

University of Idaho

CS 502

**Directed Studies: Adversarial
Machine Learning**

Dr. Alex Vakanski

Lecture 5

Evasion Attacks against Machine Learning Models

Lecture Outline

- Evasion attacks against white-box models
 - Carlini and Wagner (2017) Towards Evaluating the Robustness of Neural Networks
 - Xiao et al. (2018) Spatially Transformed Adversarial Examples
 - Other white-box evasion attacks
- Evasion attacks against black-box models
 - Transferability in Adversarial Machine Learning
 - Brendel et al. (2018) Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models
 - Bhagoji et al. (2017) Exploring the Space of Black-box Attacks on Deep Neural Networks
 - Other black-box evasion attacks

Evasion Attacks against White-box Models

- So far we covered:
- *Fast gradient sign method (FGSM) attack*
 - [Goodfellow \(2015\) - Explaining and Harnessing Adversarial Examples](#)
 - $x_{adv} = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(h(x, w), y))$
- *Basic iterative method (BIM) attack*
 - [Kurakin \(2017\) Adversarial Examples in the Physical World](#)
 - $x_{adv}^t = x^{t-1} + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(h(x^{t-1}), y))$
- *Projected gradient descent (PGD) attack*
 - [Madry \(2017\) Towards Deep Learning Models Resistant to Adversarial Attacks](#)
 - $x_{adv}^t = \Pi_\epsilon \left(x^{t-1} + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(h(x^{t-1}), y)) \right)$
- *DeepFool attack*
 - [Moosavi-Dezfooli \(2015\) DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks](#)
 - Iteratively projects the perturbed image to the hyperplane of the closest class

Carlini-Wagner Paper (C-W Attack)

- *Carlini and Wagner (2017) Towards Evaluating the Robustness of Neural Networks*
- The paper proposes three targeted white-box attacks based on different norm metrics:
 - L_∞ attack
 - L_2 attack
 - L_0 attack
- These attacks are sometimes referred to as **C-W attacks**
 - At the time of publishing, they were the strongest adversarial attacks
- Advantages of proposed approaches:
 - Low amount of perturbation
 - Resistance to defense algorithms
 - Generated adversarial images are transferrable across DL models
 - I.e., a secured model is not able to detect the adversarial examples
- Evaluated on: MNIST, CIFAR-10, and ImageNet

Carlini-Wagner Paper (C-W Attack)

- Notation

- Given an image x , a classifier F outputs a vector y , i.e., $F(x) = y$
 - The paper focuses on NN classifiers
 - The output y is treated as a probability distribution, where y_i is the probability that input x has class i

- The **assigned class** by the classifier is

$$C(x) = \operatorname{argmax}_i(y_i) = \operatorname{argmax}_i(F(x)_i)$$

- The **correct label** (true class label) of x is denoted by $C^*(x)$
- The inputs to the softmax function (i.e., the logits) are denoted by z , where the function transforming to input x to the logits is $Z(x)$, i.e.,

$$F(x) = \operatorname{softmax}(Z(x)) = \operatorname{softmax}(z) = y$$

- **Targeted attack**: create an image x' that is similar to x , such that $C(x') = t$, where the target label t is different than the true label $C^*(x)$, i.e., $t \neq C^*(x)$
- **Untargeted attack**: create an image x' that is similar to x , such that $C(x') \neq C^*(x)$
 - The paper considers only targeted attacks, as they are more challenging than untargeted attacks

Carlini-Wagner Paper (C-W Attack)

- Three approaches for selecting the target class were evaluated:
 - **Average Case**: select the target class uniformly at random among the labels that are not the correct label
 - **Best Case**: perform the attack against all incorrect classes, and report the target class that was least difficult to attack
 - **Worst Case**: perform the attack against all incorrect classes, and report the target class that was most difficult to attack
- The used NN models for MNIST and CIFAR datasets are shown below
 - For ImageNet the paper used the Inception-v3 network

Layer Type	MNIST Model	CIFAR Model
Convolution + ReLU	3×3×32	3×3×64
Convolution + ReLU	3×3×32	3×3×64
Max Pooling	2×2	2×2
Convolution + ReLU	3×3×64	3×3×128
Convolution + ReLU	3×3×64	3×3×128
Max Pooling	2×2	2×2
Fully Connected + ReLU	200	256
Fully Connected + ReLU	200	256
Softmax	10	10

TABLE I

MODEL ARCHITECTURES FOR THE MNIST AND CIFAR MODELS. THIS ARCHITECTURE IS IDENTICAL TO THAT OF THE ORIGINAL DEFENSIVE DISTILLATION WORK. [39]


Parameter	MNIST Model	CIFAR Model
Learning Rate	0.1	0.01 (decay 0.5)
Momentum	0.9	0.9 (decay 0.5)
Delay Rate	-	10 epochs
Dropout	0.5	0.5
Batch Size	128	128
Epochs	50	50


TABLE II


MODEL PARAMETERS FOR THE MNIST AND CIFAR MODELS. THESE PARAMETERS ARE IDENTICAL TO THAT OF THE ORIGINAL DEFENSIVE DISTILLATION WORK. [39]

Carlini-Wagner Paper (C-W Attack)

- Initial problem formulation
 - Create an adversarial image x' by adding small perturbation δ to the original image x (i.e., $x' = x + \delta$), such that the distance $\mathcal{D}(x, x') = \mathcal{D}(x, x + \delta)$ is minimal
 - The classifier should assign the class label t to the adversarial image x' , where t is different than the true label $C^*(x)$, i.e., $C(x') = C(x + \delta) = t \neq C^*(x)$
 - The goal is to find δ that minimizes $\mathcal{D}(x, x + \delta)$ and $C(x + \delta) = t$

minimize $\mathcal{D}(x, x + \delta)$  distance between x and $x + \delta$

such that $C(x + \delta) = t$  $x + \delta$ is classified as target class t

$x + \delta \in [0, 1]^n$  each element of $x + \delta$ is in $[0, 1]$ (to be a valid image)

Carlini-Wagner Paper (C-W Attack)

- This initial formulation of the optimization problem for creating adversarial attacks is difficult to solve
 - Because the constraint $C(x + \delta) = t$ is highly non-linear

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

- Carlini-Wagner propose the following reformulation of the optimization problem, which is solvable
 - The function f should be chosen such that $C(x + \delta) = t$ if and only if $f(x + \delta) \leq 0$
 - These two optimization problems are not identical: the reformulation by Carlini-Wagner just finds an approximated solution to the above problem
 - Adam optimization algorithm is used for solving the problem

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

Carlini-Wagner Paper (C-W Attack)

- Recall the solution of constrained optimization problems from Lecture 4 using **Lagrange multipliers**

$$\begin{array}{l} \underset{\mathbf{x}}{\text{minimize}} f(\mathbf{x}) \\ \text{subject to } c_i(\mathbf{x}) \leq 0 \end{array} \quad \longrightarrow \quad \underset{\mathbf{x}}{\text{minimize}} f(\mathbf{x}) + \sum_i \alpha_i c_i(\mathbf{x})$$

- The same approach can be applied to the Carlini-Wagner approach, and the optimization problem can be rewritten as shown below
 - The authors performed a grid search for the value of the parameter c
 - The recommended approach is to select the smallest value of c where $c > 0$, for which $f(x + \delta) \leq 0$ and the distance $\mathcal{D}(x, x + \delta)$ is minimal

$$\begin{array}{l} \text{minimize } \mathcal{D}(x, x + \delta) \\ \text{such that } f(x + \delta) \leq 0 \end{array} \quad \longrightarrow \quad \text{minimize } \mathcal{D}(x, x + \delta) + c \cdot f(x + \delta)$$

Carlini-Wagner Paper (C-W Attack)

- The authors considered several variants for the function f
 - In the equations below, $\text{loss}_{F,t}(x')$ is the loss function with respect to the target class t
 - The class labels are denoted by i
 - Other notation: $(a)^+ = \max(0, a)$; $\text{softplus}(a) = \log(1 + e^a)$
- The best results were obtained by the function $f_6(x')$

$$f_1(x') = -\text{loss}_{F,t}(x') + 1$$

$$f_2(x') = (\max_{i \neq t} (F(x')_i) - F(x')_t)^+$$

$$f_3(x') = \text{softplus}(\max_{i \neq t} (F(x')_i) - F(x')_t) - \log(2)$$

$$f_4(x') = (0.5 - F(x')_t)^+$$

$$f_5(x') = -\log(2F(x')_t - 2)$$

$$f_6(x') = (\max_{i \neq t} (Z(x')_i) - Z(x')_t)^+$$

$$f_7(x') = \text{softplus}(\max_{i \neq t} (Z(x')_i) - Z(x')_t) - \log(2)$$

Carlini-Wagner Paper (C-W Attack)

- Explanation of the function $f_6(x')$
 - In f_6 , $Z(x')_t$ is the logits value of the target class t for the perturbed image x'
 - Then, $\max_{i \neq t}(Z(x')_i)$ means the maximum logits values of other class i than the target class t (i.e., $i \neq t$)
 - The function calculates the difference in the logits between the target class t and the closest-to-the-target class
 - In some papers, this function is referred to as **margin loss function**

$$f_6(x') = (\max_{i \neq t}(Z(x')_i) - Z(x')_t)^+$$

- In the paper, a modified function f_6 is also provided
 - It introduces a confidence value k
 - The authors set $k = 0$
 - But, if k has a higher value, this will require that any other logits value exceeds the logits value of the true class $Z(x')_t$ at least by k
 - Examples with large confidence value k have enhanced transferability

$$f(x') = \max(\max\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa)$$

Carlini-Wagner Paper (C-W Attack)

- L_∞ attack

- The used distance metric is L_∞ norm, therefore $\mathcal{D}(x, x + \delta) = \|\delta\|_\infty$
- In other words, $\|\delta\|_\infty$ means the pixel in x' with the largest change from x

- The optimization problem becomes:

$$\text{minimize } \mathcal{D}(x, x + \delta) + c \cdot f(x + \delta) \quad \longrightarrow \quad \text{minimize } c \cdot f(x + \delta) + \|\delta\|_\infty$$

- However, this formulation produced poor optimization results, since the term $\|\delta\|_\infty$ penalizes only the largest component of the perturbation vector δ
- The authors proposed the following optimization method instead
 - In this case, any component of δ that exceed a threshold value τ is considered, that is, penalize all components of δ that have large values
 - The value of τ is set initially to 1, and is decreased by a factor of 0.9 after each iteration
 - I.e., $\tau \rightarrow \tau \cdot 0.9$ if all $\delta_i < \tau$, else terminate the search

$$\text{minimize } c \cdot f(x + \delta) + \sum_i [(\delta_i - \tau)^+]$$

Carlini-Wagner Paper (C-W Attack)

- *Box constraint*

- In the optimization problem, the constraint $x + \delta \in [0, 1]^n$ requires that in the perturbed images, all pixel values are in the $[0,1]$ range
- I.e., $0 \leq x_i + \delta_i \leq 1$ for all i
- This is called a box constraint
 - Or, these values can be within the range $[0,255]$ depending on how the images are scaled

- The box constraint can cause difficulties in solving the optimization problem
 - Simply clipping the values can cause that optimization to get stuck in a flat region
- The authors introduced a new variable w , such that

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

- As we know $-1 \leq \tanh(w_i) \leq 1$, therefore it follows $0 \leq x_i + \delta_i \leq 1$
- This change of variables produced more stable optimization results

Carlini-Wagner Paper (C-W Attack)

- *L₂ attack*

- The used distance metric is L_2 norm, therefore $\mathcal{D}(x, x + \delta) = \|\delta\|_2$

- Using the variable w for the box-constraint, the optimization problems becomes

$$\text{minimize } \|\delta\|_2^2 + c \cdot f(x + \delta) \quad \text{where} \quad \delta = \frac{1}{2}(\tanh(w) + 1) - x$$

$$\text{minimize } \left\| \frac{1}{2}(\tanh(w) + 1) - x \right\|_2^2 + c \cdot f\left(\frac{1}{2}(\tanh(w) + 1)\right)$$

- That is, search for w that minimizes the above term

- The function f is based on the $f_6(x')$ variant provided earlier

$$f(x') = \max(\max\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa)$$

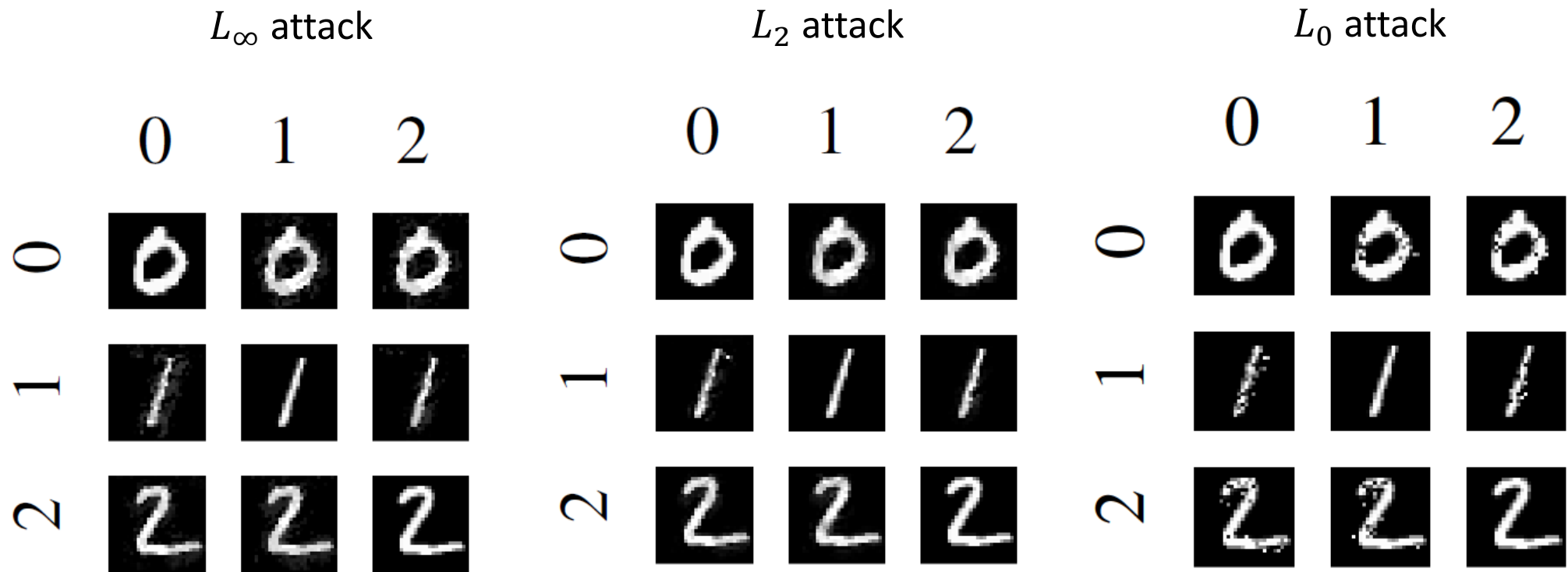
- To avoid the cases when the gradient descent algorithm become stuck in a local minimum, the authors picked multiple random starting points close to the original image x

Carlini-Wagner Paper (C-W Attack)

- *L_0 attack*
 - The used distance metric is L_0 norm, or, the number of non-zero pixels in δ
- The authors propose an iterative approach
 - Where the goal at each iteration is to find pixels that are not important and don't have much effect on the classifier's output
- The iterative procedure includes the following steps:
 - Initialization: the allowed set includes all pixels in the image
 - Perform L_2 attack to find an adversarial example $x + \delta$
 - Compute the gradient $g = \nabla f(x + \delta)$, where f is the objective function in the L_2 attack
 - Identify the least important pixel $i = \operatorname{argmin}_i g_i \delta_i$ and remove this pixel from the allowed set
 - Iterate until the L_2 attack fails to find an adversarial example
- The approach shrinks the set of pixels that are allowed to be changed, until a minimum number of pixels is found that change the class label to the target t

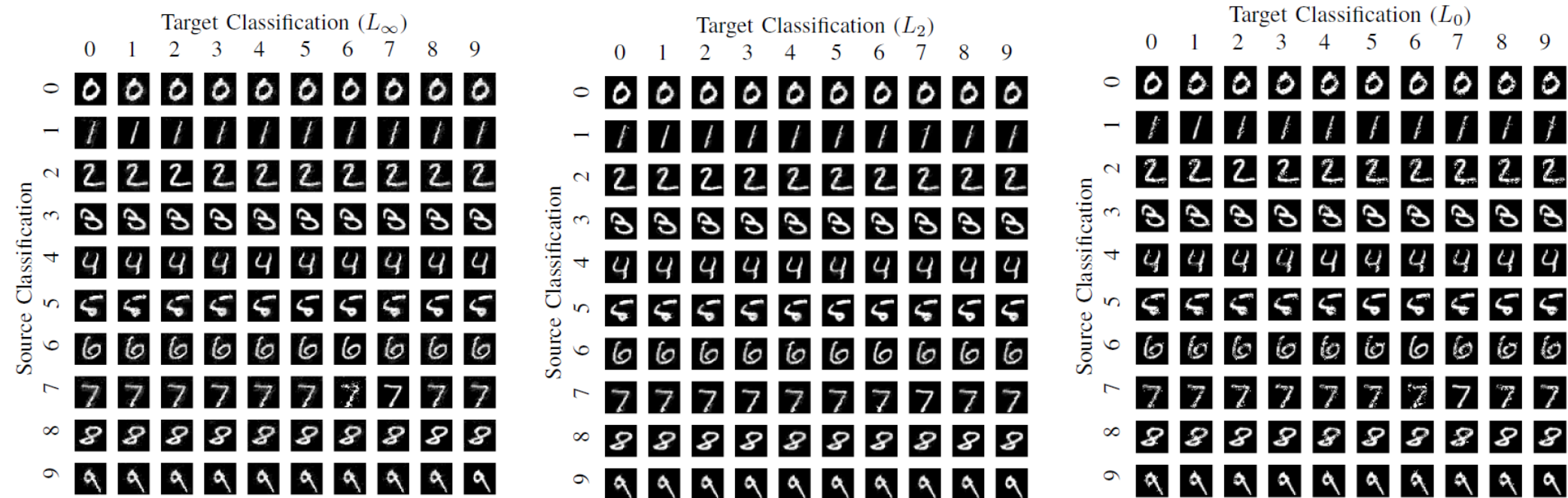
Carlini-Wagner Paper (C-W Attack)

- Results on the MNIST dataset



Carlini-Wagner Paper (C-W Attack)

- Results on the MNIST dataset



Carlini-Wagner Paper (C-W Attack)

- Validation on the MNIST and CIFAR datasets
- Comparison to JSMA (Jacobian-based Saliency Map Attack), DeepFool, Fast Gradient Sign, and Iterative Gradient Sign attacks
 - Mean is the perturbation size

	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%

TABLE IV

COMPARISON OF THE THREE VARIANTS OF TARGETED ATTACK TO PREVIOUS WORK FOR OUR MNIST AND CIFAR MODELS. WHEN SUCCESS RATE IS NOT 100%, THE MEAN IS ONLY OVER SUCCESSES.

Carlini-Wagner Paper (C-W Attack)

- Validation on the ImageNet dataset

	Untargeted		Average Case		Least Likely	
	mean	prob	mean	prob	mean	prob
Our L_0	48	100%	410	100%	5200	100%
JSMA-Z	-	0%	-	0%	-	0%
JSMA-F	-	0%	-	0%	-	0%
Our L_2	0.32	100%	0.96	100%	2.22	100%
Deepfool	0.91	100%	-	-	-	-
Our L_∞	0.004	100%	0.006	100%	0.01	100%
FGS	0.004	100%	0.064	2%	-	0%
IGS	0.004	100%	0.01	99%	0.03	98%

TABLE V

COMPARISON OF THE THREE VARIANTS OF TARGETED ATTACK TO PREVIOUS WORK FOR THE INCEPTION V3 MODEL ON IMAGENET. WHEN SUCCESS RATE IS NOT 100%, THE MEAN IS ONLY OVER SUCCESSES.

Xiao, Li, Song Paper (stAdv Attack)

- *Xiao, Zhu, Li, He, Liu, Song (2018) Spatially Transformed Adversarial Examples*
 - This method is sometimes referred to as **stAdv attack**
- The paper proposes an attack that does not manipulate the pixel intensity values under an L_p norm
- Instead, the pixels are spatially moved in an image to create an adversarial example
 - Such attack can result in a large L_p distance between the original and manipulated images
 - Still, the images are perceptually realistic
 - The perturbed images are effective against defense algorithms
- The approach minimizes the local geometric distortion of images
- Validation: MNIST, CIFAR-10, and ImageNet datasets

Xiao, Li, Song Paper (stAdv Attack)

- Example of a spatially transformed image
 - The red flow arrows indicate the local displacement of the pixels in adversarial image to the pixels in the input image

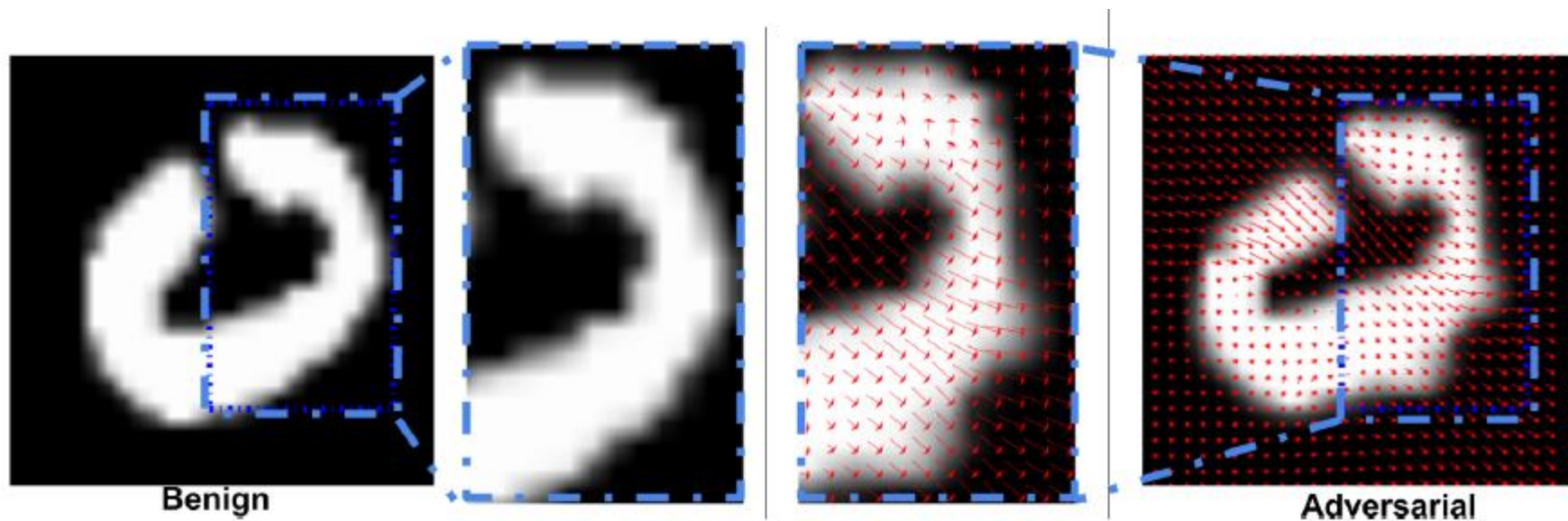
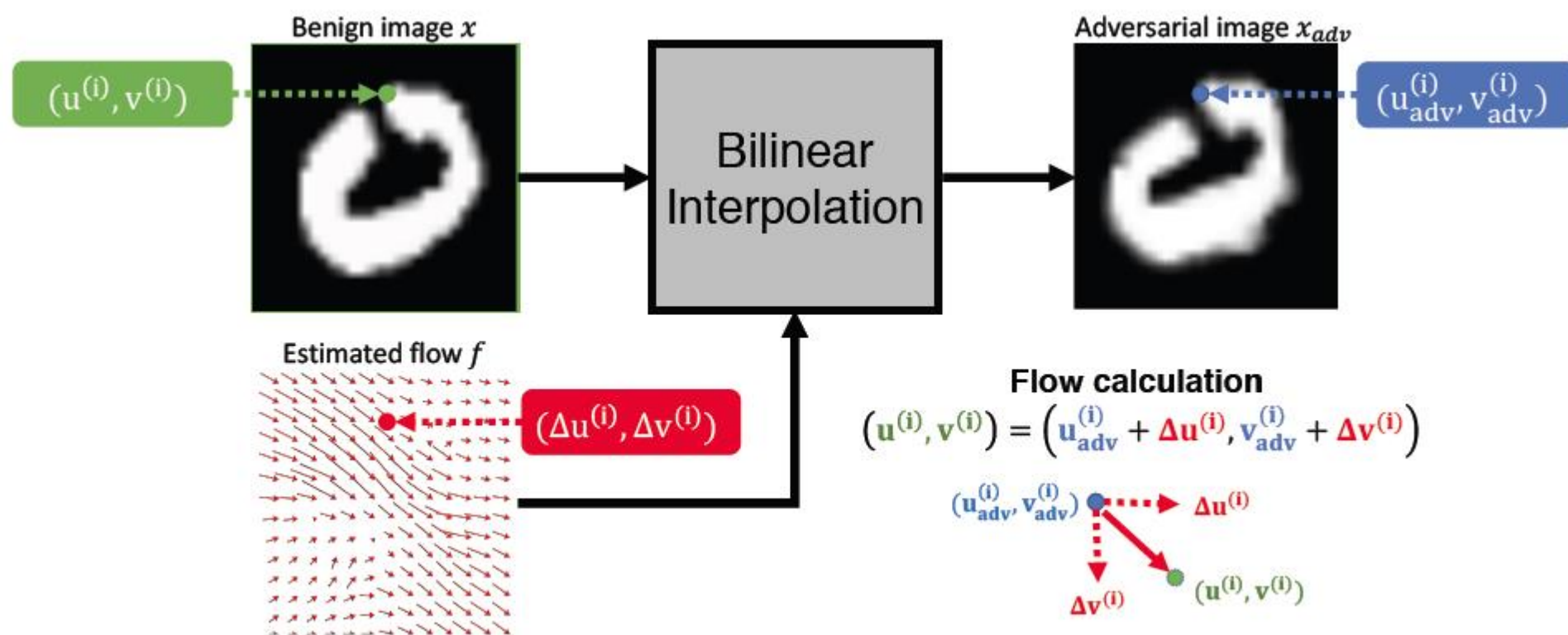


Figure 5: Flow visualization on MNIST. The digit "0" is misclassified as "2".

Xiao, Li, Song Paper (stAdv Attack)

- Green color – the pixel i in the input (benign, clean) image
- Blue color – the spatially displaced pixel i in the adversarial image
- Red arrows – the displacement flow f : horizontal ($\Delta u^{(i)}$) and vertical ($\Delta v^{(i)}$)



Xiao, Li, Song Paper (stAdv Attack)

- A **targeted white-box attack** is considered
- The problem is formulated as an optimization problem, that is very similar to the Carlini-Wagner paper
- For an image x , find the minimum local distortion f^* , such that

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

- The term \mathcal{L}_{adv} encourages the distorted image to be misclassified as the target class t
- The term \mathcal{L}_{flow} ensure that the spatial transformation is preserved
- τ is a constant that balances the two terms (set to 0.05 for validation)
- The authors adopted the $f_6(x')$ function from Carlini-Wagner for the term \mathcal{L}_{adv}
 - That maximizes the logits values of the target class t with respect to other classes

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

Xiao, Li, Song Paper (stAdv Attack)

- The term \mathcal{L}_{flow} is calculated as the sum of spatial movement distance for any two pixels p and q
 - This makes the stAdv approach computationally expensive, because it requires calculating the distances for all pairs of pixels

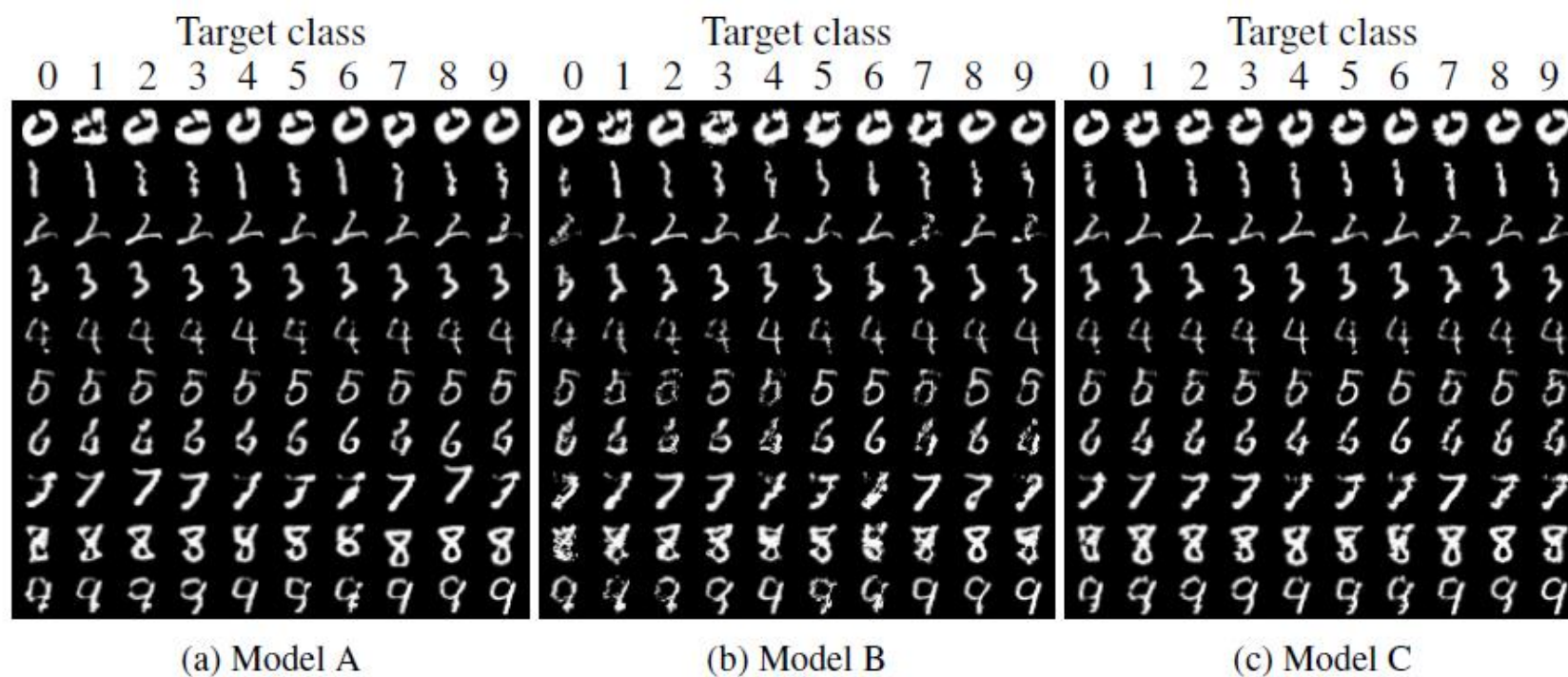
$$\mathcal{L}_{flow}(f) = \sum_p^{all\ pixels} \sum_{q \in \mathcal{N}(p)} \sqrt{\|\Delta u^{(p)} - \Delta u^{(q)}\|_2^2 + \|\Delta v^{(p)} - \Delta v^{(q)}\|_2^2}.$$

- The optimization problem is solved using the L-BFGS algorithm (Limited-memory BFGS (Broyden–Fletcher–Goldfarb–Shanno))

Xiao, Li, Song Paper (stAdv Attack)

- Validation on MNIST for three different NN model architectures A, B, and C
 - Accuracy (p) means the model classification accuracy on pristine (original) images

Model	A	B	C
Accuracy (p)	98.58%	98.94%	99.11%
Attack Success Rate	99.95%	99.98%	100.00%



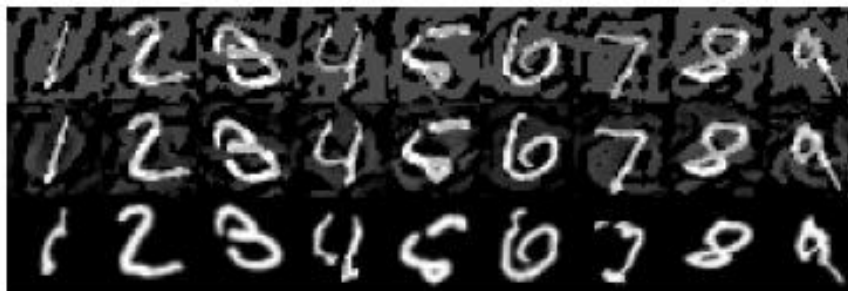
Xiao, Li, Song Paper (stAdv Attack)

- For CIFAR-10 images, they used ResNet32 and Wide ResNet34

Model	ResNet32 (0.47M)	Wide ResNet34 (46.16M)
Accuracy (p)	93.16%	95.82%
Attack Success Rate	99.56%	98.84%

- Comparison of adversarial examples generated by FGSM, C&W, and stAdv
 - Left: MNIST, right: CIFAR-10

FGSM



C&W



StAdv

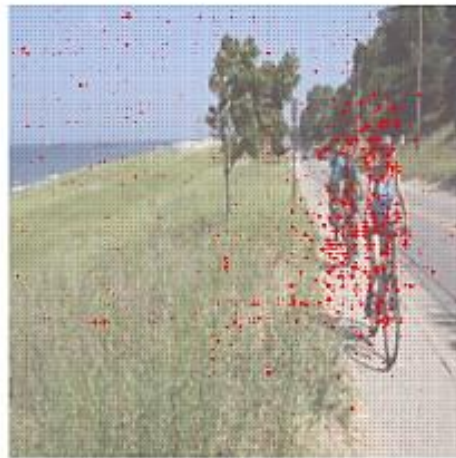


Xiao, Li, Song Paper (stAdv Attack)

- Flow visualization on ImageNet
 - (a): the original image, (b)-(c): images are misclassified into goldfish, dog and cat
 - Although there are other objects within the image, most spatial transformation flows focus on the target object – mountain bike



(a) mountain bike



(b) goldfish



(c) Maltese dog

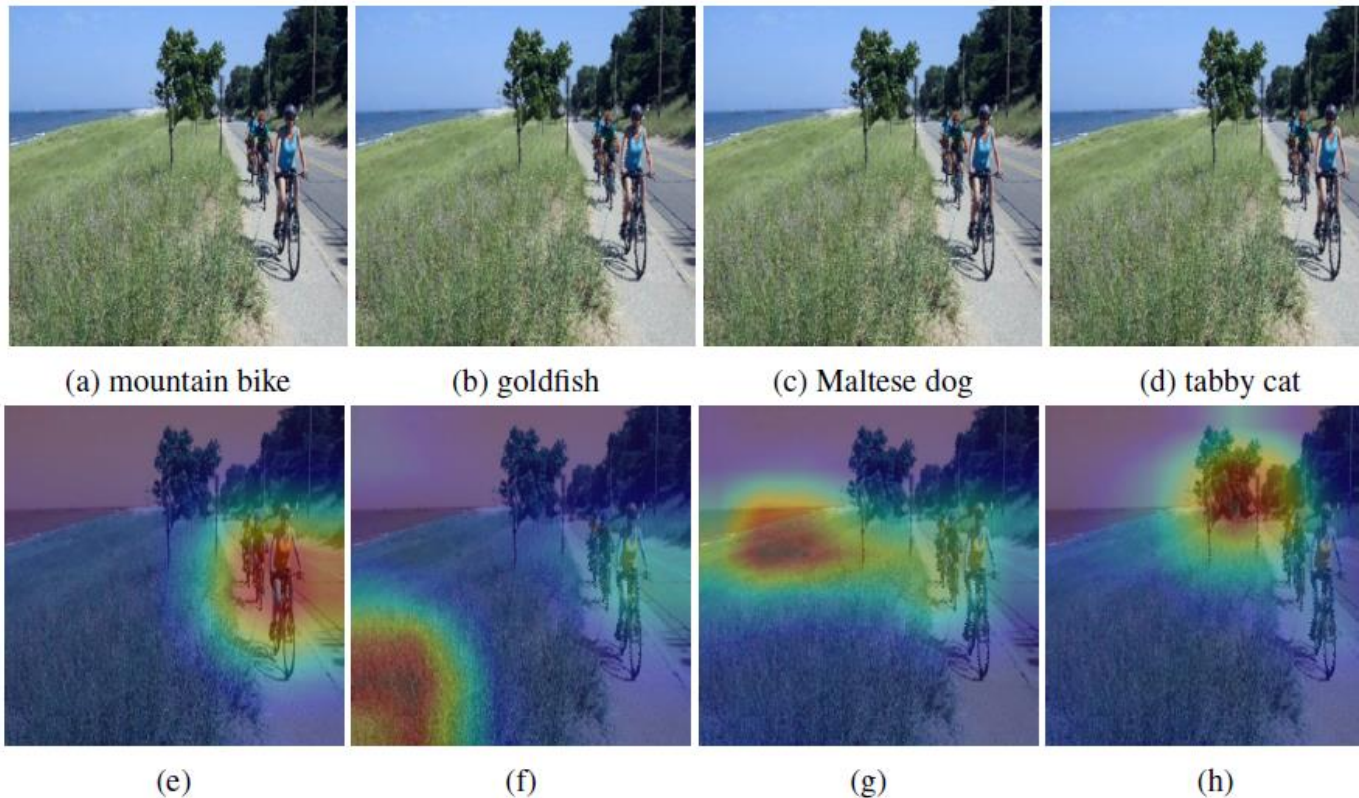


(d) tabby cat

- Human participants on Amazon Mechanical Turk (AMT) were recruited to analyze the visual perceptibility of attacked images
 - The users selected the attacked images as visually realistic

Xiao, Li, Song Paper (stAdv Attack)

- Further analysis includes visualizing the saliency maps of images
 - I.e., find the regions in the images where the model pays the most attention for assigning a particular class to an images
 - Class Activation Mapping (CAM) was used for this purpose
 - The stAdv attack misleads the model to pay attention to different regions than the bike



Xiao, Li, Song Paper (stAdv Attack)

- Attack evaluation under three defense methods: adversarial training (Adv.), ensemble adversarial training (Ens.), and projectile gradient descent (PGD)

Table 3: Attack success rate of adversarial examples generated by stAdv against models A, B, and C under standard defenses on MNIST, and against ResNet and wide ResNet on CIFAR-10.

Model	Def.	FGSM	C&W.	stAdv
A	Adv.	4.3%	4.6%	32.62%
	Ens.	1.6%	4.2%	48.07%
	PGD	4.4%	2.96%	48.38%
B	Adv.	6.0%	4.5%	50.17%
	Ens.	2.7%	3.18%	46.14%
	PGD	9.0%	3.0%	49.82%
C	Adv.	3.22%	0.86%	30.44%
	Ens.	1.45%	0.98%	28.82%
	PGD	2.1%	0.98%	28.13%

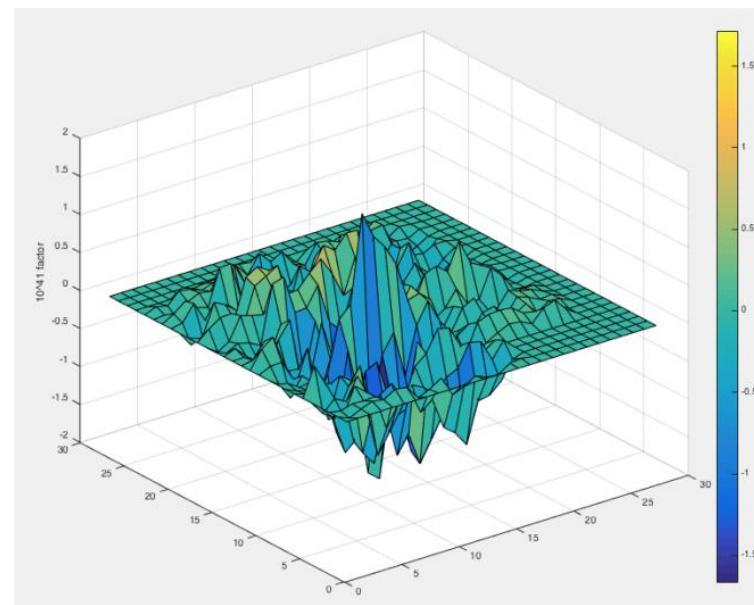
Model	Def.	FGSM	C&W.	stAdv
ResNet32	Adv.	13.10%	11.9%	43.36%
	Ens.	10.00%	10.3%	36.89%
	PGD	22.8%	21.4%	49.19%
wide ResNet34	Adv.	5.04%	7.61%	31.66%
	Ens.	4.65%	8.43%	29.56%
	PGD	14.9%	13.90%	31.6%

Other White-box Evasion Attacks

- *Jacobian-based Saliency Map Attack (JSMA)*
 - [Papernot et al. \(2016\) - The limitations of deep learning in adversarial settings](#)
- Targeted white-box attack based on controlling the L_0 norm
 - The goal is to iteratively change each pixel until misclassification
 - The key step is calculation of a saliency map that determines which pixels to be modified, in order to increase the probability of the target class
 - The map is based on the Jacobian matrix of the first partial derivatives w.r.t. input

Compute $\nabla F(x)$

Jacobian matrix



Saliency Map



Modify x

Pixels with large saliency values have large impact on the output when perturbed

Other White-box Evasion Attacks

- *NewtonFool Attack*
 - [Jang et al. \(2017\) - Objective metrics and gradient descent algorithms for adversarial examples in machine learning](#)
- The approach is similar to iterative FGSM attacks
 - Performs iterative gradient descent with an adaptive step size

Input:

x : Input to be adversarially perturbed
 η : Strength of adversarial perturbations
 i_{\max} : Maximum number of iterations

- 1: $y \leftarrow C(x), x_{\text{adv}} \leftarrow x, i \leftarrow 0$
- 2: **while** $i < i_{\max}$ **do**
- 3: Compute

$$\delta \leftarrow \min \{ \eta \cdot \|x\|_2 \cdot \|\nabla F_y(x_{\text{adv}})\|, F_y(x_{\text{adv}}) - 1/K \},$$

$$d \leftarrow -\frac{\delta \cdot \nabla F_y(x_{\text{adv}})}{\|\nabla F_y(x_{\text{adv}})\|_2^2}$$

- 4: $x_{\text{adv}} \leftarrow \text{clip}(x_{\text{adv}} + d, x_{\min}, x_{\max})$
- 5: $i \leftarrow i + 1$
- 6: **end while**

Output:

Adversarial sample x_{adv} .

Other White-box Evasion Attacks

- *Elastic Net (EAD) Attack*
 - [Chen et al. \(2017\) Ead: Elastic-net attacks to deep neural networks via adversarial exam](#)
- Modification of the C-W attack for controlling the L_1 norm of adversarial perturbations
 - Employs a box constraint based on Iterative Shrinkage-Thresholding Algorithm (ISTA)

Algorithm 1 Elastic-Net Attacks to DNNs (EAD)

Input: original labeled image (\mathbf{x}_0, t_0) , target attack class t , attack transferability parameter κ , L_1 regularization parameter β , step size α_k , # of iterations I

Output: adversarial example \mathbf{x}

Initialization: $\mathbf{x}^{(0)} = \mathbf{y}^{(0)} = \mathbf{x}_0$

for $k = 0$ to $I - 1$ **do**

$$\mathbf{x}^{(k+1)} = S_\beta(\mathbf{y}^{(k)} - \alpha_k \nabla g(\mathbf{y}^{(k)}))$$

$$\mathbf{y}^{(k+1)} = \mathbf{x}^{(k+1)} + \frac{k}{k+3}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$$

end for

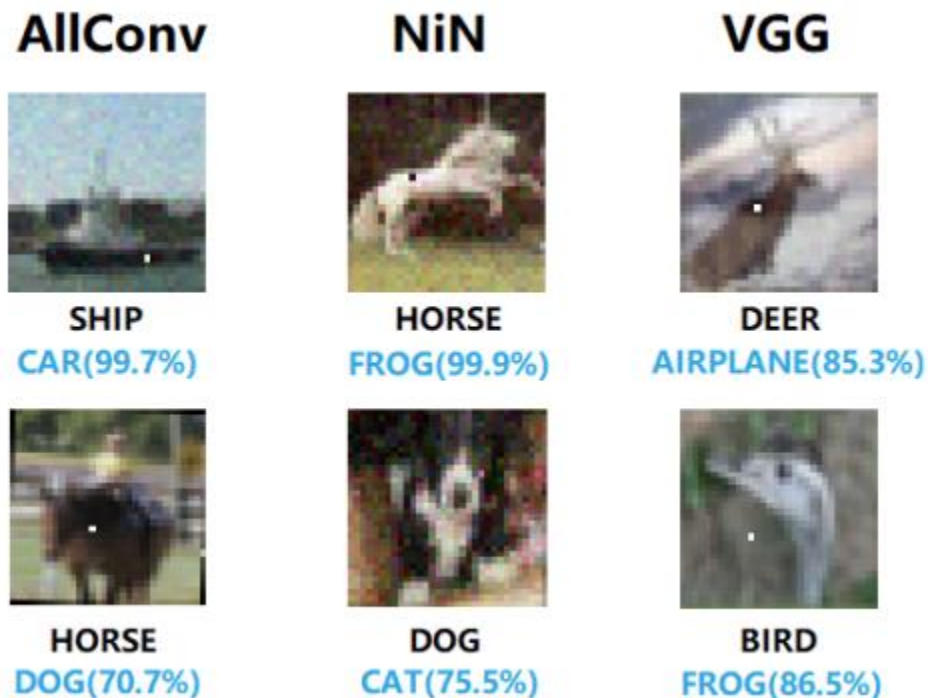
Decision rule: determine \mathbf{x} from successful examples in $\{\mathbf{x}^{(k)}\}_{k=1}^I$ (EN rule or L_1 rule).

$S_\beta : \mathbb{R}^p \mapsto \mathbb{R}^p$ is an element-wise projected shrinkage-thresholding function, which is defined as

$$[S_\beta(\mathbf{z})]_i = \begin{cases} \min\{\mathbf{z}_i - \beta, 1\}, & \text{if } \mathbf{z}_i - \mathbf{x}_{0i} > \beta; \\ \mathbf{x}_{0i}, & \text{if } |\mathbf{z}_i - \mathbf{x}_{0i}| \leq \beta; \\ \max\{\mathbf{z}_i + \beta, 0\}, & \text{if } \mathbf{z}_i - \mathbf{x}_{0i} < -\beta, \end{cases}$$

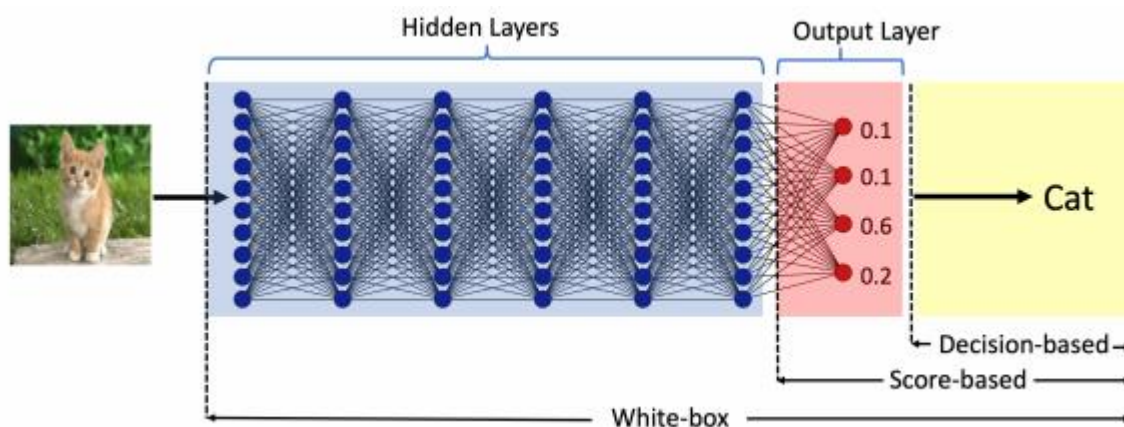
Other White-box Evasion Attacks

- *One-pixel Attack*
 - [Su et al. \(2019\) One pixel attack for fooling deep neural networks](#)
- Attack under the L_0 norm to limit the number of pixels allowed to be changed
 - Based on Differential Evolution-based optimization
- It shows that on CIFAR-10 dataset, most of the testing samples can be attacked in an untargeted manner by changing the value of only one pixel



Evasion Attacks against Black-box Models

- The black-box attacks can be classified into two categories:
 - *Query-based attacks*
 - The adversary queries the model and creates adversarial examples by using the provided information to queries
 - The queried model can provide:
 - Output class probabilities (i.e., confidence scores per class) used with **score-based attacks**
 - Output class, used with **decision-based attacks**
 - *Transfer-based attacks* (or *transferability attacks*)
 - The adversary does not query the model
 - The adversary trains its own substitute/surrogate local model, and transfers the adversarial examples to the target model
 - This type of approaches are also referred to as **zero queries attacks**



Transfer-based Black-box Models

- *Substitute model attack* (or *surrogate local model attack*)
 - [Papernot et al. \(2016\) Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples](#)
 - Uses FGSM or PGD for attacking a substitute model, and afterward transfer the generated adversarial samples to the target model
 - The ability to attack a classifier model by using a substitute model is called **transferability**
- *Ensemble of local models attack*
 - [Liu et al. \(2017\) Delving into Transferable Adversarial Examples and Black-box Attacks](#)
 - Uses an ensemble of local models for generating adversarial examples

Brendel Paper (Boundary Attack)

- *Brendel, Rauber, and Bethge (2018) Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models*
- Proposed a query-based black-box attack called **Boundary Attack**
 - The attack requires only queries of the output class, not of the logit or of output probabilities
 - Can perform both untargeted and targeted attacks
- Advantage:
 - Finds low-perturbation images only by using the output class information
 - Relevant to real-world application where access to the model may not be possible
- Disadvantage:
 - Requires many iterations to converge
- Validation on MNIST, CIFAR, and ImageNet
 - For ImageNet: VGG-19, ResNet50, and Inception-v3
 - And, on real-world applied models

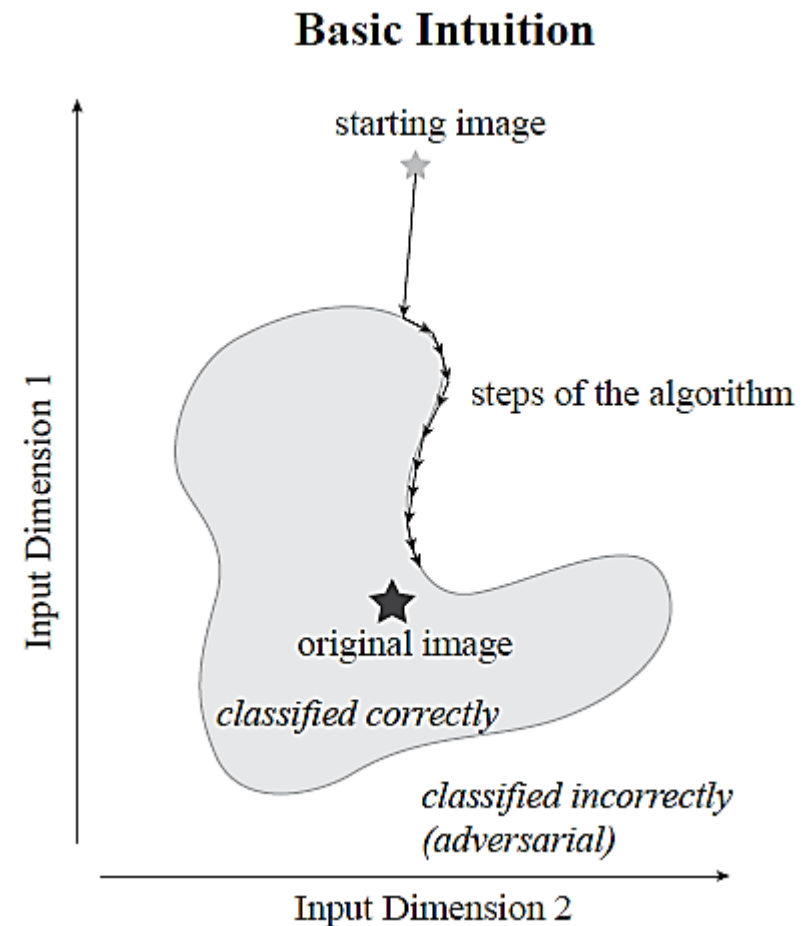
Brendel Paper (Boundary Attack)

- Attack classification
 - **Gradient-based attacks** need access to the model gradients (e.g., with respect to the inputs); defense by masking the gradients by distillation or saturation
 - **Transfer-based attacks** use a substitute model to train attack samples, needs some information about the training data; defense by adversarial training
 - **Score-based attacks** need access to either the logits or output probabilities; defense by dropout, ensemble adversarial training
 - **Decision-based attack** needs access to the final decision by the model (e.g., output class)

	Gradient-based Model M	Transfer-based Training Data T	Score-based Detailed Model Prediction Y (e.g. probabilities or logits)	Decision-based Final Model Prediction Y_{\max} (e.g. max class label)
	<i>less information</i> →			
Untargeted Flip to any label	FGSM, DeepFool	FGSM Transfer	Local Search	★ this work (Boundary Attack)
Targeted Flip to target label	L-BFGS-B, Houdini, JSMA, Carlini & Wagner, Iterative Gradient Descent	Ensemble Transfer	ZOO	★

Brendel Paper (Boundary Attack)

- Boundary Attack intuition
 - The starting image is drawn from a uniform random distribution, and is adversarial (i.e., different than the true class)
 - Iteratively reduce the L_2 distance to the original image by adding small perturbations
 - Walk along the boundary between the adversarial and the non-adversarial region, but stay in the adversarial region
 - I.e., whenever the added perturbation results in correct classification, reject those samples (a.k.a. sample rejection)
 - When the distance to the original image cannot be further reduced, or when the number of set iteration steps is reached, stop



Brendel Paper (Boundary Attack)

- Boundary Attack algorithm
 - The initial image $\tilde{\mathbf{o}}^0$ is sampled from a uniform distribution $\mathcal{U}(0,1)$
 - The adversarially perturbed image at the k^{th} step is denoted by $\tilde{\mathbf{o}}^k$
 - Adversarial criterion $c(\cdot)$ in this case is: misclassification
 - Different class than the true class (untargeted attack) or the target class (targeted attack)
 - Decision model $d(\cdot)$ is L_2 distance between the perturbed and the original image
 - The proposal distribution for the perturbation η_k is discussed on next page

Data: original image \mathbf{o} , adversarial criterion $c(\cdot)$, decision of model $d(\cdot)$

Result: adversarial example $\tilde{\mathbf{o}}$ such that the distance $d(\mathbf{o}, \tilde{\mathbf{o}}) = \|\mathbf{o} - \tilde{\mathbf{o}}\|_2^2$ is minimized

initialization: $k = 0$, $\tilde{\mathbf{o}}^0 \sim \mathcal{U}(0, 1)$ s.t. $\tilde{\mathbf{o}}^0$ is adversarial;

while $k < \text{maximum number of steps}$ **do**

 draw random perturbation from proposal distribution $\eta_k \sim \mathcal{P}(\tilde{\mathbf{o}}^{k-1})$;

if $\tilde{\mathbf{o}}^{k-1} + \eta_k$ is adversarial **then**

 set $\tilde{\mathbf{o}}^k = \tilde{\mathbf{o}}^{k-1} + \eta_k$;

else

 set $\tilde{\mathbf{o}}^k = \tilde{\mathbf{o}}^{k-1}$;

end

$k = k + 1$

end

Brendel Paper (Boundary Attack)

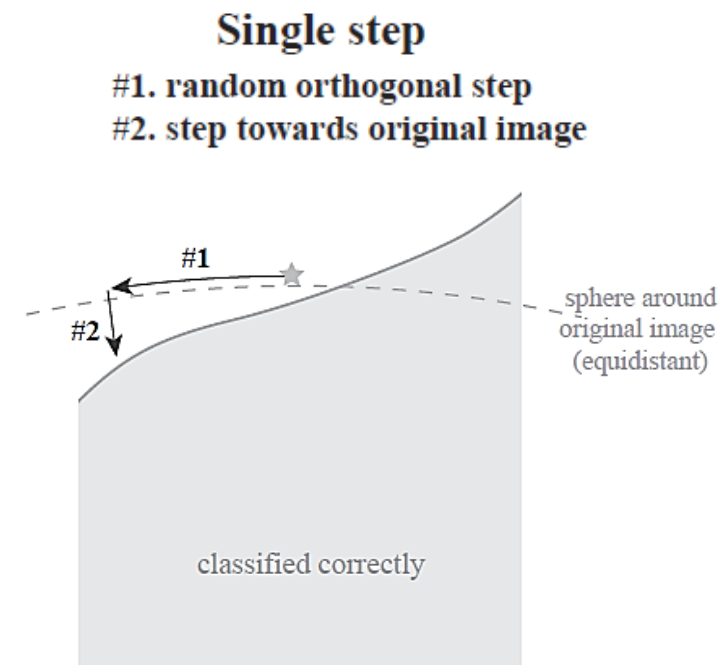
- For the proposal distribution $\mathcal{P}(\tilde{\mathbf{o}}^{k-1})$ of the perturbation η_k , the authors proposed to use a Gaussian distribution $\mathcal{N}(0,1)$
 - This perturbation is denoted as #1 – random orthogonal step
- Next, it is ensured that the proposed adversarial sample is a regular image with all pixels clipped in the range $(0,255)$

$$\tilde{\mathbf{o}}_i^{k-1} + \eta_i^k \in [0,255]$$

- It is also ensured that the perturbation η_k is within a ball with radius δ (i.e., the adversarial image $\tilde{\mathbf{o}}^{k-1}$ is projected into the δ sphere from the original image \mathbf{o})

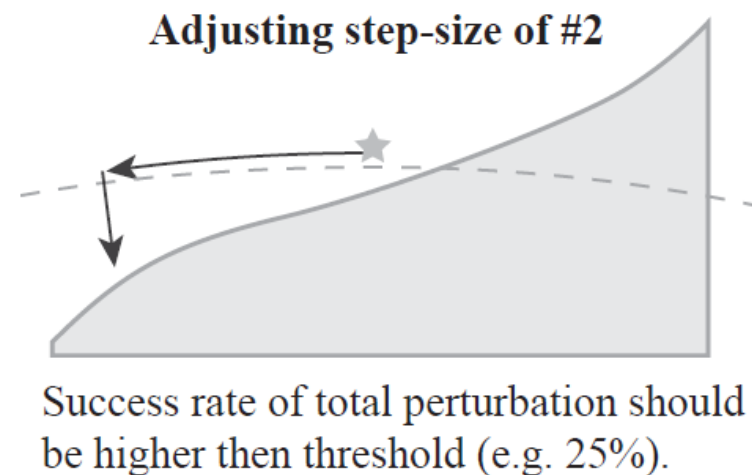
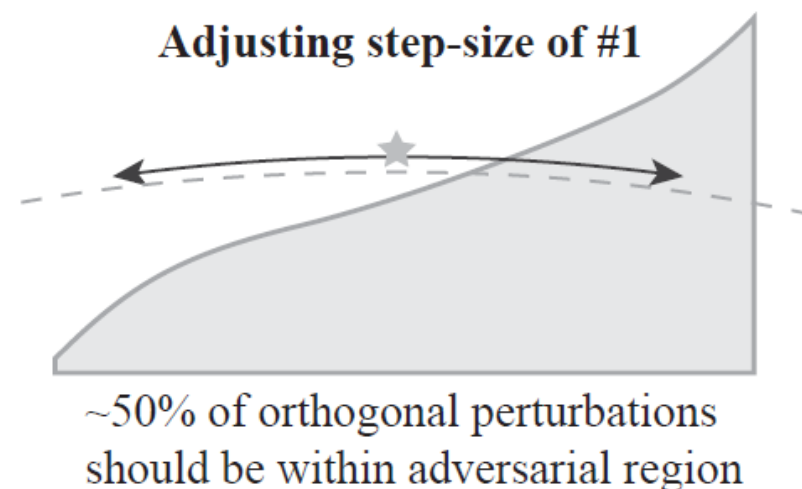
$$\|\eta^k\|_2 = \delta \cdot d(\mathbf{o}, \tilde{\mathbf{o}}^{k-1})$$

- So that $d(\mathbf{o}, \tilde{\mathbf{o}}^{k-1} + \eta^k) = d(\mathbf{o}, \tilde{\mathbf{o}}^{k-1})$
- In the last step, a small movement ϵ (#2 step in the image) is made toward the original image



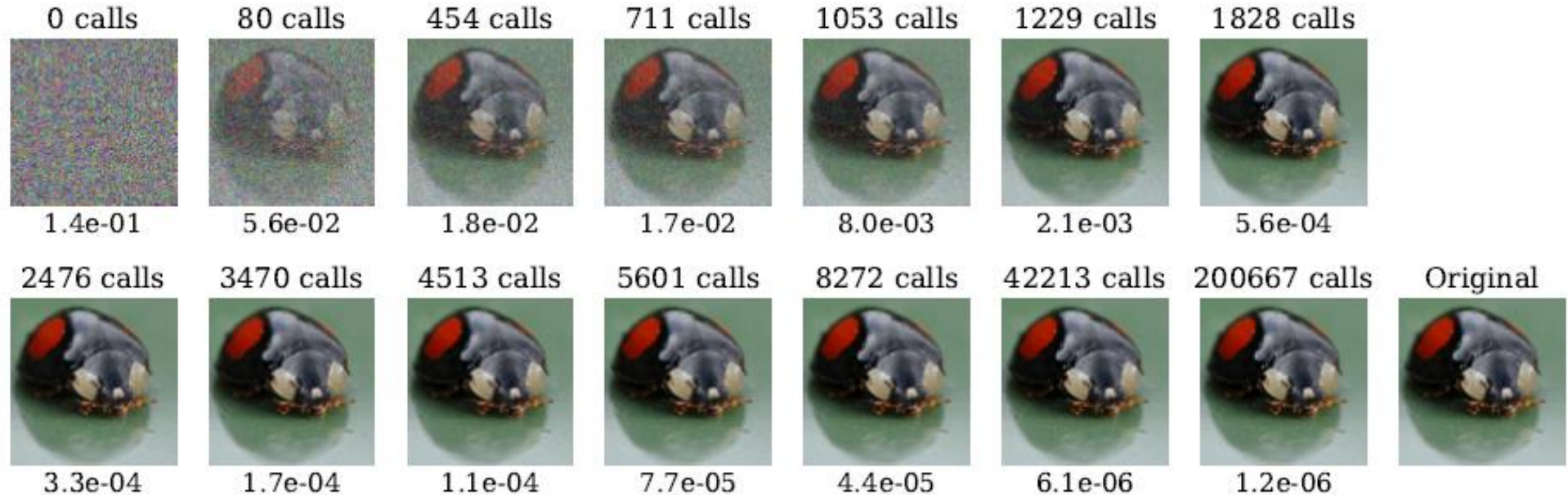
Brendel Paper (Boundary Attack)

- The two parameters δ (random orthogonal step) and ϵ (step toward the original image) are adjusted dynamically
- The parameter δ is adjusted so that about 50% of the perturbations are adversarial
 - If this ratio is much lower than 50%, the step size δ is reduced
 - In the opposite case, δ is increased
- Next, a small step ϵ toward the original image is applied
 - If the success rate is too small, ϵ is decreased
 - If it is too large, ϵ is increased
- The attack is converged whenever ϵ converges to zero



Brendel Paper (Boundary Attack)

- Example of an untargeted attack
 - From upper left to the lower right image
 - Above: total number of calls, i.e., predictions
 - Below: L_2 distance between the attacked image and the original image



Brendel Paper (Boundary Attack)

- Example of a targeted attack
 - Original class: tiger cat
 - Target class: Dalmatian dog
- Goal: create an adversarial image that is perceptually close (in L_2 distance) to a given image of a tiger cat, but is classified as a Dalmatian dog
 - The algorithm is initialized from a sample image of the target class that is correctly classified by the model



Brendel Paper (Boundary Attack)

- Comparison to FGSM, DeepFool, and Carlini-Wagner untargeted attacks
 - Presented values: median L_2 distance to the original images

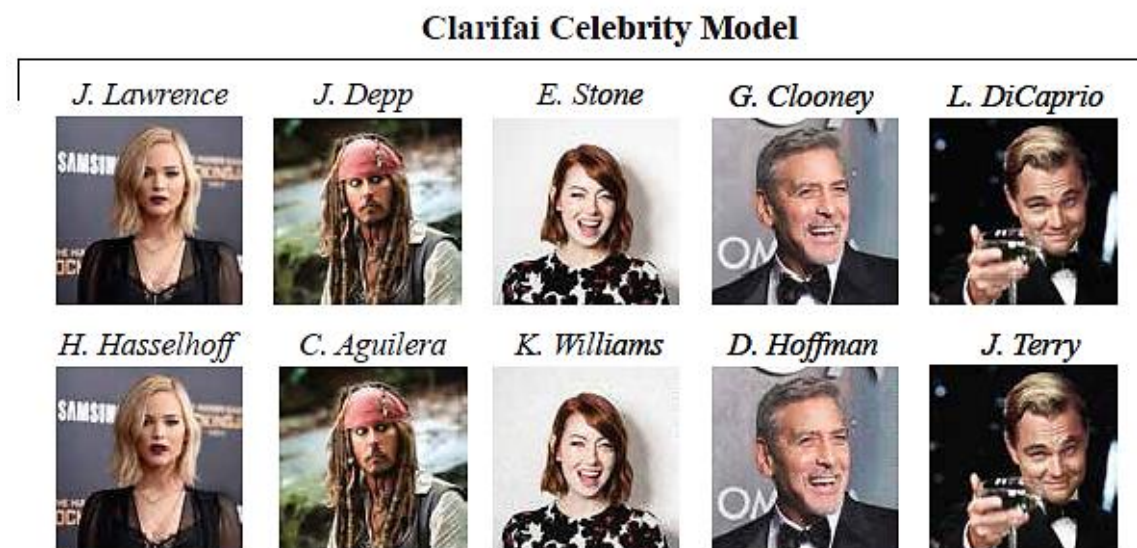
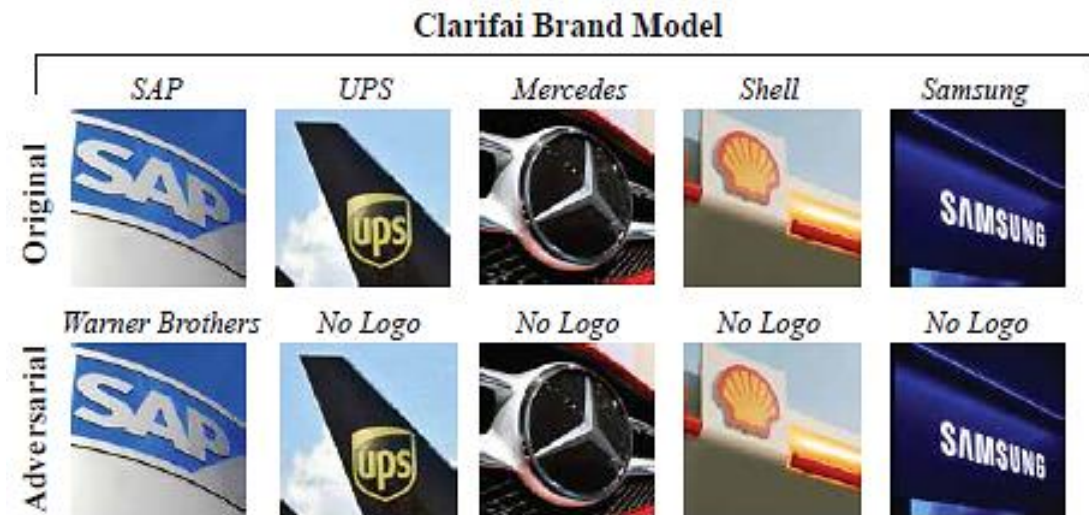
	Attack Type	MNIST	CIFAR	ImageNet		
				VGG-19	ResNet-50	Inception-v3
FGSM	gradient-based	4.2e-02	2.5e-05	1.0e-06	1.0e-06	9.7e-07
DeepFool	gradient-based	4.3e-03	5.8e-06	1.9e-07	7.5e-08	5.2e-08
Carlini & Wagner	gradient-based	2.2e-03	7.5e-06	5.7e-07	2.2e-07	7.6e-08
Boundary (ours)	decision-based	3.6e-03	5.6e-06	2.9e-07	1.0e-07	6.5e-08

- Comparison to Carlini-Wagner targeted attack

	Attack Type	MNIST	CIFAR	VGG-19
Carlini & Wagner	gradient-based	4.8e-03	3.0e-05	5.7e-06
Boundary (ours)	decision-based	6.5e-03	3.3e-05	9.9e-06

Brendel Paper (Boundary Attack)

- In many real-world applications, the attacker has no access to the architecture or the training data, but can only observe the final decision
 - E.g., security systems (face identification), autonomous cars, speech recognition (Alexa, Cortana)
- The authors applied Boundary Attack to two models by [Clarifai](#)
 - For identifying over 500 brand names in natural images
 - For identifying over 10,000 celebrities



Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- *Bhagoji, He, Li, Song (2017) Exploring the Space of Black-box Attacks on Deep Neural Networks*
- The paper introduces an approach known as **Gradient Estimation attack**
- Score-based black-box attack
 - Based on query access to the model's class probabilities
 - Both targeted and untargeted attacks are considered
- Validated on MNIST and CIFAR-10 datasets
 - Also evaluated on real-world models hosted by Clarifai
- Advantages:
 - Outperformed other black-box attacks
 - Performance results are comparable to white-box attacks
 - Good results against adversarial defenses

Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- Gradient Estimation approach
 - Use queries to directly estimate the gradient and carry out black-box attacks
 - The output to a query is the vector of class probabilities $\mathbf{p}^f(\mathbf{x})$ (i.e., confidence scores per class)
 - The logits can also be recovered from the probabilities, by taking $\log(\mathbf{p}^f(\mathbf{x}))$
- The authors employed the **method of finite differences** for gradient estimation
 - Let $g(\mathbf{x})$ is a function whose gradient needs to be estimated
 - Finite difference (FD) estimation of the gradient of g with respect to input \mathbf{x} is given by

$$\text{FD}_{\mathbf{x}}(g(\mathbf{x}), \delta) = \begin{bmatrix} \frac{g(\mathbf{x} + \delta \mathbf{e}_1) - g(\mathbf{x} - \delta \mathbf{e}_1)}{2\delta} \\ \vdots \\ \frac{g(\mathbf{x} + \delta \mathbf{e}_d) - g(\mathbf{x} - \delta \mathbf{e}_d)}{2\delta} \end{bmatrix}$$

- δ is a parameter that controls the estimation accuracy (selected 0.01 or 1)
- \mathbf{e}_i are basis vectors such that \mathbf{e}_i is 1 only for the i^{th} component and 0 everywhere else
- If the gradient exists, then $\lim_{\delta \rightarrow 0} \text{FD}_{\mathbf{x}}(g(\mathbf{x}), \delta) = \nabla_{\mathbf{x}} g(\mathbf{x})$

Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- **Approximate FGSM attack** with finite difference GE method

- Gradient of a model f is taken with respect to the cross-entropy loss $\ell_f(\mathbf{x}, y)$
 - For input \mathbf{x} with true class label y , the loss is

$$\ell_f(\mathbf{x}, y) = - \sum_{j=1}^{|\mathcal{Y}|} \mathbf{1}[j = y] \log p_j^f(\mathbf{x}) = - \log p_y^f(\mathbf{x})$$

- Recall that the derivative of a log function is $\frac{d}{dx} \log(x) = \frac{1}{x}$ and thus $\frac{d}{dx} \log(h(x)) = \frac{h'(x)}{h(x)}$
- Therefore, the gradient of the loss function $\ell_f(\mathbf{x}, y)$ with respect to the input \mathbf{x} is

$$\nabla_{\mathbf{x}} \ell_f(\mathbf{x}, y) = - \frac{\nabla_{\mathbf{x}} p_y^f(\mathbf{x})}{p_y^f(\mathbf{x})}$$

- An untargeted FGSM adversarial sample can be generated by using the FD estimate of the gradient $\nabla_{\mathbf{x}} p_y^f(\mathbf{x})$, i.e.,

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \epsilon \cdot \text{sign} \left(\frac{\text{FD}_{\mathbf{x}}(p_y^f(\mathbf{x}), \delta)}{p_y^f(\mathbf{x})} \right)$$

- Similarly, a targeted FGSM adversarial sample with class T can be found by using

$$\mathbf{x}_{\text{adv}} = \mathbf{x} - \epsilon \cdot \text{sign} \left(\frac{\text{FD}_{\mathbf{x}}(p_T^f(\mathbf{x}), \delta)}{p_T^f(\mathbf{x})} \right)$$

Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- **Approximate C-W attack** with finite difference GE method

- Carlini-Wagner attack uses a loss function based on the logits values $\phi(\cdot)$

$$\ell(\mathbf{x}, y) = \max(\phi(\mathbf{x} + \delta)_y - \max\{\phi(\mathbf{x} + \delta)_i : i \neq y\}, -\kappa).$$

- Logits values $\phi(\cdot)$ can be computed by taking the logarithm of the softmax probabilities, up to an additive constant
- For an untargeted C-W attack, the loss is the difference between the logits for the true class y and the second-most-likely class y' , i.e., $\phi(\mathbf{x} + \delta)_y - \phi(\mathbf{x} + \delta)_{y'}$
 - Since the loss is the difference of logits, the additive constant is canceled
 - By using FD approximation of the gradient, it is obtained

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}}(\phi(\mathbf{x})_{y'} - \phi(\mathbf{x})_y))$$

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \epsilon \cdot \text{sign}(\text{FD}_{\mathbf{x}}(\phi(\mathbf{x})_{y'} - \phi(\mathbf{x})_y, \delta))$$

- For a targeted C-W attack, the adversarial sample is

$$\mathbf{x}_{\text{adv}} = \mathbf{x} - \epsilon \cdot \text{sign}(\text{FD}_{\mathbf{x}}(\max(\phi(\mathbf{x})_i : i \neq T) - \phi(\mathbf{x})_T, \delta))$$

Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- **Iterative FGSM attack** with finite difference GE method
 - This is similar to the Basic Iterative Method and Projected Gradient Descent attacks, which use several iterations of the FGSM attack and achieve higher success rate than the single step FGSM attack
 - An iterative FD attack with $t + 1$ iterations using the cross-entropy loss is

$$\mathbf{x}_{\text{adv}}^{t+1} = \Pi_{\mathcal{H}} \left(\mathbf{x}_{\text{adv}}^t + \alpha \cdot \text{sign} \left(\frac{\text{FD}_{\mathbf{x}_{\text{adv}}^t} p_y^f(\mathbf{x}_{\text{adv}}^t)}{p_y^f(\mathbf{x}_{\text{adv}}^t)} \right) \right)$$

- **Iterative C-W attack** is also applied in a similar manner by modifying the single-step approach presented on the previous page

Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- Validation of untargeted black-box attacks using Gradient Estimation with FD
 - The table presents the success rate and average distortion (in parenthesis)
 - Baseline methods:
 - D. of M. – Difference of Means attack, uses the mean difference between the true class and the target class as added perturbation
 - Rand. – Random perturbation by adding random noise from a distribution (e.g., Gaussian)
 - ‘xent’ is for cross-entropy loss, ‘logit’ is C-W logits loss, ‘I’ is iterative
 - MNIST with L_∞ constraint of $\epsilon = 0.3$, and CIFAR-10 with L_∞ constraint of $\epsilon = 8$
 - Iterative C-W attack produced best results

MNIST	Baseline		Gradient Estimation using Finite Differences				Transfer from Model B			
Model	D. of M.	Rand.	Single-step		Iterative		Single-step		Iterative	
			FD-xent	FD-logit	IFD-xent	IFD-logit	FGS-xent	FGS-logit	IFGS-xent	IFGS-logit
A	44.8 (5.6)	8.5 (6.1)	51.6 (3.3)	92.9 (6.1)	75.0 (3.6)	100.0 (2.1)	66.3 (6.2)	80.8 (6.3)	89.8 (4.75)	88.5 (4.75)
B	81.5 (5.6)	7.8 (6.1)	69.2 (4.5)	98.9 (6.3)	86.7 (3.9)	100.0 (1.6)	-	-	-	-
C	20.2 (5.6)	4.1 (6.1)	60.5 (3.8)	86.1 (6.2)	80.2 (4.5)	100.0 (2.2)	49.5 (6.2)	57.0 (6.3)	79.5 (4.75)	78.7 (4.75)
D	97.1 (5.6)	38.5 (6.1)	95.4 (5.8)	100.0 (6.1)	98.4 (5.4)	100.0 (1.2)	76.3 (6.2)	87.6 (6.3)	73.3 (4.75)	71.4 (4.75)
CIFAR-10	Baseline		Gradient Estimation using Finite Differences				Transfer from Resnet-28-10			
Model	D. of M.	Rand.	Single-step		Iterative		Single-step		Iterative	
			FD-xent	FD-logit	IFD-xent	IFD-logit	FGS-xent	FGS-logit	IFGS-xent	IFGS-logit
Resnet-32	9.3 (440.5)	19.4 (439.4)	49.1 (217.1)	86.0 (410.3)	62.0 (149.9)	100.0 (65.7)	74.5 (439.4)	76.6 (439.4)	99.0 (275.4)	98.9 (275.6)
Resnet-28-10	6.7 (440.5)	17.1 (439.4)	50.1 (214.8)	88.2 (421.6)	46.0 (120.4)	100.0 (74.9)	-	-	-	-
Std.-CNN	20.3 (440.5)	22.2 (439.4)	80.0 (341.3)	98.9 (360.9)	66.0 (202.5)	100.0 (79.9)	37.4 (439.4)	37.7 (439.4)	33.7 (275.4)	33.6 (275.6)

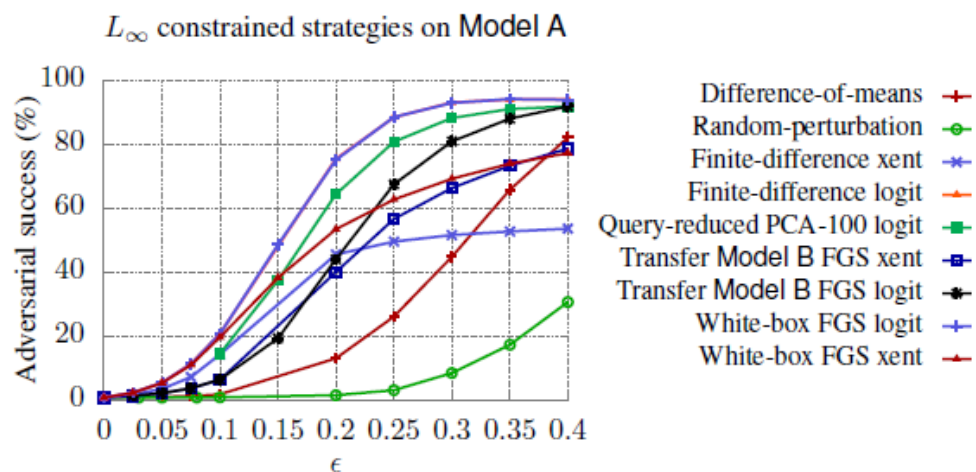
Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- Validation of targeted black-box attacks using Gradient Estimation with FD
 - Iterative FGSM attack produced best results on MNIST
 - Iterative C-W attack produced best results on CIFAR-10

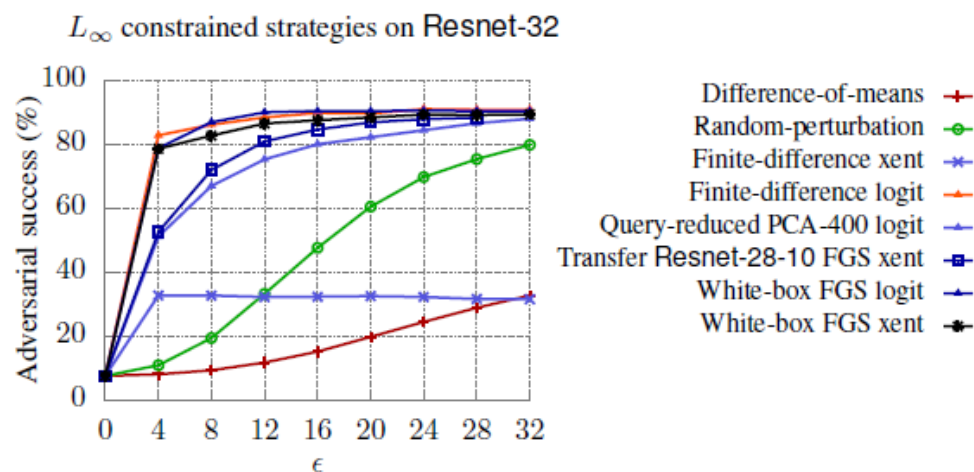
MNIST	Baseline	Gradient Estimation using Finite Differences				Transfer from Model B			
Model	D. of M.	Single-step		Iterative		Single-step		Iterative	
		FD-xent	FD-logit	IFD-xent	IFD-logit	FGS-xent	FGS-logit	IFGS-xent	IFGS-logit
A	15.0 (5.6)	30.0 (6.0)	29.9 (6.1)	100.0 (4.2)	99.7 (2.7)	18.3 (6.3)	18.1 (6.3)	54.5 (4.6)	46.5 (4.2)
B	35.5 (5.6)	29.5 (6.3)	29.3 (6.3)	99.9 (4.1)	98.7 (2.4)	-	-	-	-
C	5.84 (5.6)	34.1 (6.1)	33.8 (6.4)	100.0 (4.3)	99.8 (3.0)	14.0 (6.3)	13.8 (6.3)	34.0 (4.6)	26.1 (4.2)
D	59.8 (5.6)	61.4 (6.3)	60.8 (6.3)	100.0 (3.7)	99.9 (1.9)	16.8 (6.3)	16.7 (6.3)	36.4 (4.6)	32.8 (4.1)
CIFAR-10	Baseline	Gradient Estimation using Finite Differences				Transfer from Resnet-28-10			
Model	D. of M.	Single-step		Iterative		Single-step		Iterative	
		FD-xent	FD-logit	IFD-xent	IFD-logit	FGS-xent	FGS-logit	IFGS-xent	IFGS-logit
Resnet-32	1.2 (440.3)	23.8 (439.5)	23.0 (437.0)	100.0 (110.9)	100.0 (89.5)	15.8 (439.4)	15.5 (439.4)	71.8 (222.5)	80.3 (242.6)
Resnet-28-10	0.9 (440.3)	29.2 (439.4)	28.0 (436.1)	100.0 (123.2)	100.0 (98.3)	-	-	-	-
Std.-CNN	2.6 (440.3)	44.5 (439.5)	40.3 (434.9)	99.0 (178.8)	95.0 (126.8)	5.6 (439.4)	5.6 (439.4)	5.1 (222.5)	5.9 (242.6)

Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- Graphs of the success rate versus the perturbation size ϵ
 - The proposed black-box attack has almost the same curve as white-box C-W attack (e.g., 'White-box FGS logit')



(a) Model A (MNIST)



(b) Resnet-32 (CIFAR-10)

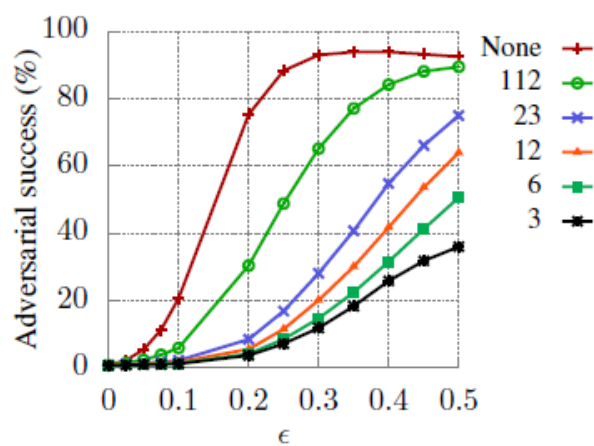
Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- Shortcoming of the proposed approach:
 - Requires $O(d)$ queries per input, where d is the dimension of the input
 - The presented FD approximation required $2 \cdot d$ queries
- The authors propose two approaches for reducing the number of queries
 - Random grouping
 - The gradient is estimated only for a random group of selected features
 - PCA (Principal Component Analysis)
 - Compute the gradient only along a number of principal component vectors

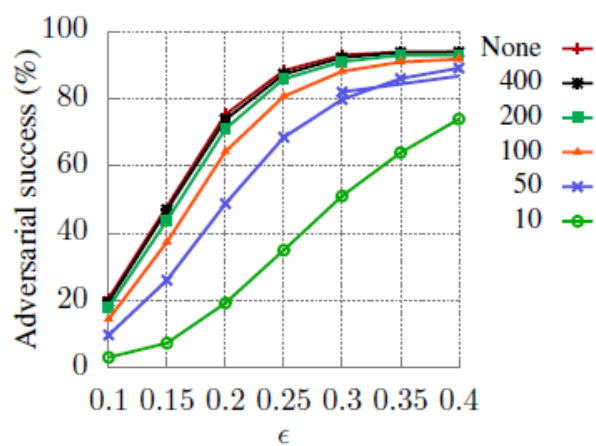
Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- Validation of the methods for query reduction
 - For random grouping, the success rate decreases with decreasing the group size
 - For PCA, the success rate is still high as the number of PC decreases

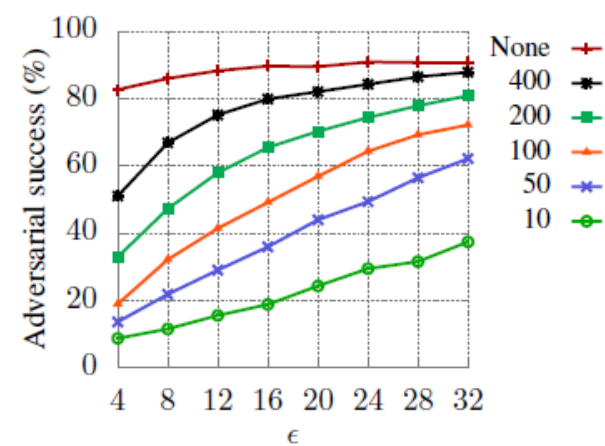
Random feature groupings for Model A



PCA-based query reduction for Model A

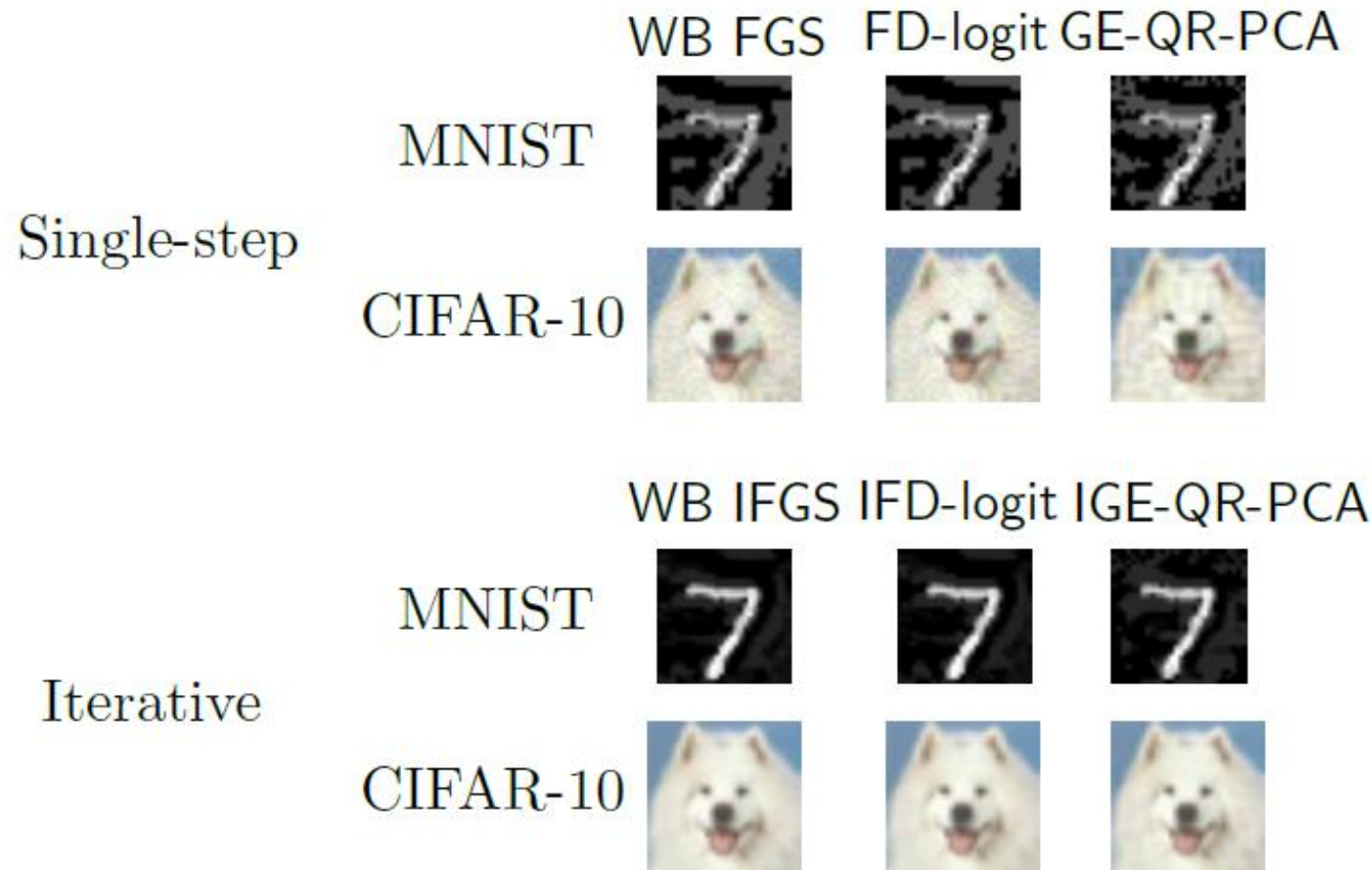


PCA-based query reduction for Resnet-32



Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- Untargeted adversarial samples
 - GE-QR-PCA stands for Gradient Estimation with Query Reduction using PCA



Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- Evaluation against adversarial defenses
 - Adversarial training (Szagedy et al, 2014)
 - Ensemble adversarial training (Tramer et al, 2017)
 - Iterative adversarial training (Madry et al, 2017)
- The accuracy is almost the same as for benign non-attacked images

Dataset (Model)	Benign	Adv	Adv-Ens	Adv-Iter
MNIST (A)	99.2	99.4	99.2	99.3
CIFAR-10 (Resnet-32)	92.4	92.1	91.7	79.1

Bhagoji, Li, Song Paper (Gradient Estimation Attack)

- Attacks on two real-world models hosted by Clarifai
 - Not Safe For Work (NSFW)
 - Two categories: 'safe', 'not safe'
 - Content Moderation
 - Five categories: 'safe', 'suggestive', 'explicit', 'drug,' and 'gore'
 - Example: an adversary could upload violent adversarially-modified images, which may be marked incorrectly as 'safe' by the Content Moderation model



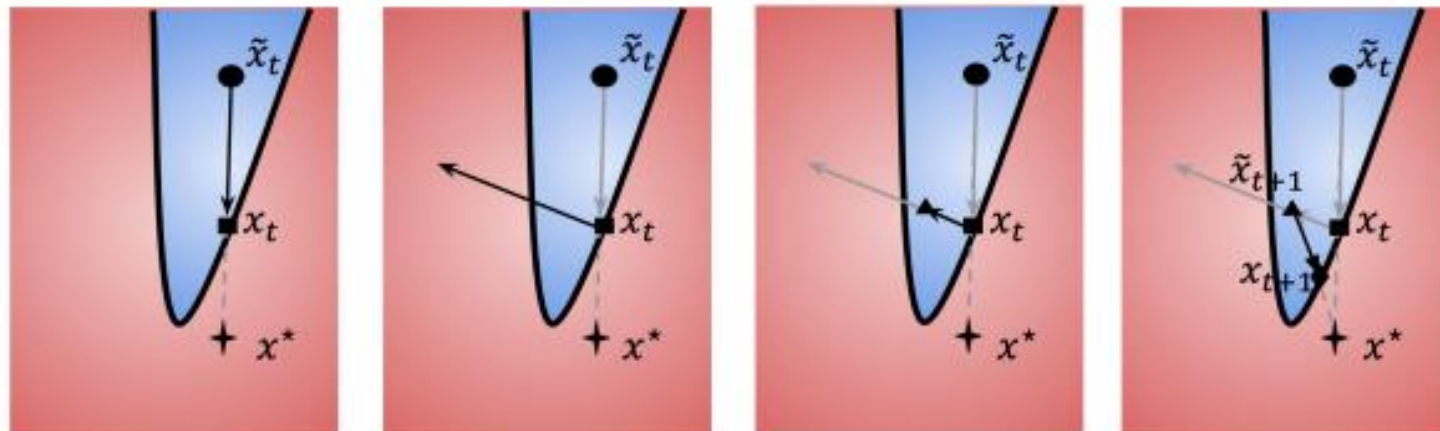
Original image
Class: 'drug'
Confidence: 0.99



Adversarial image
Class: 'safe'
Confidence: 0.96

Other Black-box Evasion Attacks

- *HopSkipJumpAttack*
 - [Chen and Jordan \(2019\) Boundary attack++: Query-efficient decision-based adversarial attack](#)
- The attack is an extension of the Boundary Attack
 - Requires significantly fewer queries than Boundary Attack
 - It includes both untargeted and targeted attacks
- HopSkipJumpAttack is based on a novel approach for estimation of the gradient direction along the decision boundary
 - Perform a binary search to find the boundary, estimate the gradient direction at the boundary point, and update until the closest sample to the original sample x^* is found



Other Black-box Evasion Attacks

- *ZOO attack*
 - [Chen \(2017\) Zoo: Zeroth-order optimization based black-box attacks to deep neural networks without training substitute models](#)
- **Zeroth-order optimization** refers to optimization based on access to the function values $f(x)$ only (as opposed to first-order optimization via the gradient $\nabla f(x)$)
 - E.g., score-based and decision-based black-box approaches
- ZOO attack is a score-based version of the Carlini-Wagner attack
 - The gradient is estimated based on logits values
 - It employs a zeroth-order stochastic coordinate descent
 - At each iteration, one randomly-selected variable (coordinate) is updated with the goal to optimize the objective function

Semi-white Box (Grey-box) Attacks

- *AdvGAN*
 - [Xiao et al. \(2018\) Generating adversarial examples with adversarial networks](#)
- A GAN (Generative Adversarial Network) is trained and used to generate adversarial examples
- Semi-white box attack
 - It uses the original target classifier model to train a GAN model
 - Afterwards, it does not need access to the target model to generate adversarial perturbations for other input examples
- Advantages:
 - Fastest generation of adversarial examples
 - Naturally looking samples, difficult to detect

List of Adversarial Attacks

Attack	Publication	Similarity	Attacking Capability	Algorithm	Apply Domain
L-BFGS	(Szegedy et al., 2013)	l_2	White-Box	Iterative	Image Classification
FGSM	(Goodfellow et al., 2014b)	l_∞, l_2	White-Box	Single-Step	Image Classification
Deepfool	(Moosavi-Dezfooli et al., 2016)	l_2	White-Box	Iterative	Image Classification
JSMA	(Papernot et al., 2016a)	l_2	White-Box	Iterative	Image Classification
BIM	(Kurakin et al., 2016a)	l_∞	White-Box	Iterative	Image Classification
C & W	(Carlini & Wagner, 2017b)	l_2	White-Box	Iterative	Image Classification
Ground Truth	(Carlini et al., 2017)	l_0	White-Box	SMT solver	Image Classification
Spatial	(Xiao et al., 2018b)	Total Variation	White-Box	Iterative	Image Classification
Universal	(Metzen et al., 2017b)	l_∞, l_2	White-Box	Iterative	Image Classification
One-Pixel	(Su et al., 2019)	l_0	White-Box	Iterative	Image Classification
EAD	(Chen et al., 2018)	$l_1 + l_2, l_2$	White-Box	Iterative	Image Classification
Substitute	(Papernot et al., 2017)	l_p	Black-Box	Iterative	Image Classification
ZOO	(Chen et al., 2017)	l_p	Black-Box	Iterative	Image Classification
Biggio	(Biggio et al., 2012)	l_2	Poisoning	Iterative	Image Classification
Explanation	(Koh & Liang, 2017)	l_p	Poisoning	Iterative	Image Classification
Zugner's	(Zügner et al., 2018)	Degree Distribution, Cooccurrence	Poisoning	Greedy	Node Classification
Dai's	(Dai et al., 2018)	Edges	Black-Box	RL	Node & Graph Classification
Meta	(Zügner & Günnemann, 2019)	Edges	Black-Box	RL	Node Classification
C & W	(Carlini & Wagner, 2018)	max dB	White-Box	Iterative	Speech Recognition
Word Embedding	(Miyato et al., 2016)	l_p	White-Box	One-Step	Text Classification
HotFlip	(Ebrahimi et al., 2017)	letters	White-Box	Greedy	Text Classification
Jia & Liang	(Jia & Liang, 2017)	letters	Black-Box	Greedy	Reading Comprehension
Face Recognition	(Sharif et al., 2016)	physical	White-Box	Iterative	Face Recognition
RL attack	(Huang et al., 2017)	l_p	White-Box	RL	

Additional References

1. Nicolae et al. (2019) - Adversarial Robustness Toolbox v1.0.0.
<https://arxiv.org/abs/1807.01069>
2. Xu et al. (2019) - Adversarial Attacks and Defenses in Images, Graphs and Text: A Review <https://arxiv.org/abs/1909.08072>