNet	PNASNet	SENet	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
PGD	SGM	23.68	19.82	51.66	55.44	22.12	30.34	13.78	12.38	7.90
	SGM+BPA	**43.44**	**38.14**	**77.66**	**81.50**	**41.42**	**53.56**	**27.20**	**22.58**	**14.70**
	LinBP	27.22	23.04	59.34	59.74	22.68	33.72	16.24	13.58	7.88
	LinBP+BPA	**39.08**	**34.80**	**77.80**	**76.86**	**40.50**	**50.26**	**25.66**	**22.46**	**15.10**
	Ghost	17.74	13.68	42.36	41.06	13.92	19.10	11.60	10.34	6.04
	Ghost+BPA	**34.62**	**29.28**	**69.48**	**69.20**	**29.98**	**41.60**	**22.68**	**18.88**	**11.48**
MI-FGSM	SGM	33.78	28.84	63.06	65.84	31.90	41.54	19.56	17.48	10.98
	SGM+BPA	**56.04**	**49.10**	**85.32**	**88.08**	**52.96**	**63.30**	**36.10**	**29.78**	**20.98**
	LinBP	35.92	29.82	68.66	69.72	30.24	41.68	19.98	16.58	9.94
	LinBP+BPA	**48.74**	**43.96**	**83.30**	**83.52**	**50.00**	**59.22**	**32.60**	**28.42**	**20.32**
	Ghost	29.76	23.68	57.28	56.10	25.00	34.76	17.10	14.76	9.50
	Ghost+BPA	**50.42**	**42.84**	**83.02**	**81.24**	**44.70**	**56.50**	**32.46**	**26.82**	**18.34**

Table 2: Untargeted attack success rates (%) of various baselines combined with our method using PGD and MI-FGSM. The adversarial examples are generated on ResNet-50.

Experimental Results

NEURAL INFORMATION PROCESSING SYSTEMS

Attacker	Method	HGD	R&P	NIPS-r3	JPEG	RS	NRP
PGD	N/A	9.34	5.00	6.00	11.04	8.50	11.96
	SGM	16.80	7.50	9.44	13.96	10.50	12.76
	LinBP	16.80	7.68	10.08	15.76	10.50	13.14
	Ghost	9.60	5.06	6.42	11.92	9.50	12.06
	BPA	**23.96**	**12.02**	**15.60**	**22.52**	**14.00**	**14.08**
MI-FGSM	N/A	16.64	8.04	9.92	16.68	13.00	13.32
	SGM	24.80	11.02	13.16	20.26	14.00	14.38
	LinBP	21.98	10.32	13.26	20.56	12.50	13.22
	Ghost	17.98	8.88	10.64	18.52	13.50	13.84
	BPA	**34.30**	**17.84**	**22.04**	**30.86**	**17.50**	**15.96**

Table 3: Untargeted attack success rates (%) of several attacks on six defenses when generating the adversarial examples on ResNet-50 w/wo various model-related methods.

Attacker	ReLU	Max-pooling	Inc-v3	IncRes-v2	DenseNet	MobileNet	PNASNet	SENet	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
PGD	✗	✗	16.34	13.38	36.86	36.12	13.46	17.40	10.24	9.46	5.52
	✓	✗	29.38	24.00	62.80	61.82	24.98	34.96	17.52	14.38	8.90
	✗	✓	20.26	16.16	44.66	42.82	17.12	21.52	13.20	11.88	7.74
	✓	✓	**35.36**	**30.12**	**70.70**	**68.90**	**32.52**	**42.02**	**22.72**	**19.28**	**12.40**
MI-FGSM	✗	✗	26.20	21.50	51.50	49.68	22.92	30.12	16.22	14.58	9.00
	✓	✗	41.50	34.42	74.96	74.42	35.96	47.58	23.34	18.22	10.94
	✗	✓	34.16	29.02	61.38	59.42	32.24	37.32	21.74	19.96	14.70
	✓	✓	**47.58**	**41.22**	**80.54**	**79.40**	**44.70**	**54.28**	**32.06**	**25.98**	**17.46**

Table 5: Untargeted attack success rates (%) of PGD and MI-FGSM when generating adversarial examples on ResNet-50 w/wo modifying the backward propagation of ReLU or max-pooling.

Summary

NEURAL INFORMATION PROCESSING SYSTEMS

- To our knowledge, it is the first work that proposes and empirically validates the detrimental effect of **gradient truncation** on adversarial transferability. This finding sheds new light on improving adversarial transferability and provides new directions to boost model robustness.
- We propose a model-related attack called **BPA** to mitigate the negative impact of gradient truncation and enhance the relevance of gradient between the loss function and the input.
- Extensive experiments on ImageNet dataset demonstrate that BPA could significantly boost various untargeted and targeted transfer-based attacks.

BPA

TransferAttack

Thanks!

Xiaosen Wang[1*], Kangheng Tong[2*], Kun He[2†]

[1] Singular Security Lab

[2] School of Computer Science and Technology, Huazhong University of Science and Technology

Contact: xiaosen@hust.edu.cn

Homepage: http://xiaosenwang.com/

11/2/2023