Paper Reading —

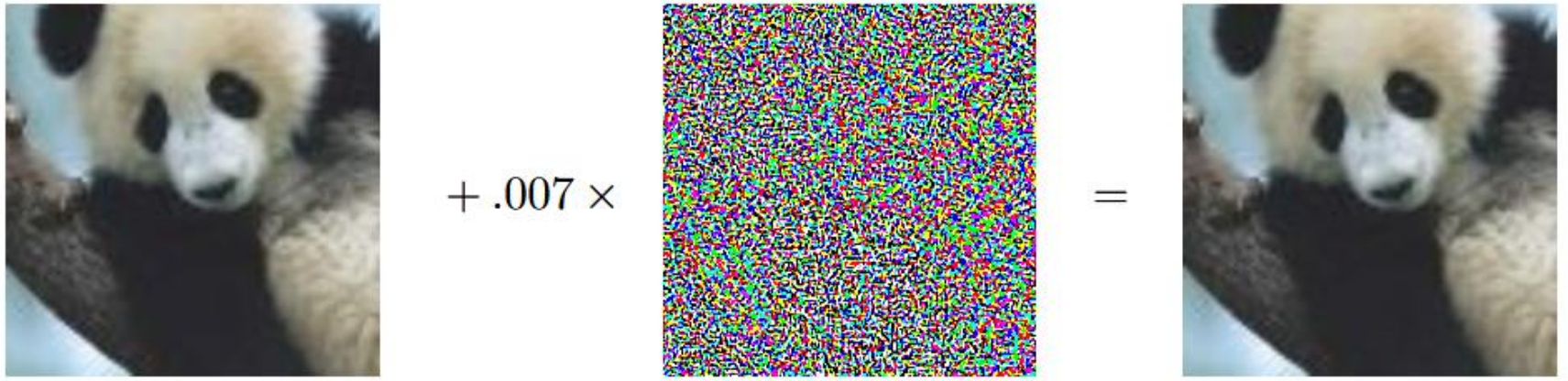
Adversarial Attack

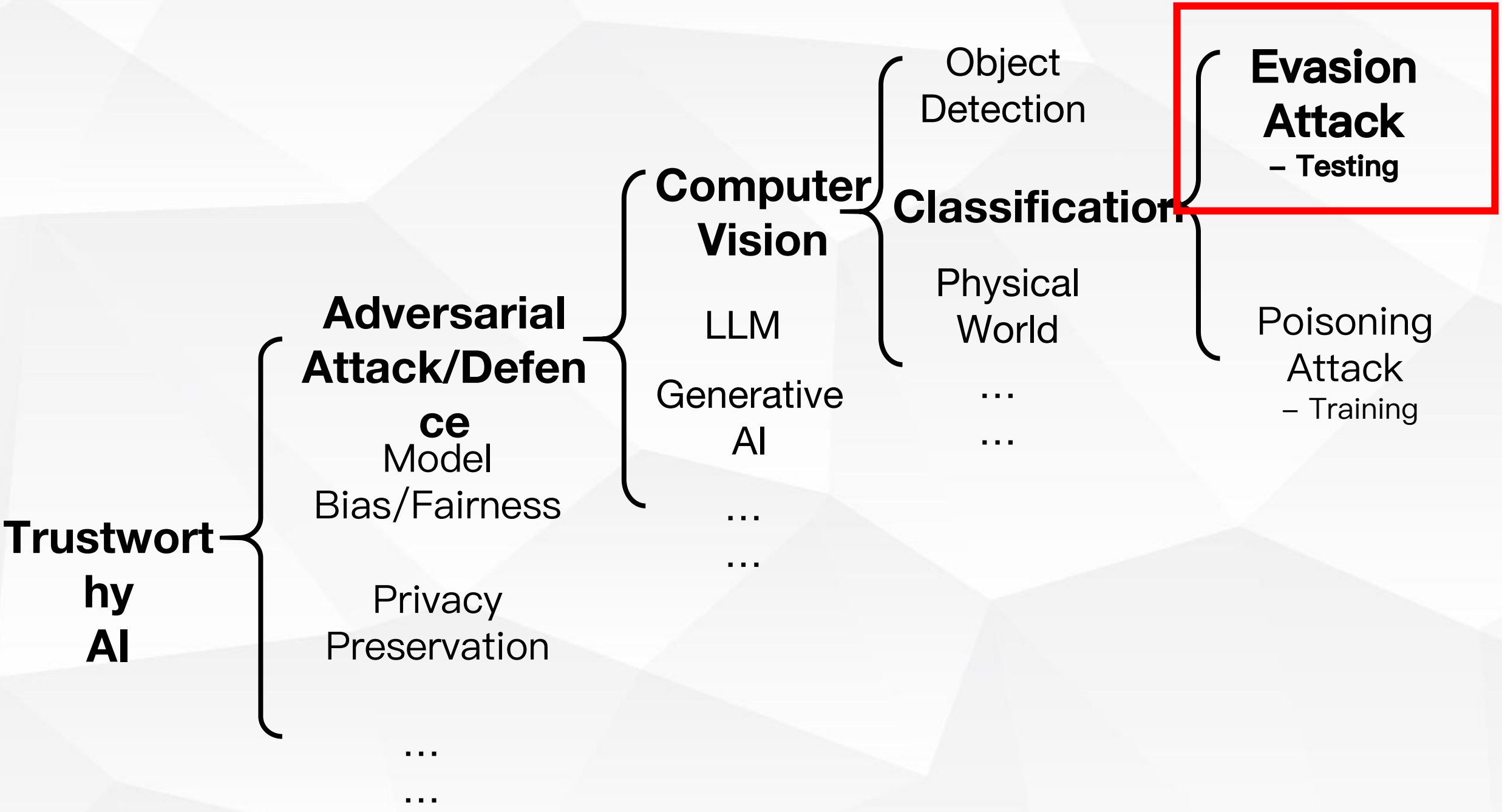
王英硕 - 2025.04.02

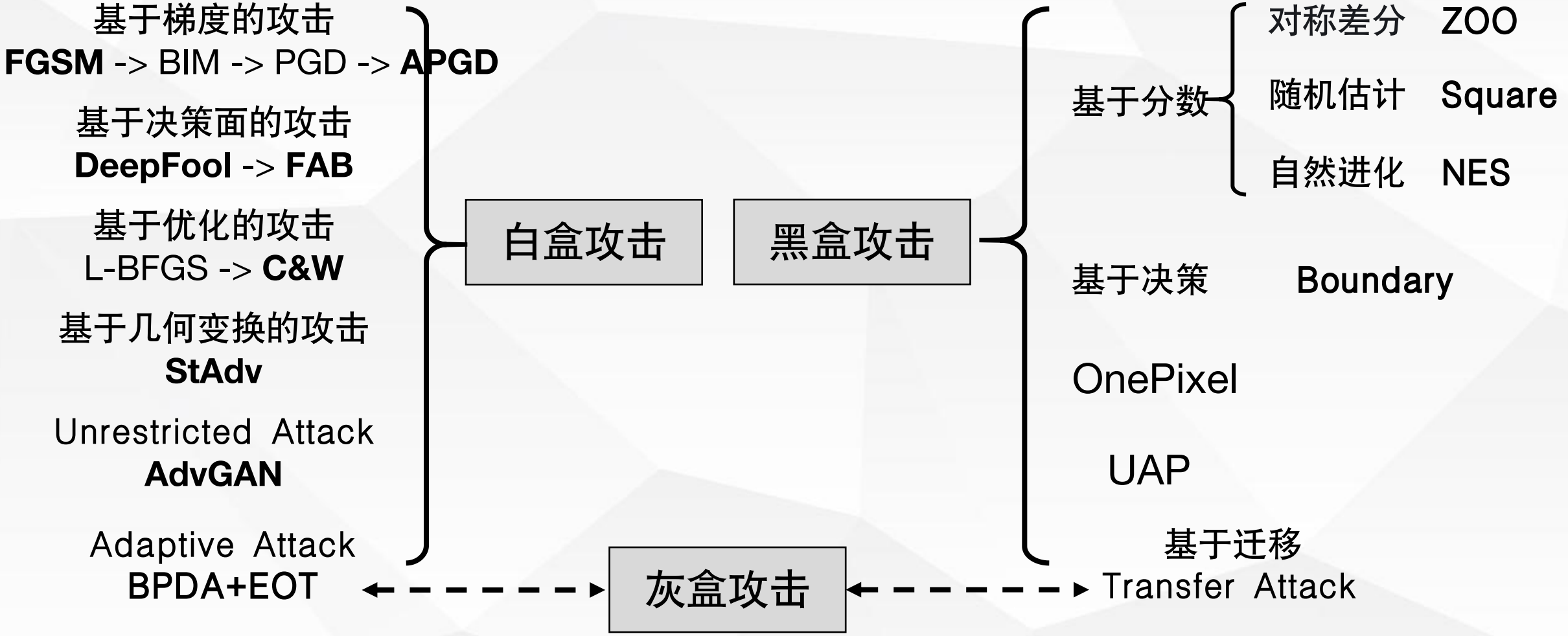
问题定义 – 对抗攻击

- 对抗样本：$x' = x + \delta, s.t., f(x') \neq y$

- 对抗扰动：$\delta = \arg\max_{\delta} \mathcal{L}(f(x+\delta), y), \|\delta\|_p \leq \epsilon$




Trustworthy AI
Adversarial Attack/Defence
Model Bias/Fairness
Privacy Preservation
…
…
Computer Vision
LLM
Generative AI
…
…
Object Detection
Classification
Physical World
…
…
Evasion Attack
– Testing
Poisoning Attack
– Training



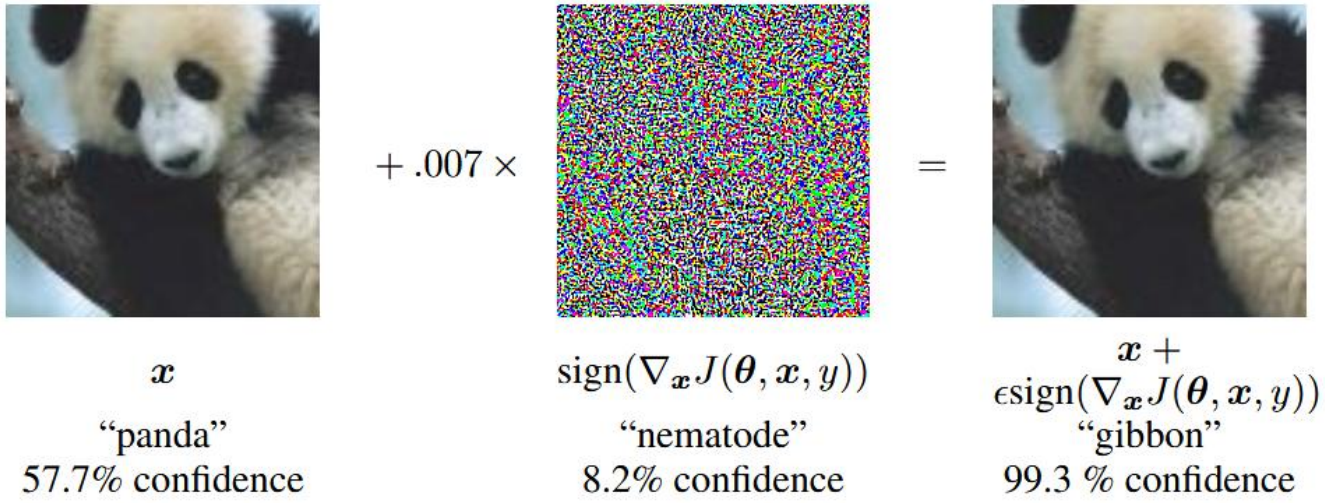
灰盒攻击：攻击者介于知晓“全部参数”和“只有输出信息”之间

白盒攻击　　黑盒攻击
logits / hard label

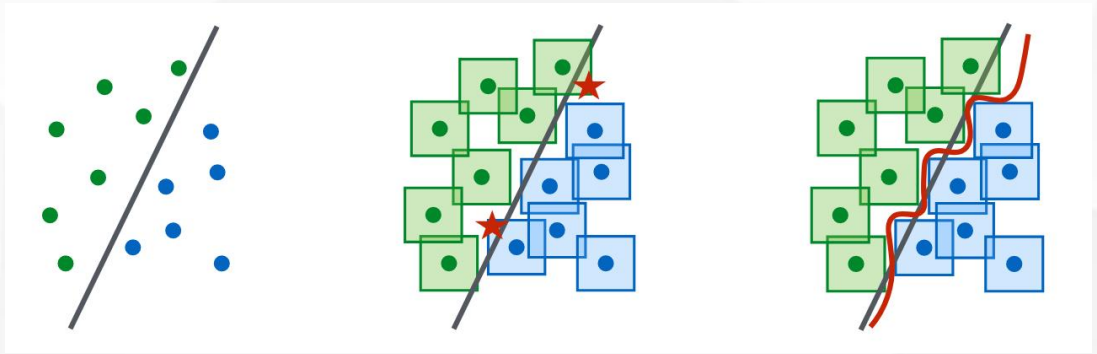
白盒攻击 White-box

FGSM - Fast Gradient Sign Method

基于梯度的攻击



$$x' = x + \eta \cdot sign(\nabla_x J(\theta, x, y))$$



All gradient - based attack methods are based on increasing the loss.

Why do adversarial examples exist?

- **Liner Nature -** Minor perturbations accumulate and amplify in a linear space.

- The model is designed to be non - linear, but in the actual high - dimensional space, it is not.

Goodfellow et al., 2014, Explaining and Harnessing Adversarial

From FGSM to Auto-PGD - Projected Gradient Descent

1. Multiple iterations: $x_{t+1} = \prod_{x+S}(x_t + \eta \cdot sign(\nabla_x J(\theta, x_t, y))$

- For a non - linear model, the direction in just one iteration is not necessarily correct.

2. Adaptive step size: $x^{(k+1)} = \prod_{x+S}(x^{(k)} + \eta^{(k)} \cdot sign(\nabla_x J(\theta, x_t, y))$

- When the value of the objective function grows fast enough, the current step size remains unchanged; otherwise, the step size is halved.

3. Momentum: $x^{(k+1)} = \prod_{x+S}(x^{(k)} + \alpha \cdot (z^{(k+1)} - x^{(k)}) + (1-\alpha) \cdot (x^{(k)} - x^{(k-1)}))$

- The perturbation in each round is related not only to the current gradient direction but also to the previously calculated gradient directions.

Madry et al., 2017, Towards Deep Learning Models Resistant to Adversarial Attacks
Francesco et al., 2020, Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-

From FGSM to Auto-PGD - Projected Gradient Descent

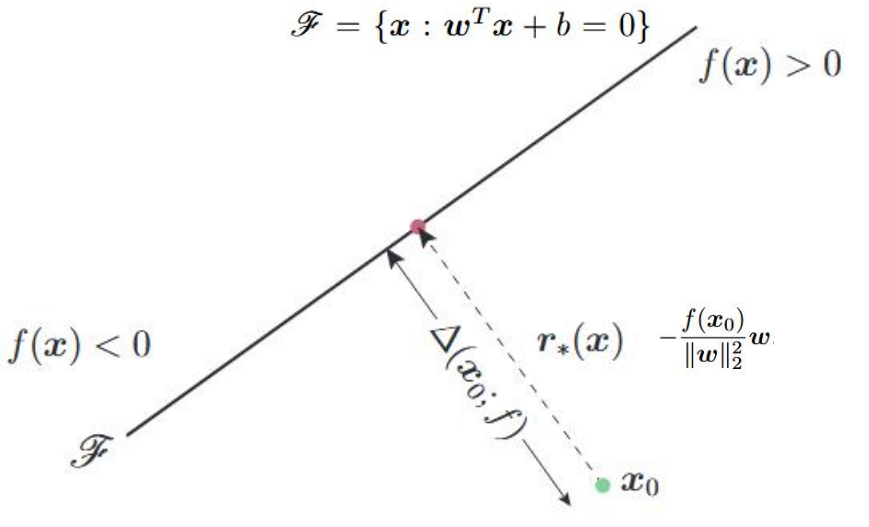
基于梯度的攻击

#	paper	clean	PGD			PGD with Momentum			APGD
			$\epsilon/10$	$\epsilon/4$	2ϵ	$\epsilon/10$	$\epsilon/4$	2ϵ	
CIFAR-10 - $\epsilon = 8/255$									
1	(Carmon et al., 2019)	89.69	60.73	60.72	62.20	60.73	60.69	61.02	**60.48**
2	(Alayrac et al., 2019)	86.46	61.78	61.85	63.23	61.69	61.61	62.10	**61.13**
3	(Hendrycks et al., 2019)	87.11	56.44	56.42	57.15	56.45	56.41	56.39	**56.20**
4	(Rice et al., 2020)	85.34	55.24	55.21	56.04	55.23	55.19	55.26	**55.07**
5	(Qin et al., 2019)	86.28	55.10	55.10	55.95	55.10	55.11	55.22	**54.99**
6	(Engstrom et al., 2019)	87.03	52.24	52.17	53.61	52.26	52.16	52.36	**51.94**
7	(Kumari et al., 2019)	87.80	51.13	51.07	52.67	51.14	51.03	51.22	**50.88**
8	(Mao et al., 2019)	86.21	49.88	49.84	51.18	49.88	49.84	50.02	**49.71**
9	(Zhang et al., 2019a)	87.20	47.13	47.09	48.27	47.11	47.06	47.08	**46.89**
10	(Madry et al., 2018)	87.14	45.98	45.82	47.47	45.95	45.77	46.11	**45.67**
11	(Pang et al., 2020)	80.89	44.40	44.41	45.83	44.41	44.36	44.66	**44.11**
12	(Wong et al., 2020)	83.34	46.04	45.94	47.72	46.03	45.93	46.26	**45.66**
13	(Shafahi et al., 2019)	86.11	44.89	45.93	49.27	44.16	45.10	47.43	**43.90**
14	(Ding et al., 2020)	84.36	51.60	51.44	51.09	51.59	51.29	50.63	**50.25**
15	(Moosavi-Dezfooli et al., 2019)	83.11	40.21	40.18	40.93	40.20	40.16	40.22	**40.10**
16	(Zhang & Wang, 2019)	89.98	55.98	55.30	**48.59**	55.99	55.27	50.91	51.65
17	(Zhang & Xu, 2020)	90.25	67.00	66.61	**57.87**	67.02	66.67	64.19	64.46
18	(Jang et al., 2019)	78.91	36.85	36.81	39.32	36.85	36.83	37.30	**36.52**
19	(Kim & Wang, 2020)	91.51	56.85	55.90	**50.26**	56.94	55.92	52.54	53.04
20	(Moosavi-Dezfooli et al., 2019)	80.41	35.38	35.37	36.86	35.38	35.35	35.48	**35.29**
21	(Wang & Zhang, 2019)	92.80	59.93	58.42	**52.56**	60.01	58.46	54.51	54.77
22	(Wang & Zhang, 2019)	92.82	66.49	65.86	**56.71**	66.60	65.76	63.53	63.80
23	(Mustafa et al., 2019)	89.16	18.27	14.75	**4.42**	18.44	14.66	5.46	4.73
24	(Chan et al., 2020)	93.79	1.32	1.28	5.51	1.30	1.29	1.29	**1.23**
25	(Pang et al., 2020)	93.52	1.88	0.54	1.42	1.59	0.53	0.47	**0.11**

Madry et al., 2017, Towards Deep Learning Models Resistant to Adversarial Attacks
Francesco et al., 2020, Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-

DeepFool

- **Decision boundary:** A hyperplane that completely separates data points of different classes.
- **Robustness:** The minimum perturbation with the minimum distance cost to the hyperplane.

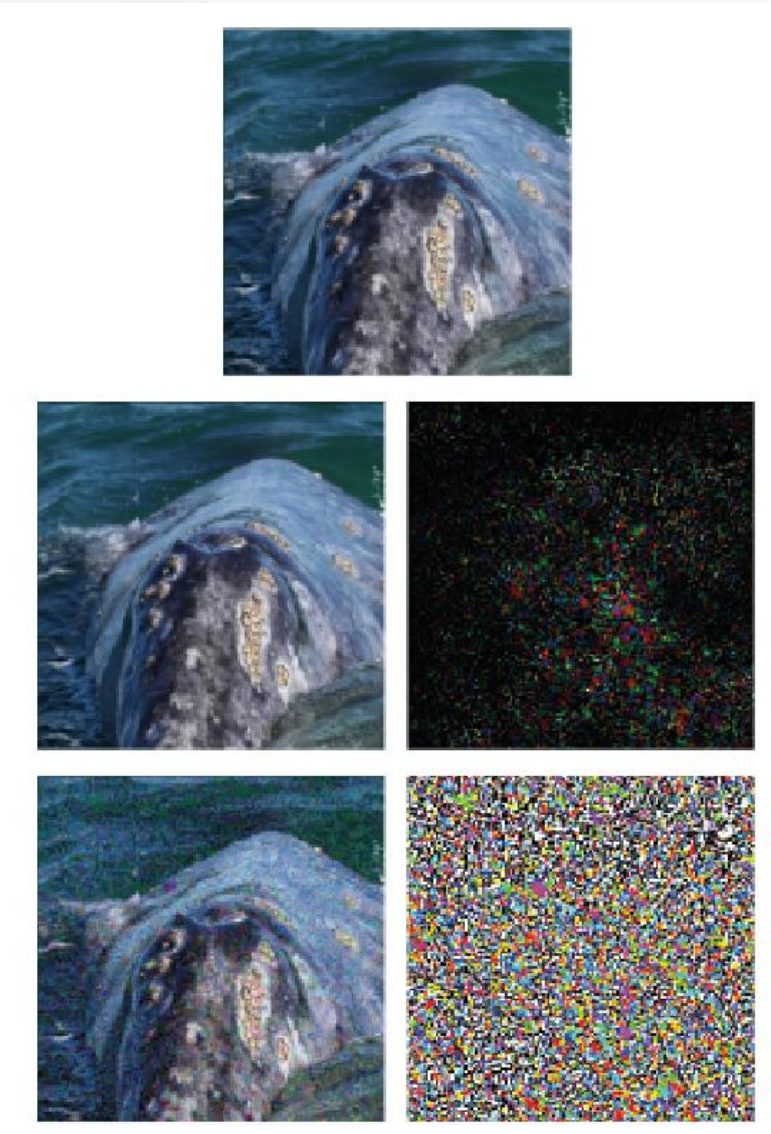


while $\text{sign}(f(\boldsymbol{x}_i)) = \text{sign}(f(\boldsymbol{x}_0))$ **do**

$$\boldsymbol{r}_i \leftarrow -\frac{f(\boldsymbol{x}_i)}{\|\nabla f(\boldsymbol{x}_i)\|_2^2} \nabla f(\boldsymbol{x}_i),$$

$$\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \boldsymbol{r}_i,$$

$$i \leftarrow i + 1.$$



- **Untargeted Attack:** Seek the point closest to the decision boundary in the high - dimensional space to be used as the label after the attack.

Moosavi et al., 2016, DeepFool: a simple and accurate method to fool deep neural

From DeepFool to FAB - Fast Adaptive Boundary Attack

1. Precise projection calculation: the projection onto the hyperplane under box constraints.

2. Introduction of bias: The iterative process is more biased towards the original point.

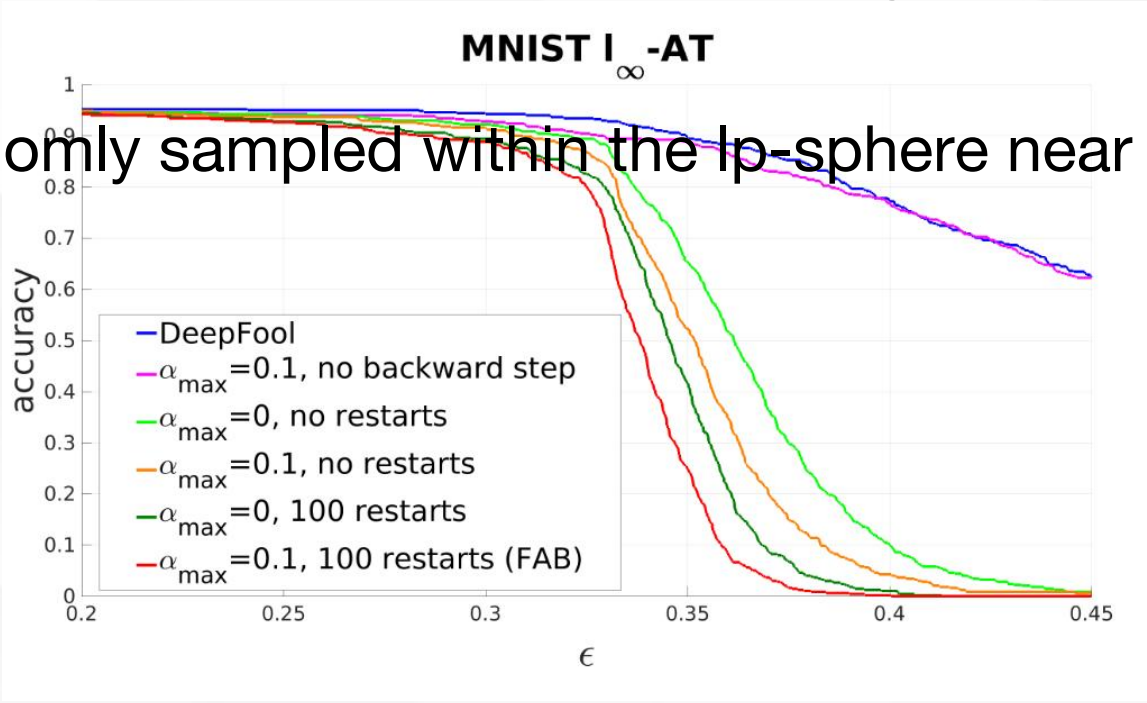
3. Backward step iteration: Adjust the current sample to approach the original point and restart the algorithm.

4. [illegible] is randomly sampled within the lp-sphere near the

$$\pi_l(z) : f_l(x^{(i)}) - f_c(x^{(i)}) + \left\langle \nabla f_l(x^{(i)}) - \nabla f_c(x^{(i)}), z - x^{(i)} \right\rangle = 0.$$

$$x^{(i+1)} = (1-\alpha)\,\mathrm{proj}_p(x^{(i)}, \pi_s, C) + \alpha\,\mathrm{proj}_p(x_{\mathrm{orig}}, \pi_s, C),$$

$$\alpha = \min\left\{ \frac{\left\|\delta^{(i)}\right\|_p}{\left\|\delta^{(i)}\right\|_p + \left\|\delta^{(i)}_{\mathrm{orig}}\right\|_p}, \alpha_{\max} \right\} \in [0,1].$$



Francesco et al., 2020, Minimally distorted Adversarial Examples with a Fast Adaptive

C&W - Carlini and Wagner

基于优化的攻击

$$\begin{aligned}&\text{minimize} \quad \mathcal{D}(x, x+\delta)\\ &\text{such that} \quad C(x+\delta) = t\\ &\qquad\quad x+\delta \in [0,1]^n\end{aligned}$$

$$\begin{aligned}&\text{minimize} \quad \mathcal{D}(x, x+\delta)\\ &\text{such that} \quad f(x+\delta) \leq 0\\ &\qquad\quad x+\delta \in [0,1]^n\end{aligned}$$

$$\begin{aligned}&\text{minimize} \quad \|\delta\|_p + c \cdot f(x+\delta)\\ &\text{such that} \quad x+\delta \in [0,1]^n\end{aligned}$$

$$f_1(x') = -\text{loss}_{F,t}(x') + 1$$
$$f_2(x') = (\max_{i \neq t}(F(x')_i) - F(x')_t)^+$$
$$f_3(x') = \text{softplus}(\max_{i \neq t}(F(x')_i) - F(x')_t) - \log(2)$$
$$f_4(x') = (0.5 - F(x')_t)^+$$
$$f_5(x') = -\log(2F(x')_t - 2)$$
$$f_6(x') = (\max_{i \neq t}(Z(x')_i) - Z(x')_t)^+$$
$$f_7(x') = \text{softplus}(\max_{i \neq t}(Z(x')_i) - Z(x')_t) - \log(2)$$

- 将寻找对抗样本的问题转化为一个优化问题，通过定义合适的距离度量和目标函数，利用优化算法迭代搜索得到最优扰动，进而生成对抗样本。

	Best Case						Average Case						Worst Case					
	Change of Variable		Clipped Descent		Projected Descent		Change of Variable		Clipped Descent		Projected Descent		Change of Variable		Clipped Descent		Projected Descent	
	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob
f_1	2.46	100%	2.93	100%	2.31	100%	4.35	100%	5.21	100%	4.11	100%	7.76	100%	9.48	100%	7.37	100%
f_2	4.55	80%	3.97	83%	3.49	83%	3.22	44%	8.99	63%	15.06	74%	2.93	18%	10.22	40%	18.90	53%
f_3	4.54	77%	4.07	81%	3.76	82%	3.47	44%	9.55	63%	15.84	74%	3.09	17%	11.91	41%	24.01	59%
f_4	5.01	86%	6.52	100%	7.53	100%	4.03	55%	7.49	71%	7.60	71%	3.55	24%	4.25	35%	4.10	35%
f_5	1.97	100%	2.20	100%	1.94	100%	3.58	100%	4.20	100%	3.47	100%	6.42	100%	7.86	100%	6.12	100%
f_6	1.94	100%	2.18	100%	1.95	100%	3.47	100%	4.11	100%	3.41	100%	6.03	100%	7.50	100%	5.89	100%
f_7	1.96	100%	2.21	100%	1.94	100%	3.53	100%	4.14	100%	3.43	100%	6.20	100%	7.57	100%	5.94	100%

$$\delta_i = \frac{1}{2}(\tanh(w_i) + 1) - x_i$$

$$f(\min(\max(x+\delta, 0), 1))$$

Carlini et al., 2017, Towards Evaluating the Robustness of Neural

C&W - Carlini and Wagner

基于优化的攻击

	Best Case				Average Case				Worst Case			
	MNIST		CIFAR		MNIST		CIFAR		MNIST		CIFAR	
	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob
Our L_0	8.5	100%	5.9	100%	16	100%	13	100%	33	100%	24	100%
JSMA-Z	20	100%	20	100%	56	100%	58	100%	180	98%	150	100%
JSMA-F	17	100%	25	100%	45	100%	110	100%	100	100%	240	100%
Our L_2	1.36	100%	0.17	100%	1.76	100%	0.33	100%	2.60	100%	0.51	100%
Deepfool	2.11	100%	0.85	100%	–	-	–	-	–	-	–	-
Our L_∞	0.13	100%	0.0092	100%	0.16	100%	0.013	100%	0.23	100%	0.019	100%
Fast Gradient Sign	0.22	100%	0.015	99%	0.26	42%	0.029	51%	–	0%	0.34	1%
Iterative Gradient Sign	0.14	100%	0.0078	100%	0.19	100%	0.014	100%	0.26	100%	0.023	100%

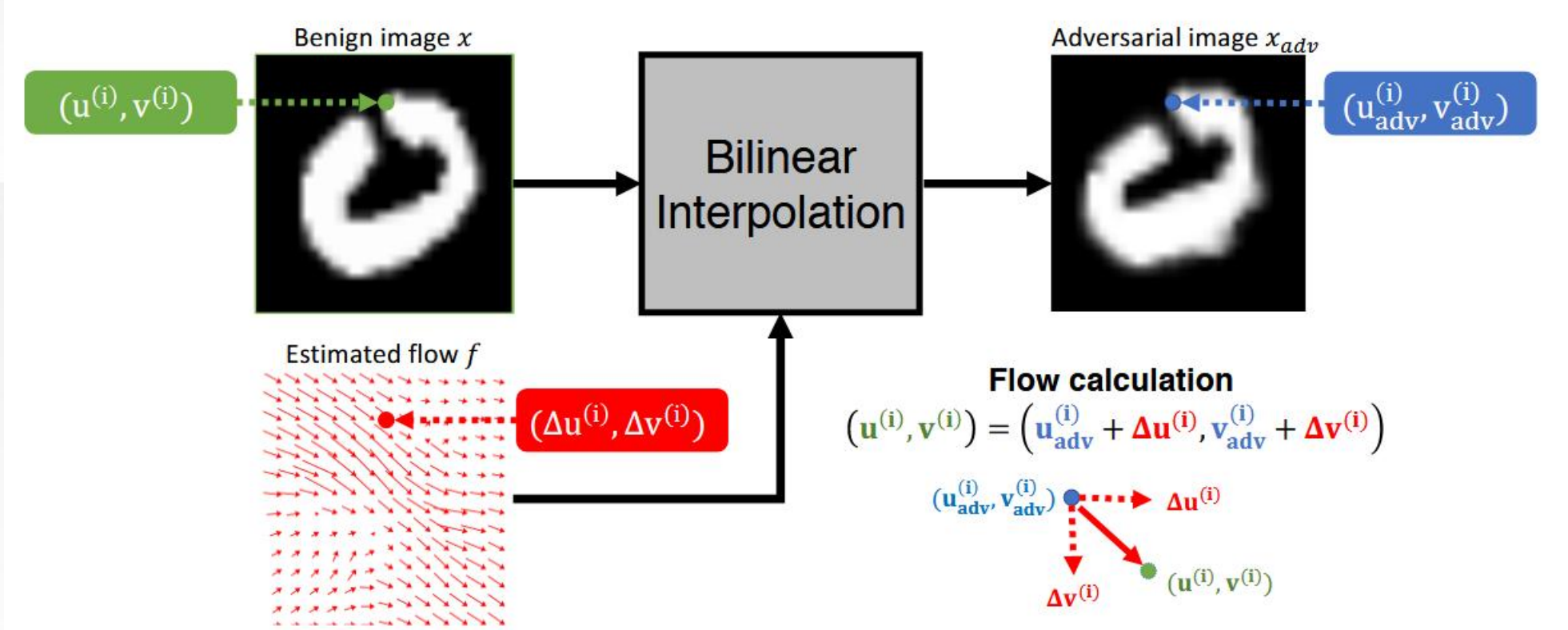
Method	FGSM PGD	Deepfool FAB	C&W
Lp	Linf L2	Linf L2	Linf L2 L0

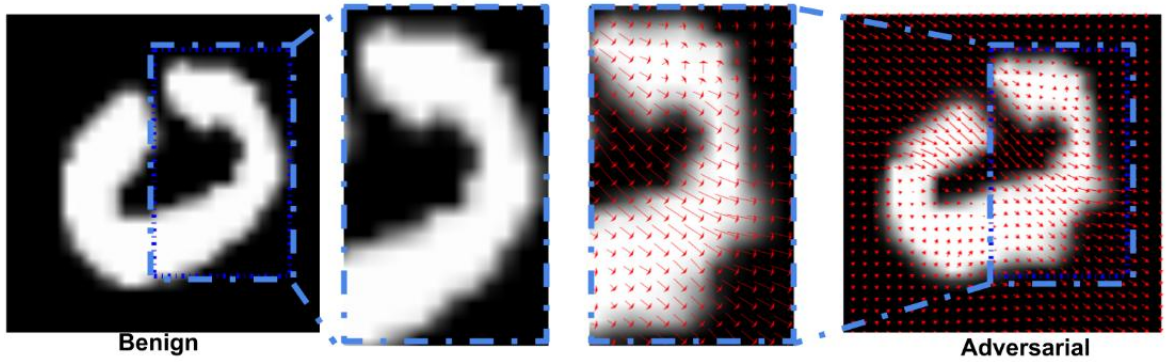
- 这些攻击都是在Lp约束范式下的攻击，但Lp约束不一定是人眼无法分辨的准确描述

Carlini et al., 2017, Towards Evaluating the Robustness of Neural

StAdv - Spatially Transformed Adversarial

基于几何变换的攻击





Xiao et al., 2018, Spatially Transformed Adversarial Examp

StAdv - Spatially Transformed Adversarial

基于几何变换的攻击

- Assume that the original image is x, and the adversarial image is x_{adv}.
- For each pixel in x_{adv}, there is a corresponding coordinate $(u_{adv}^{(i)}, v_{adv}^{(i)})$.
- A flow vector $f_i = (\Delta u^{(i)}, \Delta v^{(i)})$ is used to represent the displacement of each pixel in the horizontal and vertical directions.

$$f^* = \underset{f}{\operatorname{argmin}} \quad \mathcal{L}_{adv}(x, f) + \tau \mathcal{L}_{flow}(f),$$

- Prompt the sample to be misclassified and maximize the difference as much as possible.

$$\mathcal{L}_{adv}(x, f) = \max(\max_{i \neq t} g(\mathbf{x}_{\text{adv}})_i - g(\mathbf{x}_{\text{adv}})_t, \kappa)$$

- Ensure the smoothness of the spatial transformation and reduce the displacement difference between adjacent pixels.

$$\mathcal{L}_{flow}(f) = \sum_{p}^{all\ pixels} \sum_{q \in \mathcal{N}(p)} \sqrt{||\Delta u^{(p)} - \Delta u^{(q)}||_2^2 + ||\Delta v^{(p)} - \Delta v^{(q)}||_2^2}.$$

Xiao et al., 2018, Spatially Transformed Adversarial Examp

StAdv - Spatially Transformed Adversarial

基于几何变换的攻击

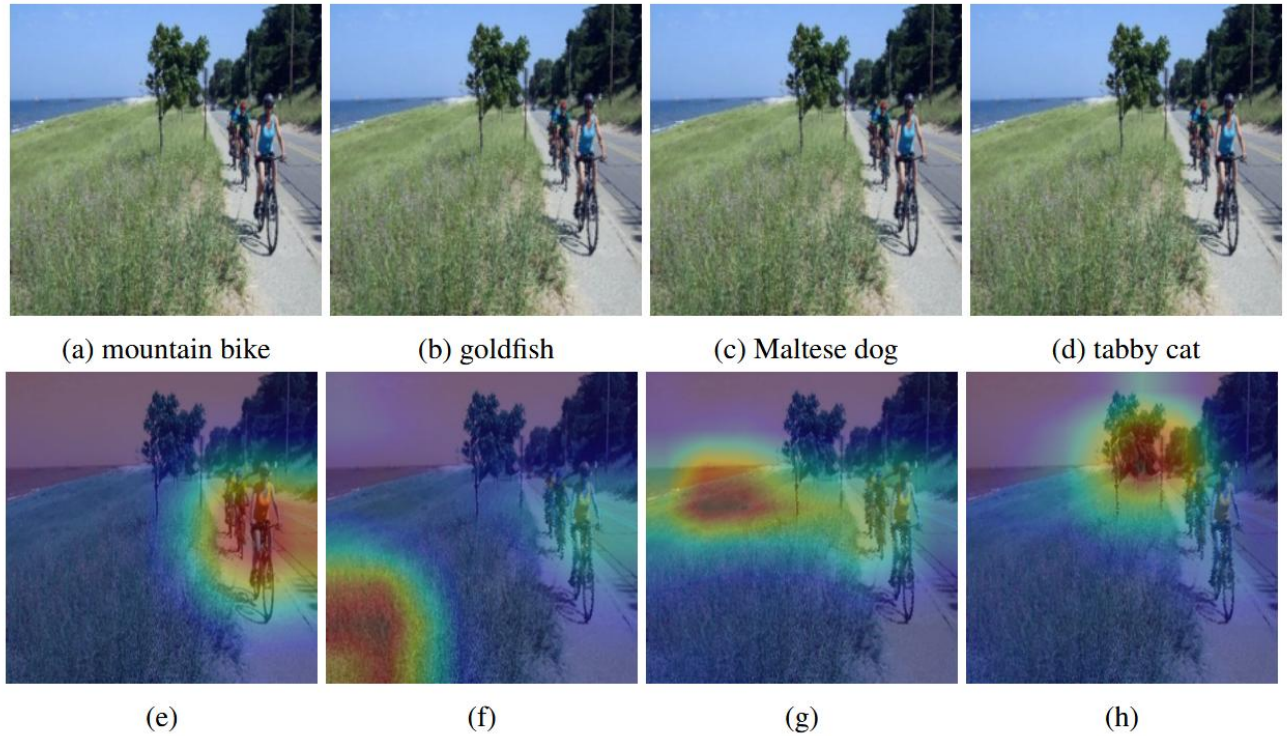


Figure 8: CAM attention visualization for ImageNet inception_v3 model. (a) the original image and (b)-(d) are stAdv adversarial examples targeting different classes. Row 2 shows the attention visualization for the corresponding images above.

Model	Def.	FGSM	C&W.	stAdv
ResNet32	Adv.	13.10%	11.9%	**43.36%**
	Ens.	10.00%	10.3%	**36.89%**
	PGD	22.8%	21.4%	**49.19%**
wide ResNet34	Adv.	5.04%	7.61%	**31.66%**
	Ens.	4.65%	8.43%	**29.56%**
	PGD	14.9%	13.90%	**31.6%**

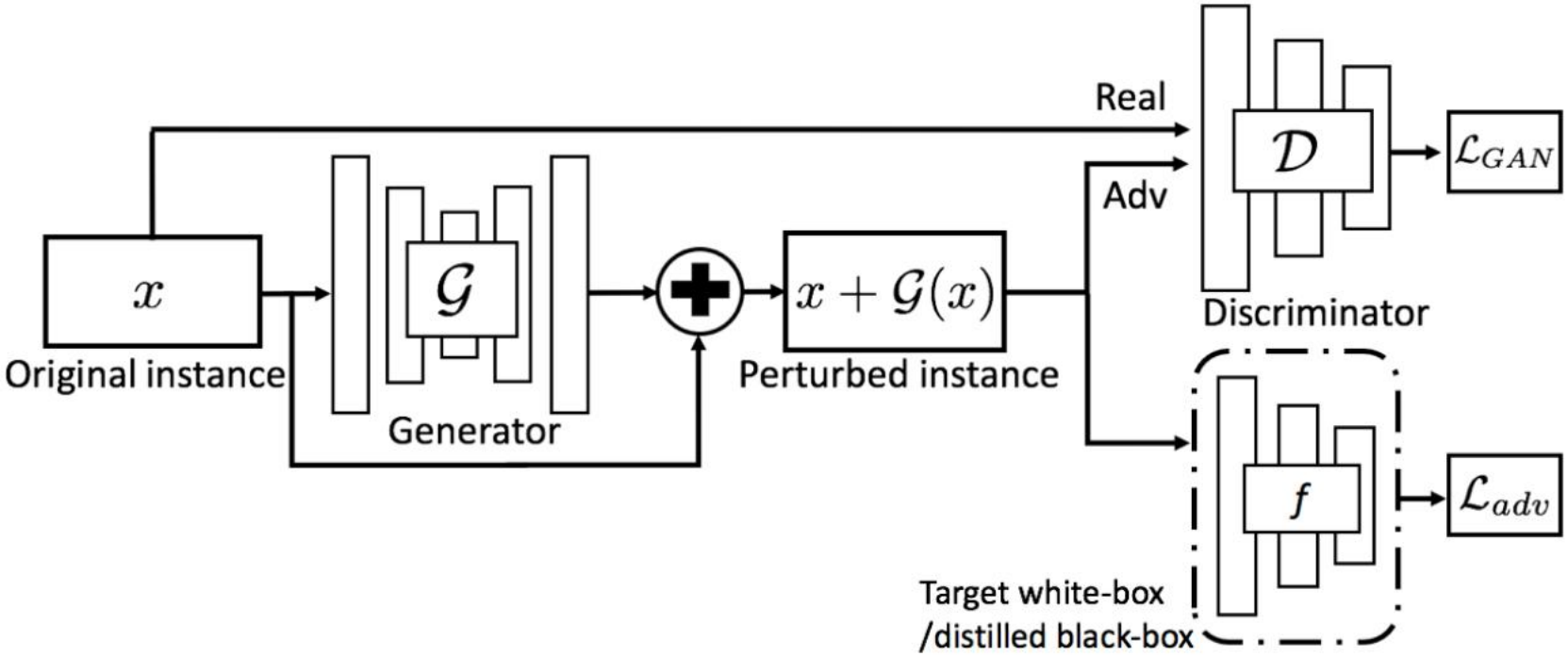
- 基于几何变换的攻击常常用于评估模型鲁棒性的补充，用来测试非Lp范式下模型的鲁棒能力

Xiao et al., 2018, Spatially Transformed Adversarial Examp

AdvGAN – Unrestricted Attack

(a) Strawberry (b) Toy poodle

(c) Buckeye (d) Toy poodle



$$\mathcal{L} = \mathcal{L}_{adv}^{f} + \alpha \mathcal{L}_{GAN} + \beta \mathcal{L}_{hinge}$$

- It is sometimes argued that the perturbation norm is not a good indicator of the perceptual difference between the two images.

Xiao et al., 2018, Generating Adversarial Examples with Adversarial

AdvGAN – Unrestricted Attack

$$\mathcal{L} = \mathcal{L}_{adv}^{f} + \alpha \mathcal{L}_{GAN} + \beta \mathcal{L}_{hinge}$$

- Encourage the generated instances to be similar to the data from the original class.

$$\mathcal{L}_{GAN} = \mathbb{E}_x log \mathcal{D}(x) + \mathbb{E}_x log(1 - \mathcal{D}(x + \mathcal{G}(x)))$$

- Prompt the perturbed image to be misclassified as the target class t.

$$\mathcal{L}_{adv}^{f} = \mathbb{E}_x \ell_f(x + \mathcal{G}(x), t)$$

- Limit the magnitude of the perturbation and stabilize the GAN training.

$$\mathcal{L}_{hinge} = \mathbb{E}_x max\left(0, \|\mathcal{G}(x)\|_2 - c\right)$$

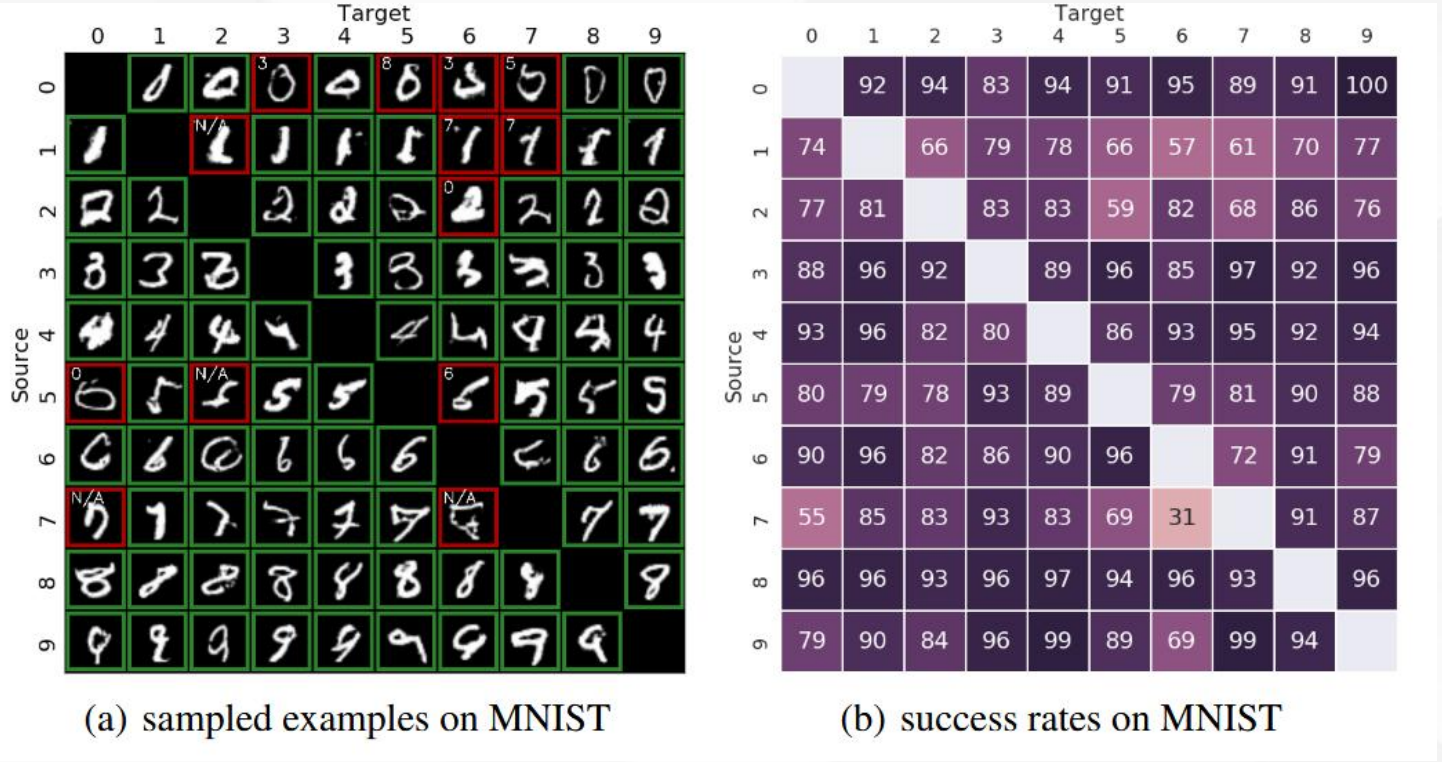
- **Semi-whitebox attack:** After the generator G is trained, adversarial perturbations can be generated for any input instance without accessing the target model.

- **Black-box attack:** Static distillation (Transfer-based), dynamic distillation (with better performance)

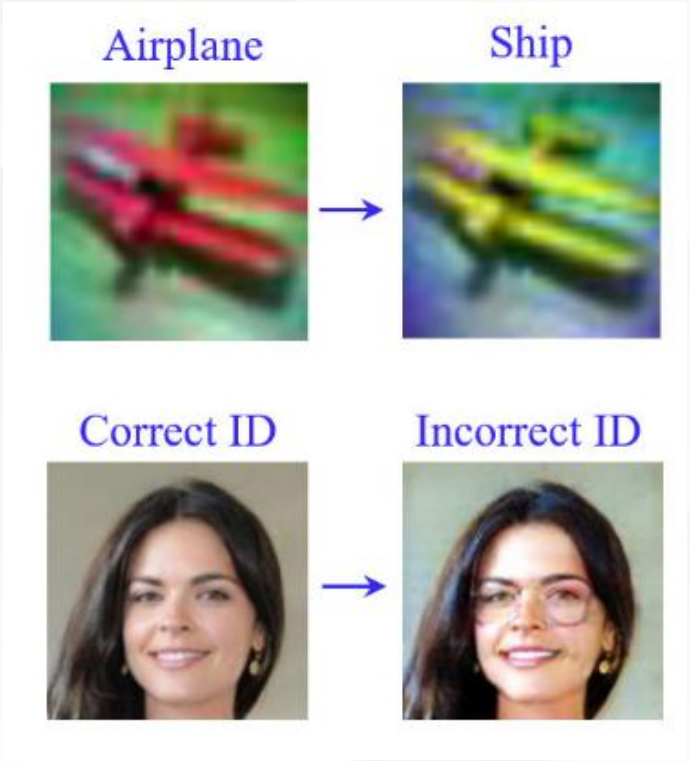
	FGSM	Opt.	Trans.	AdvGAN
Run time	0.06s	>3h	-	<0.01s
Targeted Attack	✓	✓	Ens.	✓
Black-box Attack			✓	✓

Xiao et al., 2018, Generating Adversarial Examples with Adversarial

Unrestricted Attack



(a) sampled examples on MNIST (b) success rates on MNIST



- 利用生成模型，使生成的对抗样本在保证高感知质量的同时能有效误导目标模型，且不受传统攻击方式中生成样本方式的限制，直接生成全新的内容

Song et al., 2018, Constructing Unrestricted Adversarial Examples with

黑盒攻击 Black-box

ZOO - Zeroth Order Optimization

基于分数的攻击 – 梯度估计

C&W

$$\begin{aligned} &\text{minimize} && \|\delta\|_p + c \cdot f(x+\delta) \\ &\text{such that} && x+\delta \in [0,1]^n \end{aligned}$$

$$f(\mathbf{x}, t) = \max\{\max_{i \neq t}[Z(\mathbf{x})]_i - [Z(\mathbf{x})]_t, -\kappa\},$$

1. The **logit** layer representation is an internal state information of a DNN;

2. **Back propagation** on the targeted DNN is required for solving.

ZOO

$$\hat{g}_i := \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h},$$

$$f(\mathbf{x}, t) = \max\{\max_{i \neq t} \log[F(\mathbf{x})]_i - \log[F(\mathbf{x})]_t, -\kappa\},$$

1. Modify the loss function f (x, t) such that it only depends on the **output** F of a DNN and the desirec class label t;

2. Compute an approximate gradient using a finite difference method, and solve the optimization problem **via zeroth order optimization**.

Compute an update δ^* by approximately minimizing

$$\arg\min_{\delta} f(\mathbf{x} + \delta \mathbf{e}_i)$$

Update $\mathbf{x}_i \leftarrow \mathbf{x}_i + \delta^*$

Chen et al., 2017, ZOO: Zeroth Order Optimization based Black-box Attacks to Deep Neural Networks without Training Substitute Models

ZOO - Zeroth Order Optimization

Compute an update δ^* by approximately minimizing

$$\arg\min_{\delta} f(\mathbf{x} + \delta \mathbf{e}_i)$$

Update $\mathbf{x}_i \leftarrow \mathbf{x}_i + \delta^*$

Acceleration methods:

- Attack-space dimension reduction
- Hierarchical attack
- Optimize the important pixels

	MNIST					
	Untargeted			Targeted		
	Success Rate	Avg. L_2	Avg. Time (per attack)	Success Rate	Avg. L_2	Avg. Time (per attack)
White-box (C&W)	100 %	1.48066	0.48 min	100 %	2.00661	0.53 min
Black-box (Substitute Model + FGSM)	40.6 %	-	0.002 sec (+ 6.16 min)	7.48 %	-	0.002 sec (+ 6.16 min)
Black-box (Substitute Model + C&W)	33.3 %	3.6111	0.76 min (+ 6.16 min)	26.74 %	5.272	0.80 min (+ 6.16 min)
Proposed black-box (ZOO-ADAM)	100 %	1.49550	1.38 min	98.9 %	1.987068	1.62 min
Proposed black-box (ZOO-Newton)	100 %	1.51502	2.75 min	98.9 %	2.057264	2.06 min
	CIFAR10					
	Untargeted			Targeted		
	Success Rate	Avg. L_2	Avg. Time (per attack)	Success Rate	Avg. L_2	Avg. Time (per attack)
White-box (C&W)	100 %	0.17980	0.20 min	100 %	0.37974	0.16 min
Black-box (Substitute Model + FGSM)	76.1 %	-	0.005 sec (+ 7.81 min)	11.48 %	-	0.005 sec (+ 7.81 min)
Black-box (Substitute Model + C&W)	25.3 %	2.9708	0.47 min (+ 7.81 min)	5.3 %	5.7439	0.49 min (+ 7.81 min)
Proposed Black-box (ZOO-ADAM)	100 %	0.19973	3.43 min	96.8 %	0.39879	3.95 min
Proposed Black-box (ZOO-Newton)	100 %	0.23554	4.41 min	97.0 %	0.54226	4.40 min

- 基于梯度估计 + 优化的方法查询效率很低，花费的query和时间资源大

Chen et al., 2017, ZOO: Zeroth Order Optimization based Black-box Attacks to Deep Neural Networks without Training Substitute Models

NES - Natural Evolutionary Strategies

$\theta = x + \sigma\delta \quad \delta \sim N(0, I)$

- **Search distribution:** Random Gaussian noise around the current image x.
- Generate n search points in the sear $\nabla\mathbb{E}[F(\theta)] \approx \frac{1}{\sigma n}\sum_{i=1}^{n}\delta_i F(\theta + \sigma\delta_i)$ **stimate the gradient** using points with high fitness.
- Use PGD for attack.

Partial-Information – top-k probabilities

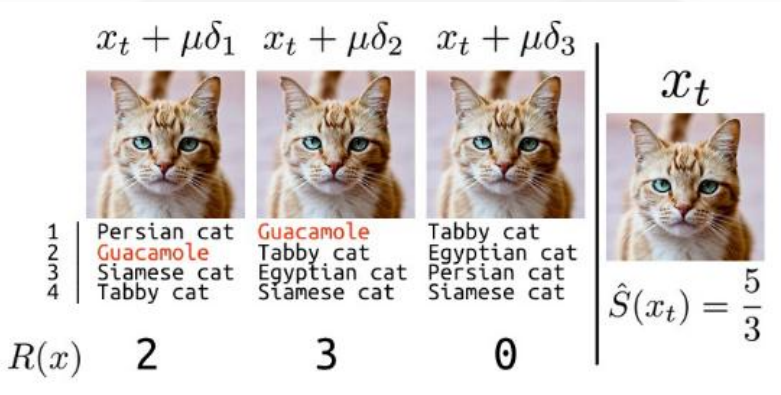
- Start the attack from an image of the **target class**.
- Project onto Linf boxes of decreasing sizes centered at the original image, ensuring that the adversarial class **remains within the top-k classes** at all times.
- Perturb the image.

Label-Only – top-k labels

A proxy for the softmax probabilit

$$R(x^{(t)}) = k - rank(y_{adv}|x^{(t)})$$

$$\hat{S}(x^{(t)}) = \frac{1}{n}\sum_{i=1}^{n} R(x^{(t)} + \mu\delta_i)$$



Threat model	Success rate	Median queries
QL	99.2%	11,550
PI	93.6%	49,624
LO	90%	2.7×10^6

Ilyas et al., 2018, Black-box Adversarial Attacks with Limited Queries and

Square Attack

AutoAttack = APGD + FAB + Square

- 目前最强大且常用的攻击评测方式

基于分数的攻击 – 随机查找

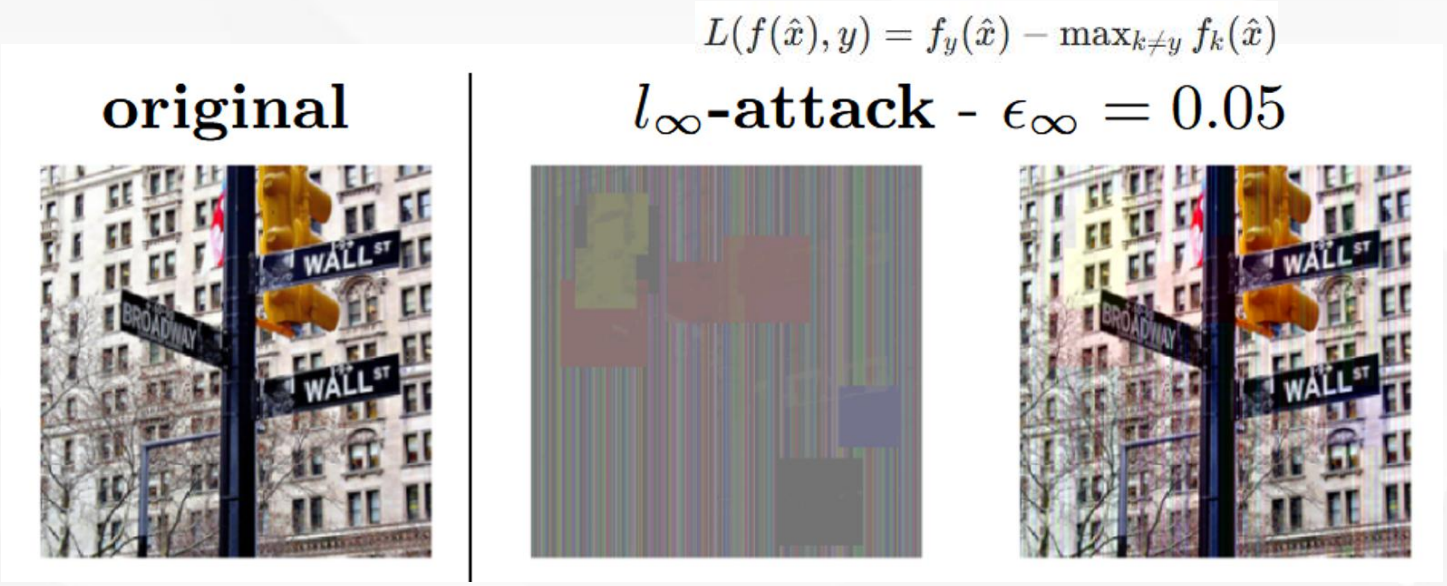
Linf- Square Attack:

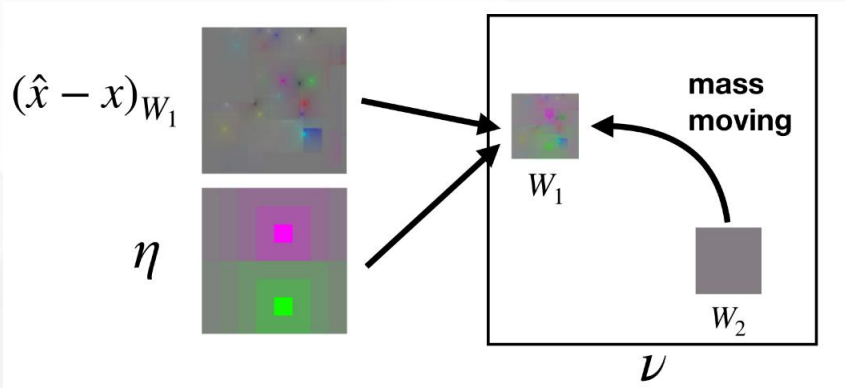
- Initialization: Use vertical stripe perturbations with a width of 1 and values in (–ε, ε), as convolutional networks are sensitive to high-frequency perturbations.
- Randomly select a square, and the perturbation ranges from -2ε to 2ε.

L2- Square Attack:

- Mass moving: Select two regions of the same size, set the perturbation of one region to zero, and use its perturbation budget to increase the perturbation norm of the other region.

$$L(f(\hat{x}), y) = f_y(\hat{x}) - \max_{k \neq y} f_k(\hat{x})$$

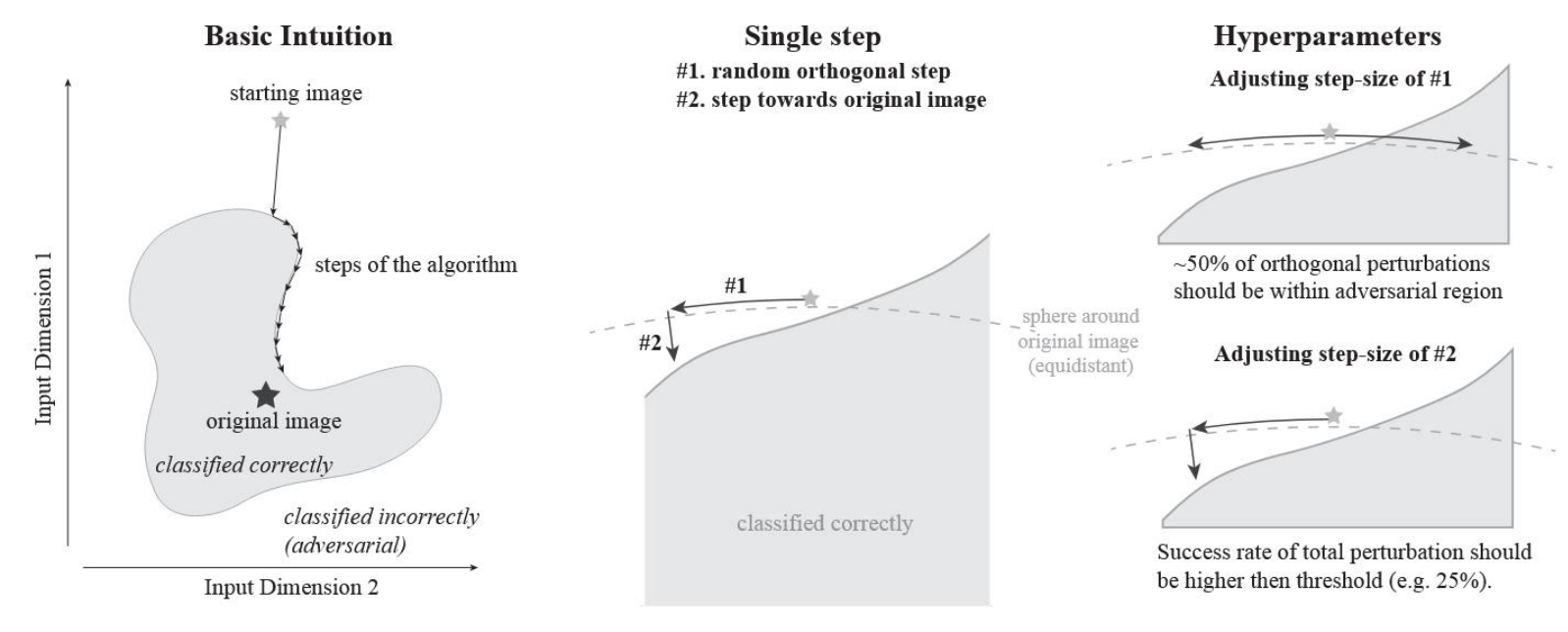




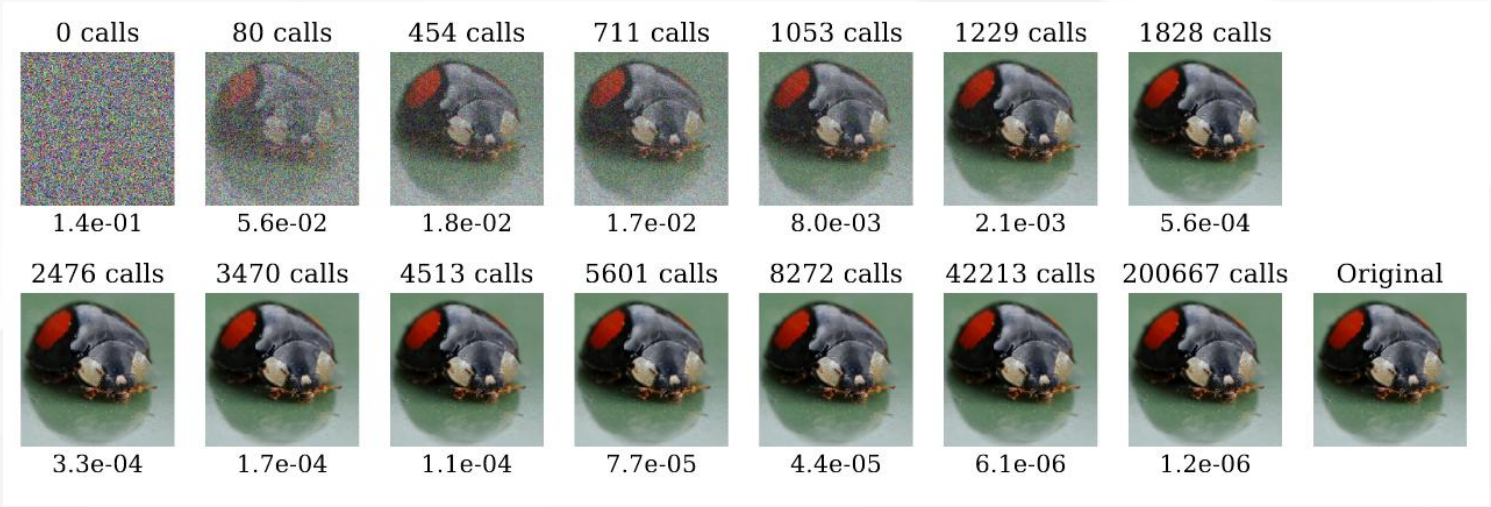
Andriushchenko et al., 2019, Square Attack: a query-efficient black-box adversarial attack via rando

Boundary Attack

基于决策的攻击 – 随机查找



- 从一个已经是对抗的点开始，通过在对抗和非对抗区域的**边界**进行**随机游走**，在保持对抗性的同时减小与目标图像的距离。



1. Sample the perturbation vector for each dimension from the Gaussian distribution.

2. Project the perturbation vector η so that it lies on the sphere centered at the original image.

3. Make a small move η towards the original image.

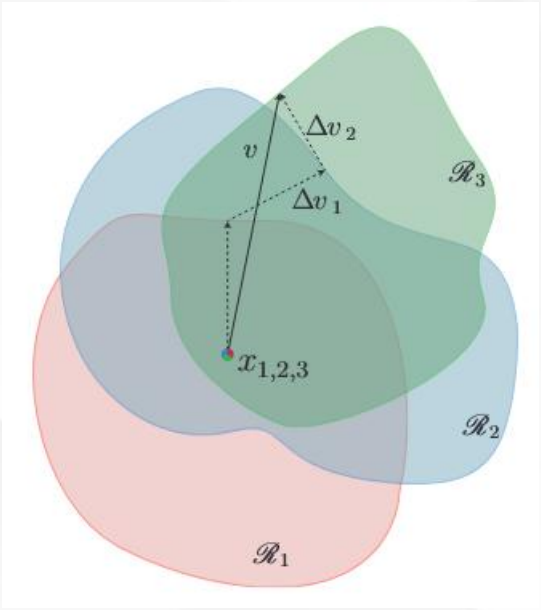
Brendel et al., 2017, Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine

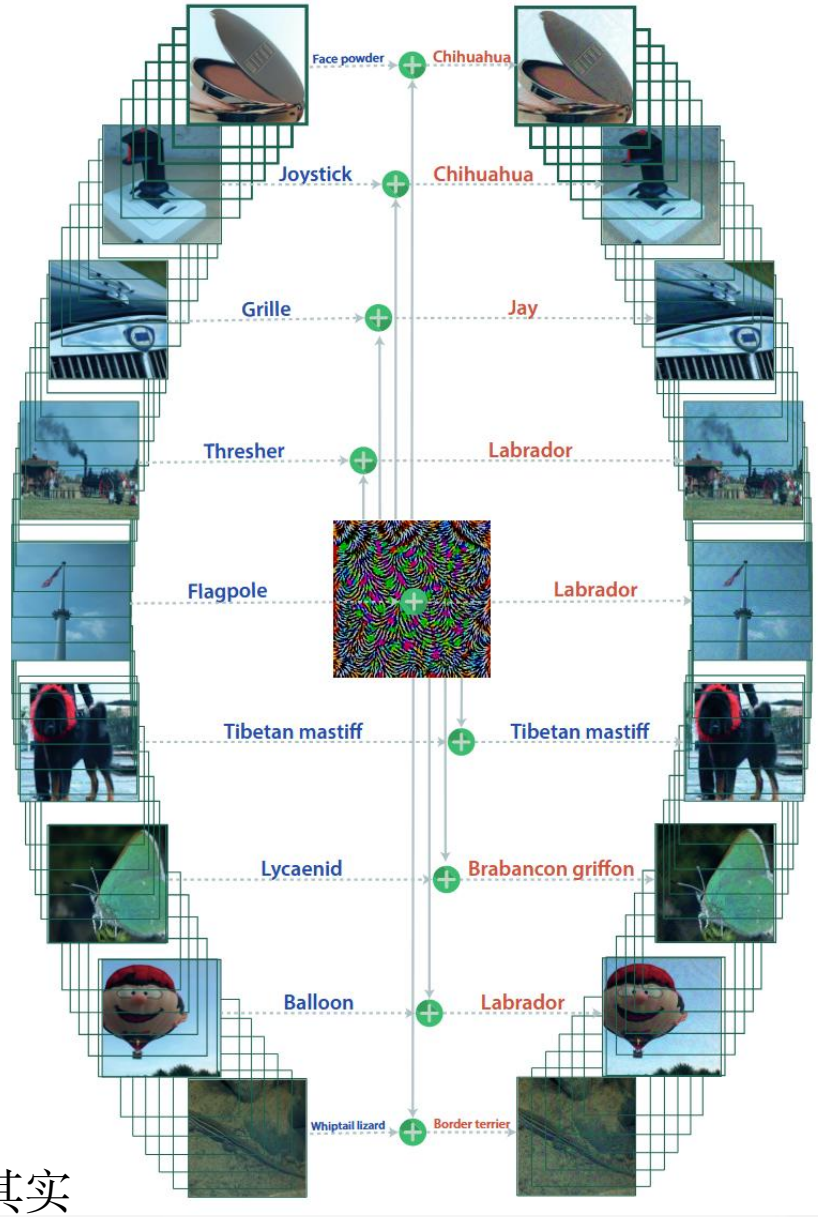
UAP - Universal adversarial perturbation

Can we find a small perturbation that can fool the **entire dataset** of a model?

$$\hat{k}(x+v) \neq \hat{k}(x) \text{ for "most" } x \sim \mu.$$

- Sample an image set X from the distribution.
- Iteratively calculate the minimum perturbation Δv that misclassifies each data point x and accumulate it onto the current universal perturbation v. **DeepFool**
- After each update, project v onto the Lp ball until the desired accuracy is achieved.

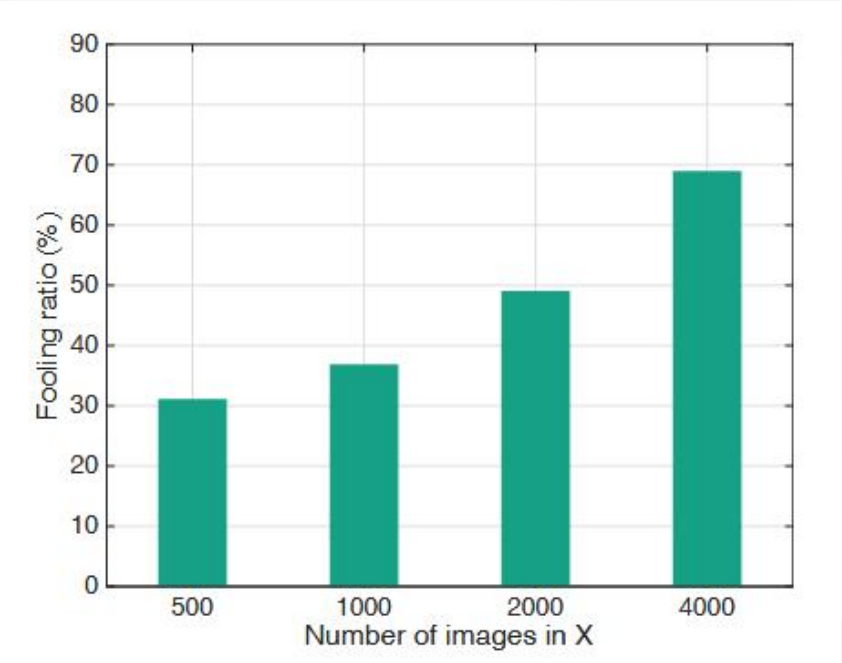


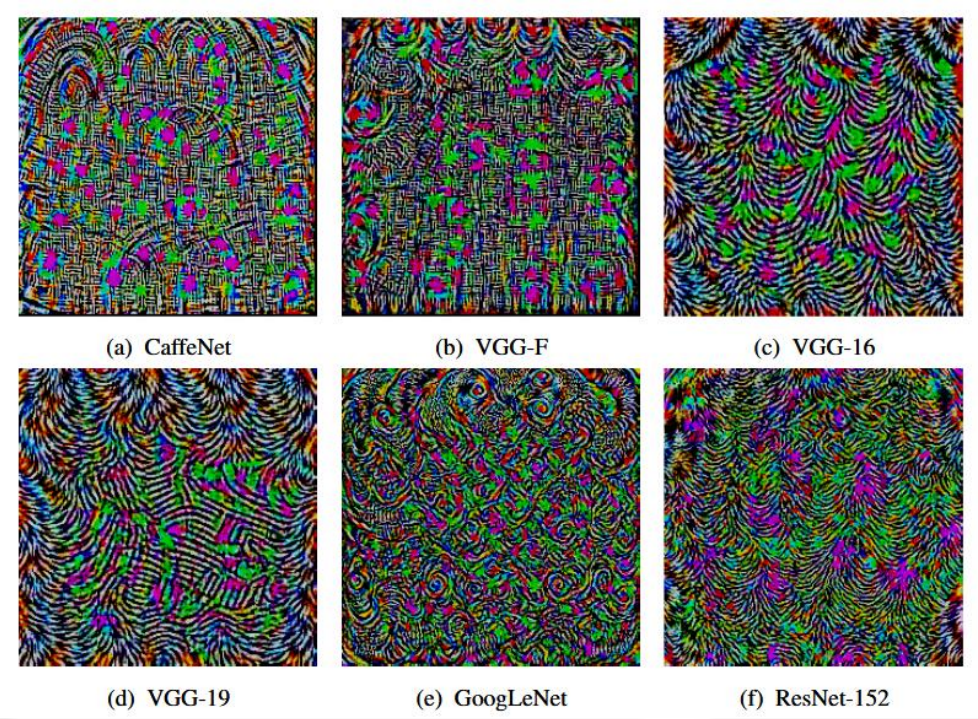


- 几何结构的冗余：实际情况是，虽然决策边界处于高维空间，大部分法向量其实可以由少数几个向量来近似表示，这些向量构成了一个低维子空间，利用这些近似向量就可以实现通用扰动

Moosavi et al., 2017, Universal adversarial

UAP - Universal adversarial perturbations





	VGG-F	CaffeNet	GoogLeNet	VGG-16	VGG-19	ResNet-152
VGG-F	**93.7%**	71.8%	48.4%	42.1%	42.1%	47.4 %
CaffeNet	74.0%	**93.3%**	47.7%	39.9%	39.9%	48.0%
GoogLeNet	46.2%	43.8%	**78.9%**	39.2%	39.8%	45.5%
VGG-16	63.4%	55.8%	56.5%	**78.3%**	73.1%	63.4%
VGG-19	64.0%	57.2%	53.6%	73.5%	**77.8%**	58.0%
ResNet-152	46.3%	46.3%	50.5%	47.0%	45.5%	**84.0%**

- 计算一种网络的UAP对其他网络也有较高误分类率，说明UAP在数据点和网络架构上都具有通用性

Moosavi et al., 2017, Universal adversarial

灰盒攻击 Grey-box

Transfer-based Attack

攻击者训练**替代模型**，利用对抗样本的**可迁移性**对远程未知模型进行攻击。

1. 训练替代模型：攻击者首先训练一个与目标模型执行相同任务的**替代模型**，通过不断迭代优化，使替代模型尽可能逼近目标模型的决策边界。

2. 生成对抗样本：针对训练好的替代模型，运用**白盒攻击**方法生成对抗样本。

3. 迁移攻击：利用相同机器学习技术训练的模型间对抗样本的可迁移性较高，利用不同机器学习技术训练的模型间对抗样本也有一定的的可迁移性。

	RMSD	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
ResNet-152	22.83	0%	13%	18%	19%	11%
ResNet-101	23.81	19%	0%	21%	21%	12%
ResNet-50	22.86	23%	20%	0%	21%	18%
VGG-16	22.51	22%	17%	17%	0%	5%
GoogLeNet	22.58	39%	38%	34%	19%	0%

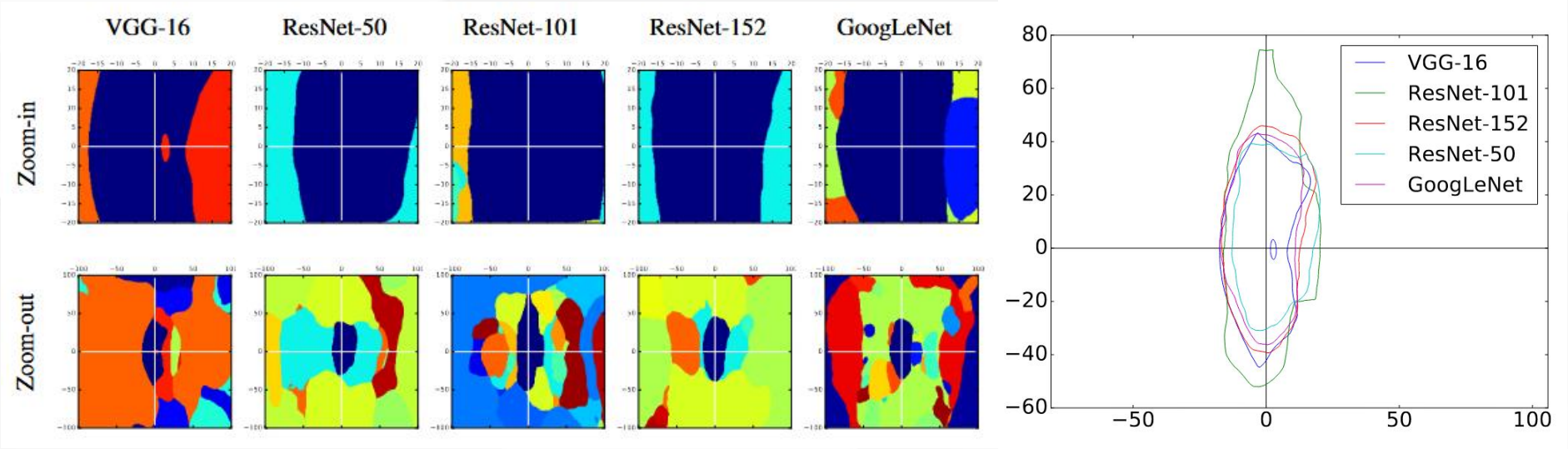
Panel A: Optimization-based approach

	RMSD	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
ResNet-152	23.45	4%	13%	13%	20%	12%
ResNet-101	23.49	19%	4%	11%	23%	13%
ResNet-50	23.49	25%	19%	5%	25%	14%
VGG-16	23.73	20%	16%	15%	1%	7%
GoogLeNet	23.45	25%	25%	17%	19%	1%

Panel B: Fast gradient approach

Brendel et al., 2017, Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine

Transfer-based Attack



不同模型的梯度方向近乎正交，模型正确分类区域集中在中心，且不同模型决策边界对齐良好。

如何提高Transfer-based攻击的迁移效果？

- 以替代模型上的对抗样本作为**先验知识**，再在目标模型上通过**黑盒**攻击方法调整梯度　P-RGF
- 改变替代模型自身**架构**，提高样本迁移能力　DFM　SASD-WS

Brendel et al., 2017, Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine

BPDA - Backward Pass Differentiable Approximation

Adaptive Attack

混淆梯度：

•**破碎梯度：**非可微操作、引入数值不稳定（如温度计编码中，对图像像素进行离散化的温度计编码，这种离散性质导致无法直接进行梯度下降）

•**随机梯度：**网络本身进行随机化（随机激活剪枝），或将输入馈送到分类器之前对输入进行随机变换

•**消失/爆炸梯度：**多次神经网络评估迭代的防御（对抗净化）

当神经网络 $f(\cdot) = f^{1\ldots j}(\cdot)$ 中存在非可微（或不可有效求导）的层 $f^i(\cdot)$ 时，BPDA先找到一个**可微近似函数** $g(x)$，使得 $g(x) \approx f^i(x)$。在计算 $\nabla_x f(x)$ 时，前向传播正常进行（包括通过 $f^i(x)$ 的前向计算），但在**反向传播**时，用 $g(x)$ 替换 $f^i(x)$。只要 $f^i(x)$ 和 $g(x)$ 相似，这种略微不准确的梯度在构建对抗样本时仍然有用。

Straight-through Estimator：g(x) = x $\nabla_x g(x) \approx \nabla_x x = 1$

$$\nabla_x f(g(x))|_{x=\hat{x}} \approx \nabla_x f(x)|_{x=g(\hat{x})}$$

Athalye et al., 2018, Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to

EOT - Expectation over Transformation

Adaptive Attack

如果在分类器对输入进行了随机变换，或者使用随机分类器，就会出现**随机梯度**。

若要攻击这类防御模型，就需要估计随机梯度。

当面对一个先对输入进行随机变换 $t(\cdot)$，再进行分类的模型 $f(\cdot)$ 时，通过对多个不同的随机变换实例 t 对应的 $\nabla f(t(x))$ 进行**期望计算**，来近似得到 $\nabla \mathbb{E}_{t\sim T} f(t(x))$ 的梯度。

在实际操作中，由于直接计算期望往往不可行，通常在每个梯度下降步骤中，计算这些实例下的 $\nabla f(t(x))$ 并求平均，以此来近似期望梯度 $\sum_{i=1}^{k} \nabla_x f(x)$

Attack Evaluation

- **AutoAttack** = APGD + FAB + Square — Lp范式攻击
- **StAdv** — 非Lp范式攻击
- **BPDA + EOT** (+ PGD/C&W ……) — Adaptive攻击

Athalye et al., 2018, Synthesizing Robust Adversarial

Summary

基于梯度的攻击
FGSM -> BIM -> PGD -> **APGD**

基于决策面的攻击
DeepFool -> **FAB**

基于优化的攻击
L-BFGS -> **C&W**

基于几何变换的攻击
StAdv

Unrestricted Attack
AdvGAN

白盒攻击

黑盒攻击

灰盒攻击

基于迁移
Transfer Attack

Adaptive Attack
BPDA EOT

基于分数
- 对称差分 ZOO
- 随机估计 Square
- 自然进化 NES

基于决策 Boundary

OnePixel

UAP