University of Idaho

CS 404/504

Special Topics: Adversarial Machine Learning

Dr. Alex Vakanski

Lecture 6

Evasion Attacks against Black-box Machine Learning Models

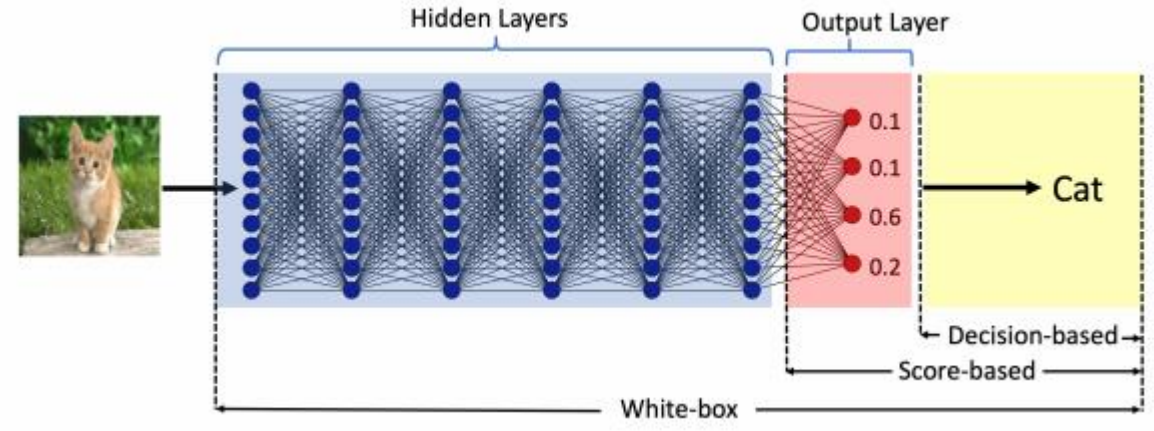
Lecture Outline

- 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
- Chen (2017) ZOO: Zeroth-order Optimization Based Black-box Attacks to Deep Neural Networks Without Training Substitute Models
- Transferability in Adversarial Machine Learning
  - Substitute model attack
  - Ensemble of local models attack
- Other black-box evasion attacks
  - Simple black-box attack
  - HopSkipJump attack

Evasion Attacks against Black-box Models

Black-box Evasion Attacks

- 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



Decision Based Adversarial Attacks: Reliable Attacks Against Black Box Machine Learning Models

Wieland Brendel, Jonas Rauber, Matthias Bethge

Background of Paper

- Authors are from the Werner Reichardt Centre for Integrative Neuroscience in Germany
- This was a conference paper at the International Conference on Learning Representations in 2018
- Focuses on black-box attacks rather than white-box attacks.
- Creates a black box attack they call a “Boundary” attack

Image: https://www.cin.uni-tuebingen.de/about-cin/background/the-cin-in-a-nutshell.html

Paper Motivation

- The authors discuss that most of the current attack methods require knowledge of some part of the model
  - Model parameters
  - Training Data
- The authors do not find this realistic in a real-world setting.

Paper Motivation

- Adversarial attacks are potentially harmful
  - Turn stop sign to 200 km/h sign
- Adversarial attacks are biologically interesting
  - Illustrate differences between human and machine perception

Image: www.pixabay.com

Paper Motivation

- “Robustness” of model hard to determine
- Authors look at some attacks that are easy to defend against
- Authors think models may not be “robust”, but instead have only been attacked weakly

Previous Attack Types

- Gradient-based
  - Uses the gradient of the loss function to find the perturbation
  - FGSM, BIM, DeepFool, JSMA, Houdini, Carlini & Wagner
- Score-based
  - Use the predicted class scores to try and estimate the gradient
  - Variants of JSMA, Carlini & Wagner, generator networks
- Transfer-based
  - Get training data information to train “substitute” models
  - Ensemble success rate

Previous Attack Defenses

- Gradient-based:
  - Masking gradients (defensive distillation, saturation non-linearities, non-differentiable classifiers)
- Score-based:
  - Add “stochastic” elements to make gradient estimate hard (paper mentions dropout)
- Transfer-based:
  - Training model with adversarial examples (data augmentation)

Note on Defensive Distillation

- Defensive Distillation [4]:
  - One model trained on normal data
  - Second model trained on the output probabilties of the first model
  - “uncertainty is used to train the second model as an additional filter” [4]
- It’s more difficult for an attacker to fool both of the models
- The authors of the paper look at this defense later in their attack

Decision-Based Attacks

- ONLY use final model's decision
- Only have access to 1 class (highest probability class)
  - Some real-world models return more
- More realistic/applicable to machine learning models, since it assumes you don't have access to the model parameters

Prior Decision-Based Attacks

- Authors note that there hasn’t been a ton of work in this area
- One previous transfer attack used a synthetic dataset
- Authors unsure this approach would scale to more complex datasets
- Other previous decision-based attacks have very noticeable perturbations

The Boundary Attack

- Assumptions
  - Attacker wants minimal perturbation to image
  - Attacker can see final model output
  - Already knows an adversarial perturbation for an image (can be a very large one)

The Boundary Attack

- Starts with an image that is already adversarial
- Traverses the area between the adversarial region and non adversarial region such that:
  - 1. it stays in the adversarial region, and
  - 2. the distance between the adversarial image and regular image is reduced.
- “Rejection sampling with suitable proposal distribution P”

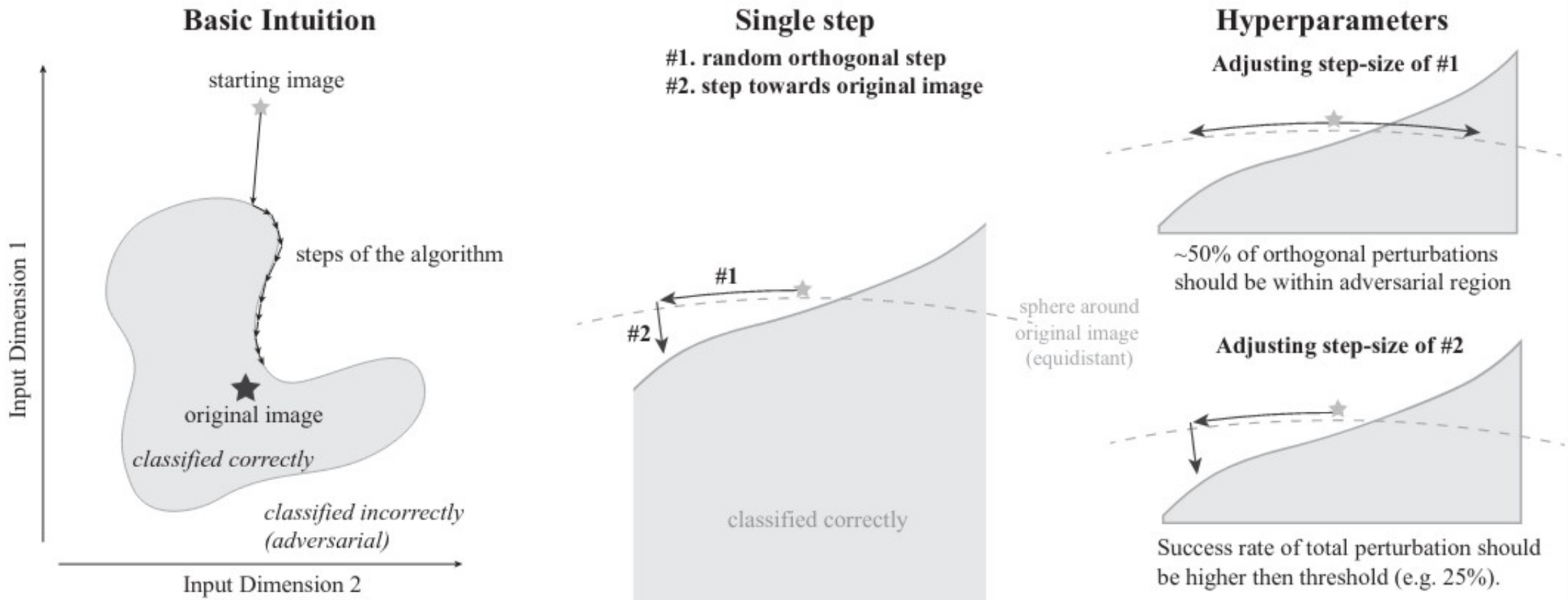
The Boundary Attack

- Terminology and basic algorithm:

Data: original image $\mathbf{o}$, adversarial criterion $c(.)$, decision of model $d(.)$
Result: adversarial example $\tilde{\boldsymbol{o}}$ such that the distance $d(\boldsymbol{o}, \tilde{\boldsymbol{o}}) = \|\boldsymbol{o} - \tilde{\boldsymbol{o}}\|_2^2$ is minimized
initialization: $k = 0$, $\tilde{\boldsymbol{o}}^0 \sim \mathcal{U}(0, 1)$ s.t. $\tilde{\boldsymbol{o}}^0$ is adversarial;
while $k <$ *maximum number of steps* **do**
  draw random perturbation from proposal distribution $\boldsymbol{\eta}_k \sim \mathcal{P}(\tilde{\boldsymbol{o}}^{k-1})$;
  if $\tilde{\boldsymbol{o}}^{k-1} + \boldsymbol{\eta}_k$ *is adversarial* **then**
    set $\tilde{\boldsymbol{o}}^k = \tilde{\boldsymbol{o}}^{k-1} + \boldsymbol{\eta}_k$;
  else
    set $\tilde{\boldsymbol{o}}^k = \tilde{\boldsymbol{o}}^{k-1}$;
  end
  $k = k + 1$
end

Algorithm 1: Minimal version of the Boundary Attack.

The Boundary Attack



The Boundary Attack

- The authors identified the proposal distribution as being a very important part of this process
- They narrow down the proposal distribution to the following criteria:

1. The perturbed sample lies within the input domain,

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

2. The perturbation has a relative size of δ,

$$\left\|\boldsymbol{\eta}^k\right\|_2 = \delta \cdot d(\mathbf{o}, \tilde{\mathbf{o}}^{\mathbf{k-1}}). \tag{2}$$

3. The perturbation reduces the distance of the perturbed image towards the original input by a relative amount ϵ,

$$d(\mathbf{o}, \tilde{\mathbf{o}}^{\mathbf{k-1}}) - d(\mathbf{o}, \tilde{\mathbf{o}}^{\mathbf{k-1}} + \boldsymbol{\eta}^k) = \epsilon \cdot d(\mathbf{o}, \tilde{\mathbf{o}}^{\mathbf{k-1}}). \tag{3}$$

The Boundary Attack

- Essentially, the parameters delta and epsilon control the way the algorithm changes the image
- Delta is the radius of a sphere that the perturbation must be within [2]
  - Delta is adjusted up/down to make sure that about 50% of perturbations are adversarial [2]
- Epsilon controls the step size toward the original image
  - it’s adjustment depends on the success rate
  - Converging epsilon to 0 (or maximizing the iterations) ends the attack. [2]

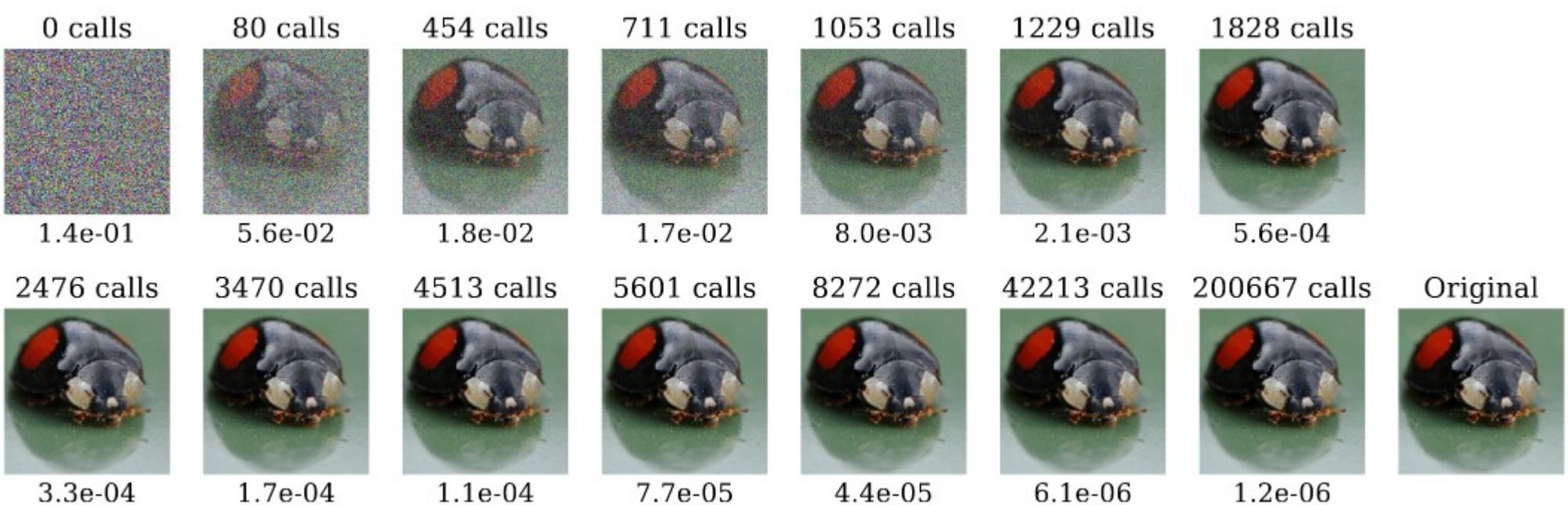
Boundary Attacks vs. Other Attacks

- Evaluated on both targeted and untargeted attacks
- For untargeted: compared with FGSM, DeepFool, and Carlini and Wagner

				ImageNet		
	Attack Type	MNIST	CIFAR	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

The Boundary Attack vs. Other Attacks

- This attack takes far more calls to the model than the other attacks (somewhat computationally intense)
- However – that is to minimize the distance between the original image and the adversarial image
- Needs less calls to make “imperceptible” perturbations



Model Calls Comparisons

- Average “passes” over 20 samples for ResNet50
- DeepFool: 7 forward, 37 backward
- Carlini and Wagner: 16,000 forward, 16,000 backward
- Boundary: 1,200,000 forward, 0 backward

The Boundary Attack vs. Other Attacks

- Compared with Carlini and Wagner
- Starts with an adversarial image (one from a different category)

	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

The Boundary Attack vs. Other Attacks

- Boundary Attack succeeded even with defensive distillation

		MNIST		CIFAR	
	Attack Type	standard	distilled	standard	distilled
FGSM	gradient-based	4.2e-02	fails	2.5e-05	fails
Boundary (ours)	decision-based	3.6e-03	4.2e-03	5.6e-06	1.3e-05

Real-World Model Boundary Attacks

- The authors also tested their attacks on two models from Clarifai
- Clarifai is a provides cloud machine learning models
- They used a brand recognition and celebrity dataset

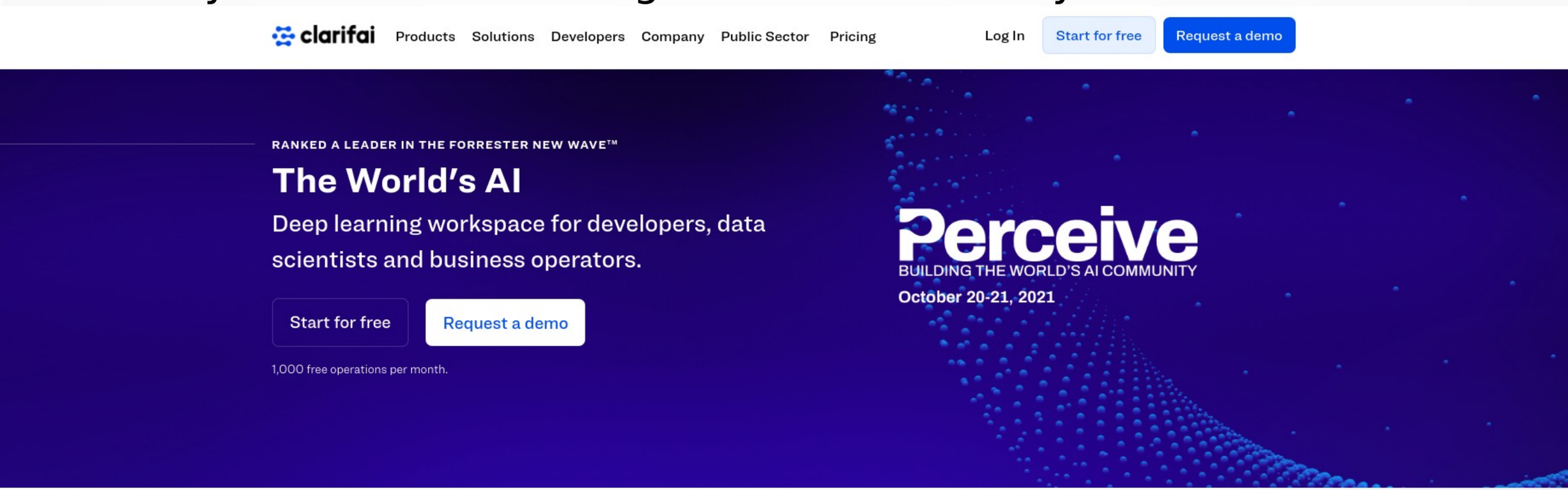
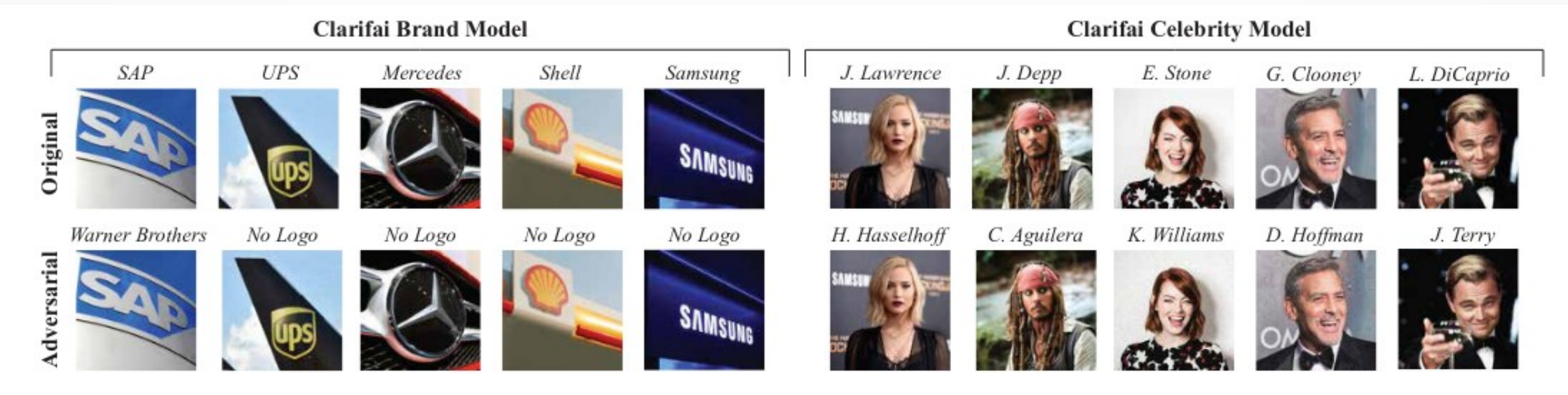


Image: www.clarifai.com

Real-World Model Boundary Attacks

- Attacks were successful
- Authors note it was more difficult to attack real-world models
- Sometimes noise is noticeable in the images



Paper Discussion Points

- The Boundary attack is important because it really tests model's 'robustness'
- When you are working only with the model output, you can get a good idea of what the model is susceptible to
- More realistic applications
- Illustrates differences between machine and human decision making

References

1. Wieland Brendel, Jonas Rauber, and Matthias Bethge. “Decision Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models”. International Conference on Learning Representations. 2018. https://arxiv.org/abs/1712.04248.

2. Aleksander Vakanski. “Lecture 6: Evasion Attacks Against Machine Learning Models”. University of Idaho, CS 502. 2020.

3. Roger D. Peng. “Rejection Sampling”. Advanced Statistical Computing, Section 6.2. 2021. https://bookdown.org/rdpeng/advstatcomp/rejection-sampling.html

4. “What is Defensive Distillation?”. Deep AI. https://deepai.org/machine-learning-glossary-and-terms/defensive-distillation

Questions?

Gradient Estimation Attack

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

Gradient Estimation Attack

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) for an input $\mathbf{x}$
    - The logits can also be recovered from the probabilities, by taking $\log\left(\mathbf{p}^f(\mathbf{x})\right)$
- 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

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

  - 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 $\lim_{\delta \to \mathbf{0}} \mathrm{FD}_{\mathbf{x}}(g(\mathbf{x}),\delta) = \nabla_{\mathbf{x}}\, g(\mathbf{x})$

Gradient Estimation Attack

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

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

    - 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

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

Gradient Estimation Attack

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., $\phi(x+\delta)_y - \phi(x+\delta)_{y'}$
    - 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

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

Gradient Estimation Attack

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

Experimental Validation

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**			
			Single-step		Iterative		Single-step		Iterative	
Model	D. of M.	Rand.	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**			
			Single-step		Iterative		Single-step		Iterative	
Model	D. of M.	Rand.	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)

Experimental Validation

Gradient Estimation Attack

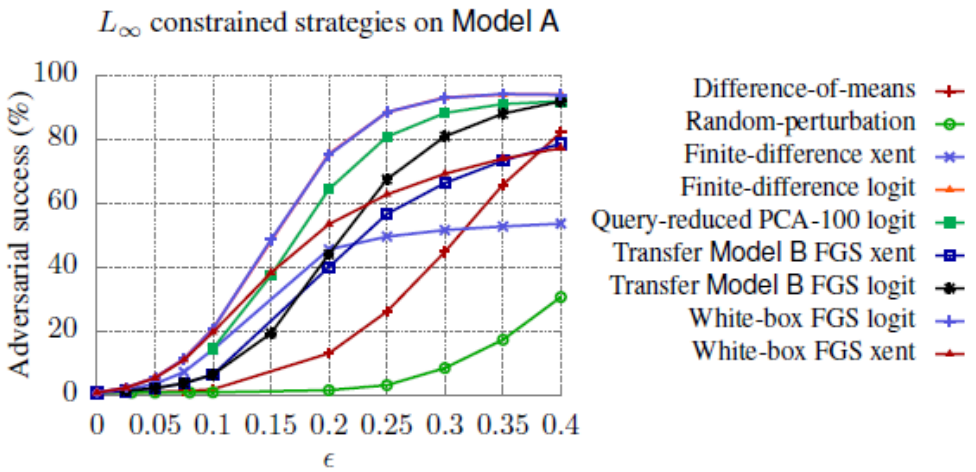
- Validation of targeted black-box attacks using Gradient Estimation with FD
  - Iterative FGSM attack produced best results on MNIST
  - Iterative C-W attack produced best results on CIFAR-10

MNIST	Baseline	Gradient Estimation using Finite Differences				Transfer from Model B			
		Single-step		Iterative		Single-step		Iterative	
Model	D. of M.	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**			
		Single-step		Iterative		Single-step		Iterative	
Model	D. of M.	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)

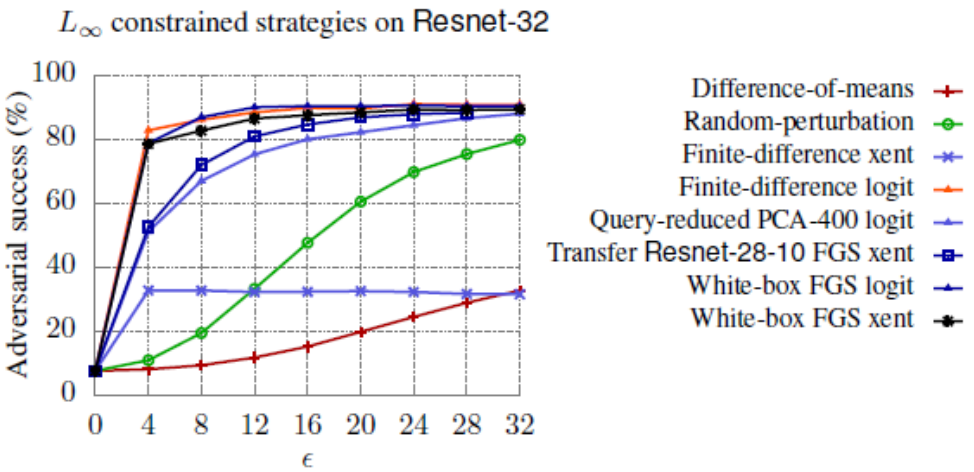
Experimental Validation

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)

Query Reduction

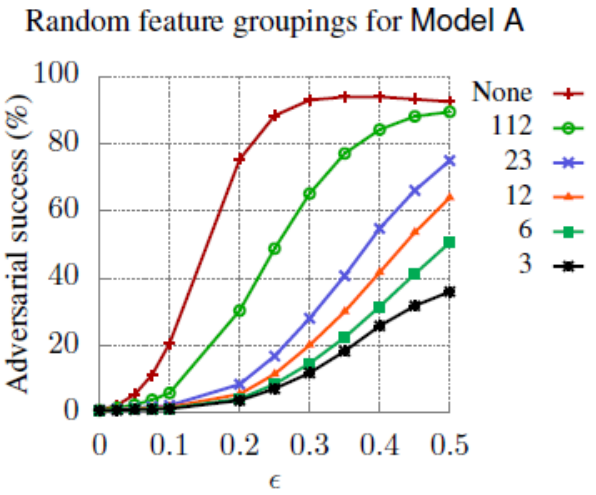
Gradient Estimation Attack

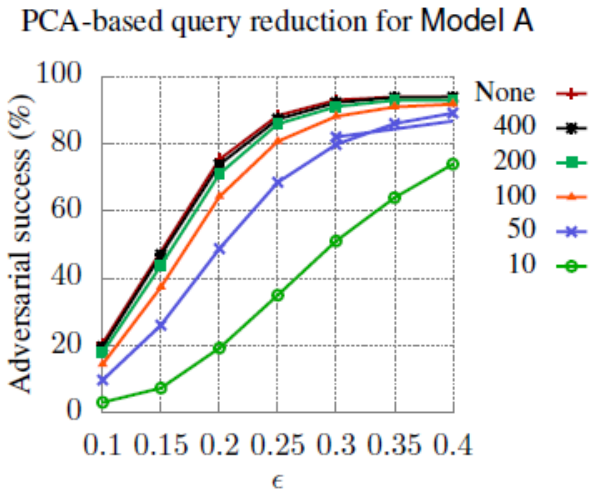
- Shortcoming of the proposed approach:
  - Requires $O(d)$ queries per input, where d is the dimension of the input
  - The presented FD approximation required $2 \cdot d$ queries
- The authors propose two approaches for reducing the number of queries
  - Random grouping
    - The gradient is estimated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

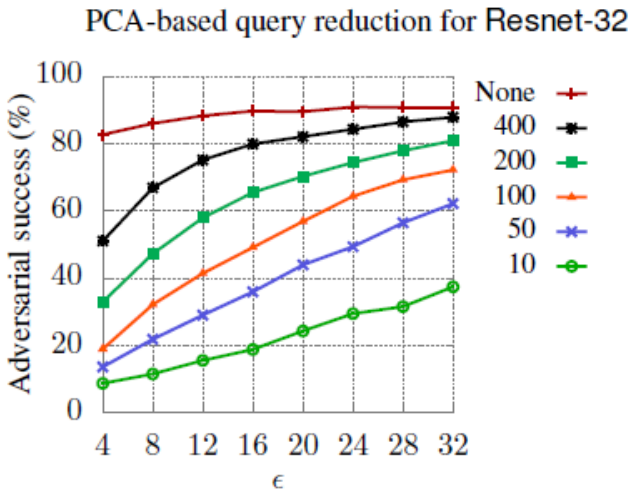
Query Reduction

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



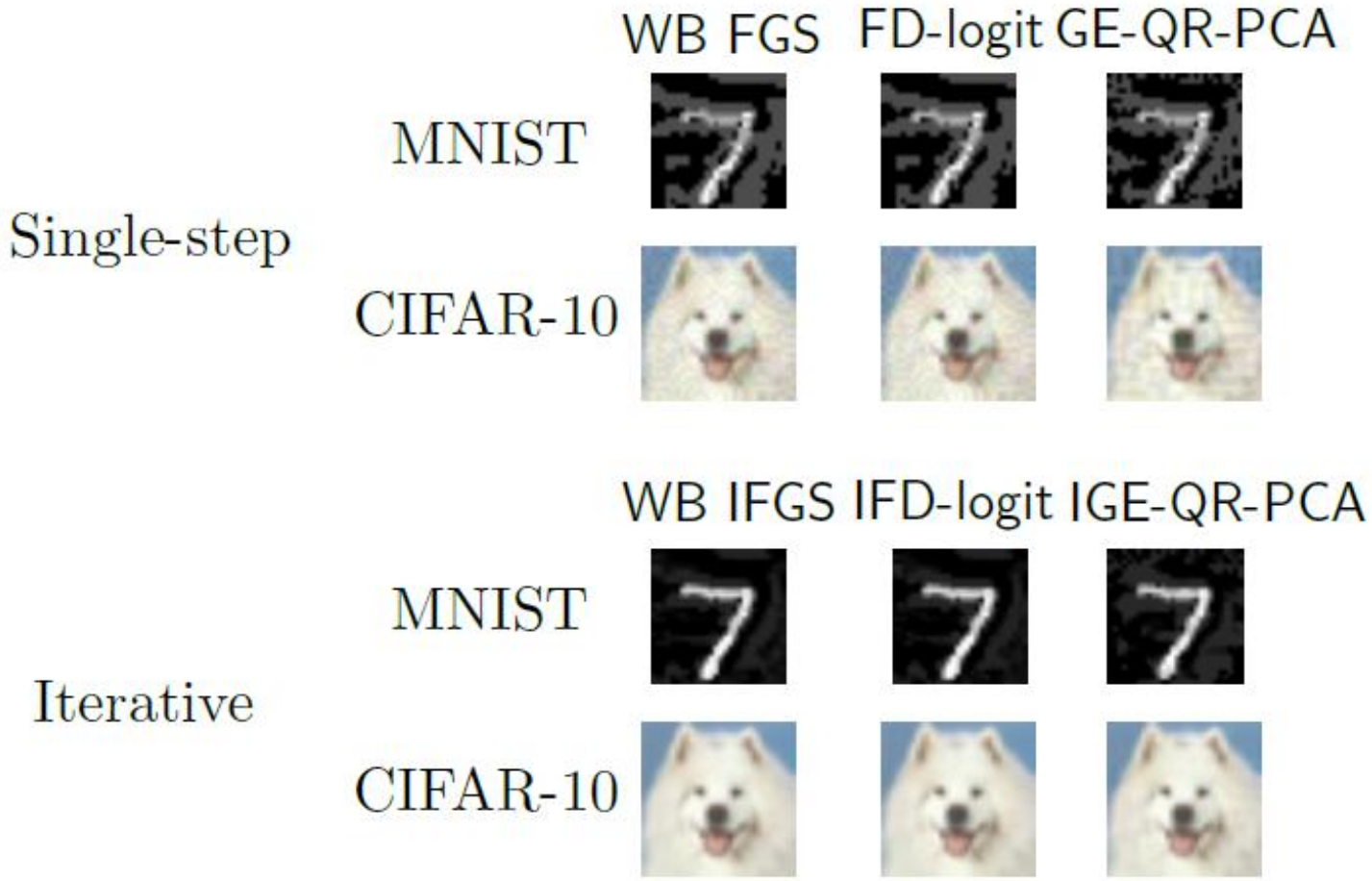




Adversarial Samples

Gradient Estimation Attack

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



Defense Evaluation

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

Attacks on Real Models

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

ZOO Attack

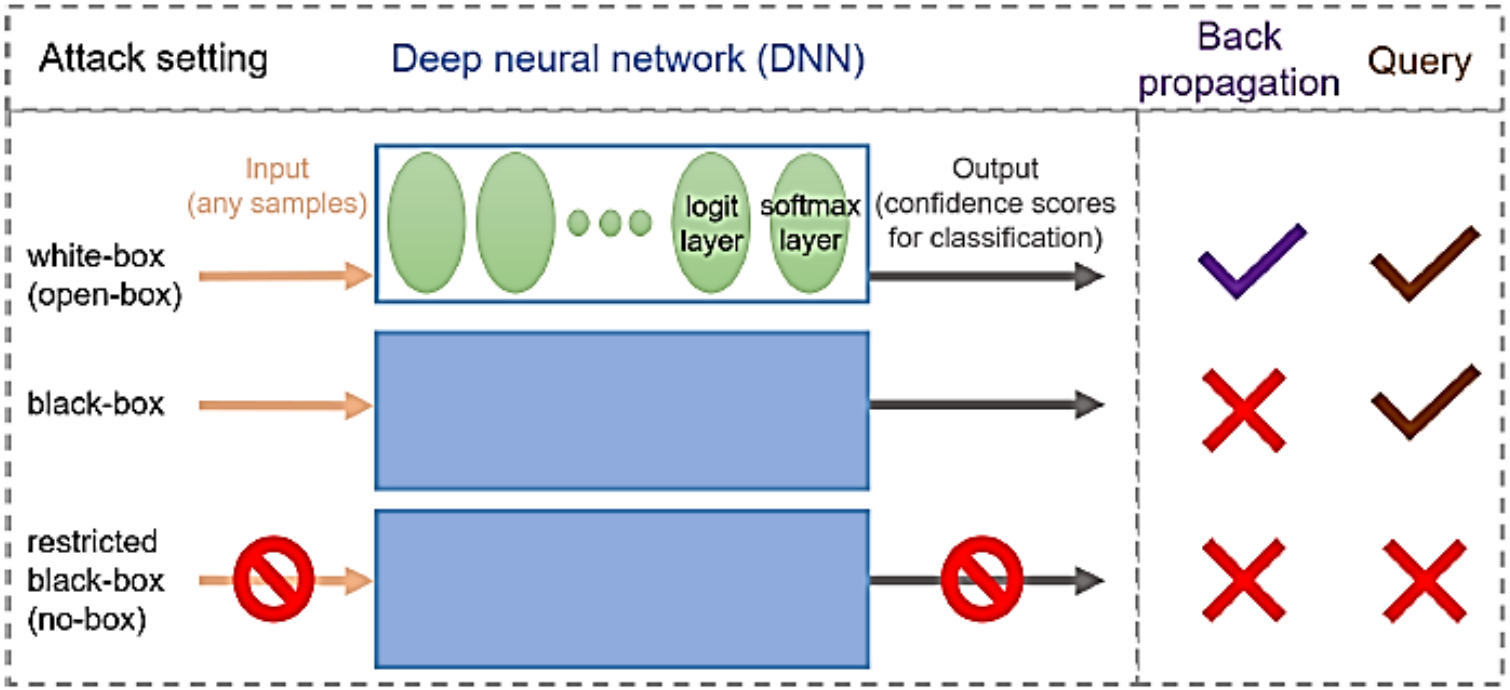
ZOO Attack

- ***ZOO attack***
  - Chen (2017) ZOO: Zeroth-order Optimization Based Black-box Attacks to Deep Neural Networks Without Training Substitute Models
- Zeroth-order optimization (ZOO) refers to optimization based on access to the predictions by the model $f(x)$ only
  - As opposed to first-order optimization via the gradient $\nabla f(x)$, as in white-box models
  - E.g., score-based and decision-based black-box approaches are zeroth-order optimization approaches
- ZOO attack is a score-based attack
  - It is very similar to the Gradient Estimation attack by Bhagoji et al. (2017)

ZOO Attack

ZOO Attack

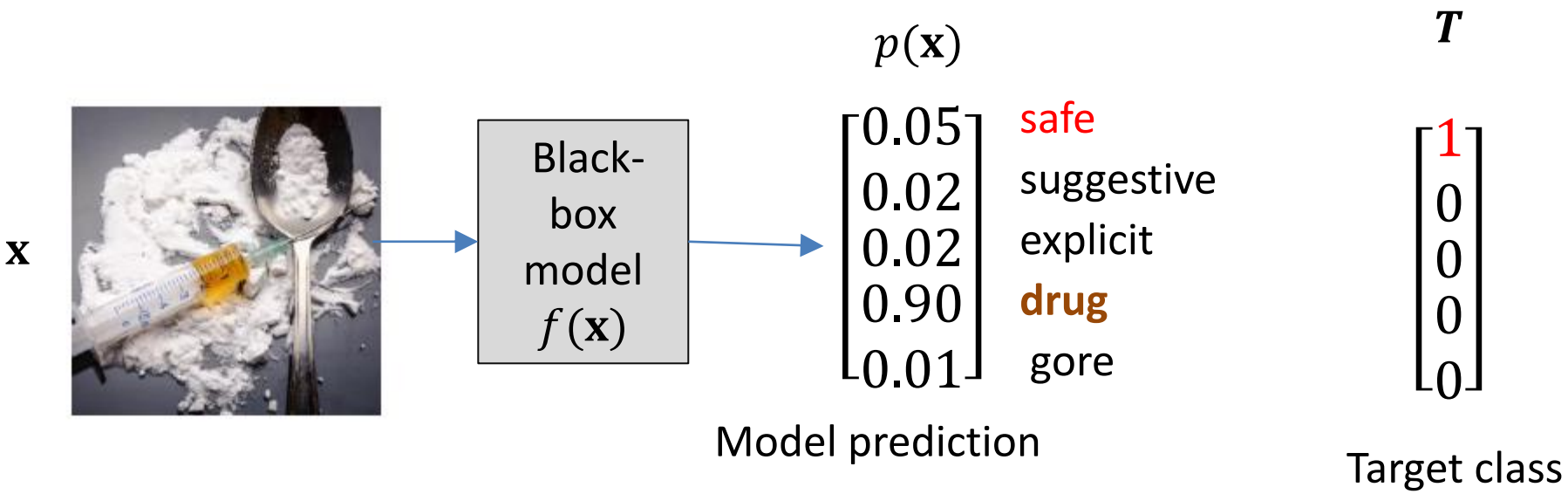
- Taxonomy of AML attacks



Loss function

ZOO Attack

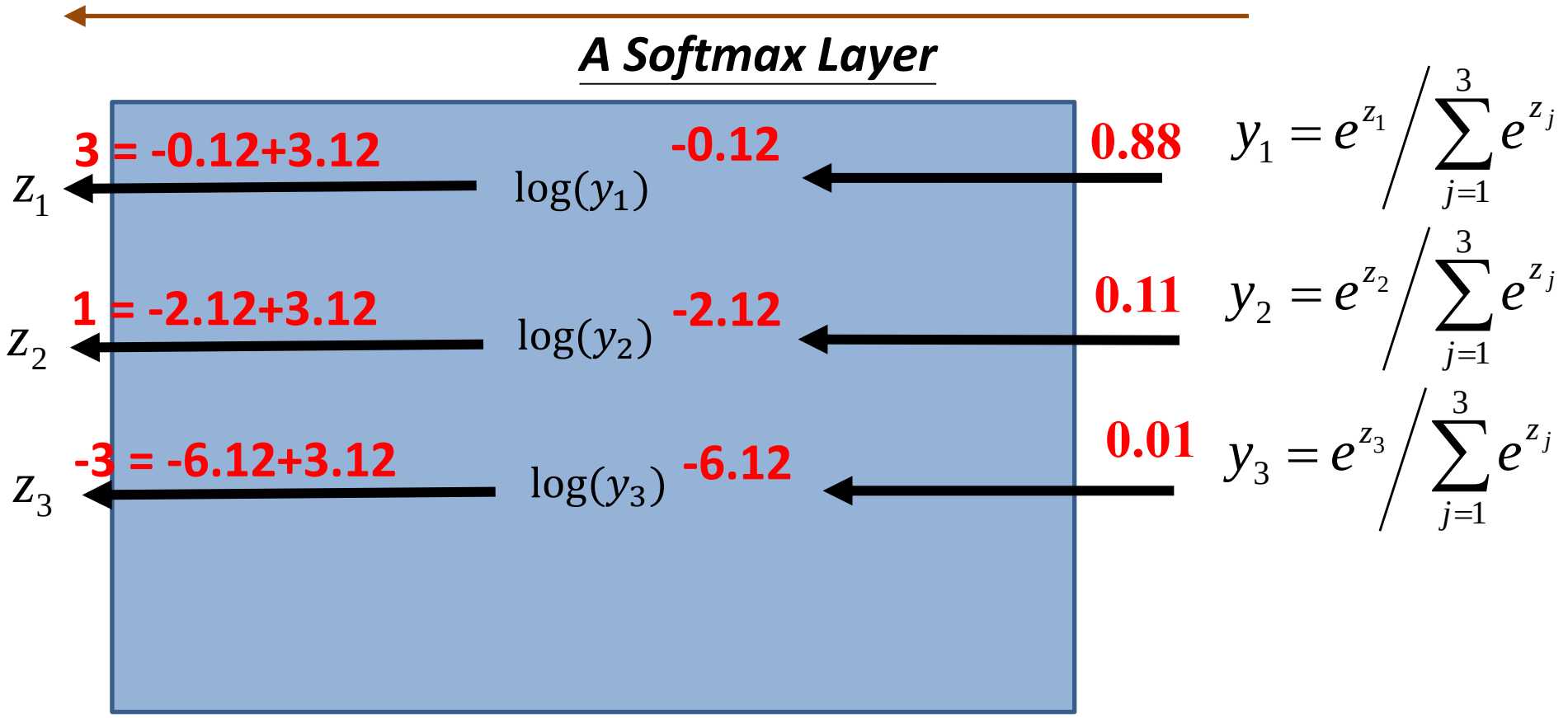
- Loss function: the logit loss from the C&W approach is used
- For ***targeted attack*** the loss is: $\ell(\mathbf{x}, \boldsymbol{t}) = max\{Z(\mathbf{x})_i : i \neq T\} - Z(\mathbf{x})_T$



Loss function

ZOO Attack

- From probabilities to logits
  - We can recover the logits up to an additive constant
  - In the example below, the additive constant is +3.12



Loss function

ZOO Attack

- The loss is: $\ell(\mathbf{x}, \boldsymbol{t}) = max\{Z(\mathbf{x})_i : i \neq T\} - Z(\mathbf{x})_T = (-0.11 + \mathrm{c}) - (-2.99 + \mathrm{c}) = -0.11 - \mathrm{c} + 2.99 - \mathrm{c} = 2.88$
- Note that if $\ell(\mathbf{x}, \boldsymbol{t}) < 0$, then the targeted attack will be successful
  - To create attacks, minimize the loss of the target class

$Z(\mathbf{x}) = \log(p(\mathbf{x})) + c$

$$\begin{bmatrix} -2.99 + c \\ \text{not needed} \\ \text{not needed} \\ -0.11 + \mathrm{c} \\ \text{not needed} \end{bmatrix}$$

Logits

$p(\mathbf{x})$

$$\begin{bmatrix} 0.05 \\ 0.02 \\ 0.02 \\ 0.90 \\ 0.01 \end{bmatrix}$$

safe
suggestive
explicit
drug
gore

Model prediction

$\boldsymbol{T}$

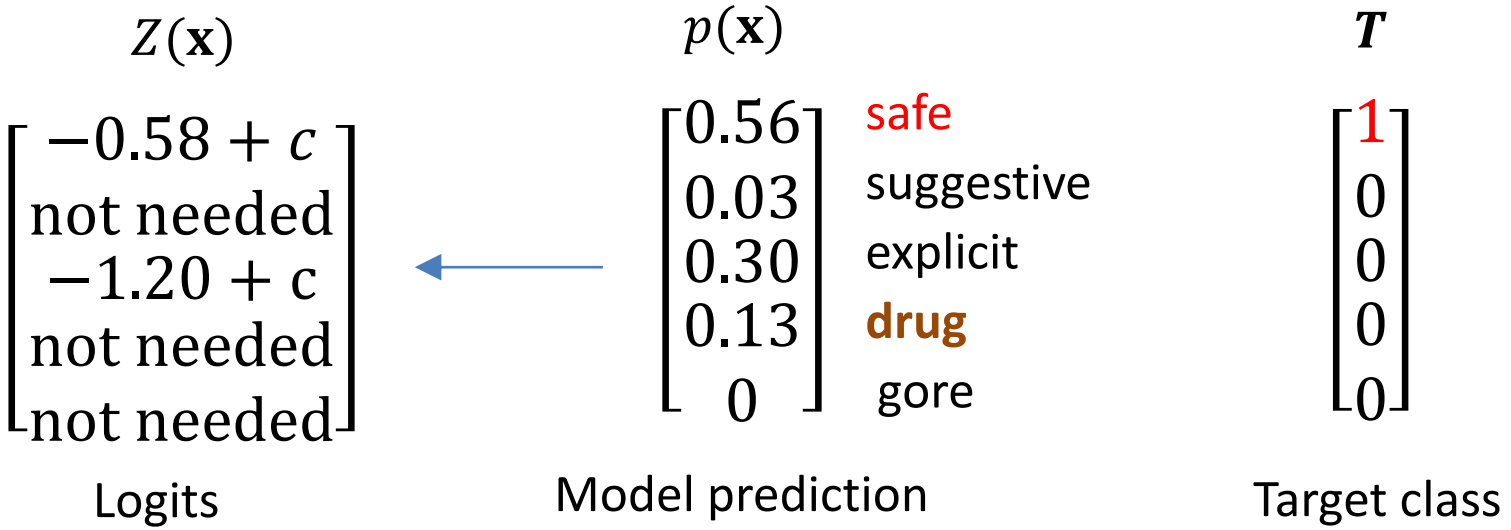
$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Target class

Loss function

ZOO Attack

- The loss is: $\ell(\mathbf{x}, \boldsymbol{t}) = max\{Z(\mathbf{x})_i : i \neq T\} - Z(\mathbf{x})_T = (-1.20 + c) - (-0.58 + c) = -1.20 + c + 0.58 - c = -0.62$
- Since $\ell(\mathbf{x}, \boldsymbol{t}) < 0$, the targeted attack is successful



Loss function

ZOO Attack

- For ***untargeted attack*** the loss is: $\ell(\mathbf{x}, \boldsymbol{t}) = max\{Z(\mathbf{x})_i : i \neq y\} - Z(\mathbf{x})_y$
  - The loss is: $\ell(\mathbf{x}, \boldsymbol{t}) = (-1.52 + c) - (-0.51 + c) = -1.52 + c + 0.51 - c = -1.01$
  - If $\ell(\mathbf{x}, \boldsymbol{t}) > 0$, then the untargeted attack will be successful

$Z(\mathbf{x}) = \log(p(\mathbf{x})) + c$

$$\begin{bmatrix} \text{not needed} \\ -1.52 + \text{c} \\ \text{not needed} \\ -0.51 + \text{c} \\ \text{not needed} \end{bmatrix}$$

Logits

$p(\mathbf{x})$

$$\begin{bmatrix} 0.05 \\ 0.22 \\ 0.02 \\ 0.60 \\ 0.11 \end{bmatrix}$$

safe
suggestive
explicit
drug
gore

Model prediction

$\boldsymbol{y}$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Ground truth label

Gradient Estimation

ZOO Attack

- ZOO (zero order optimization) of the loss $\ell(\mathbf{x}, \boldsymbol{t})$
- The gradient of the loss can be approximated by using the finite difference approach

$$\boldsymbol{g} = \nabla_{\boldsymbol{x}} \boldsymbol{f}(\mathbf{x}) \approx \frac{f(\mathbf{x}+h) - \boldsymbol{f}(\mathbf{x}-h)}{2h}$$

- I.e., if the intensity of a pixel x_i is 150, and $h = 10$, then we will query the model to give us the predictions for $f(150 + 10) = f(160)$ and for $f(150 - 10) = f(140)$, so we can estimate the gradient $\widehat{\boldsymbol{g}_i} = \nabla_{x_i} \boldsymbol{f}(\mathbf{x})$ for the pixel x_i
- Repeat for all pixels
  - E.g., for a 28×28 pixels = 784 pixels, we need to do 2 queries for each pixel, that is, we need to do $2 \cdot 784 = 1{,}568$ queries to estimate the gradient

Adversarial Attack

ZOO Attack

- First, recall the Gradient Estimation attack
  - Similar to FGSM targeted white-box attack: $\mathbf{x} = \mathbf{x_0} - \epsilon \cdot \text{sign}\left(\nabla_{\boldsymbol{x}} \boldsymbol{f}(\mathbf{x}, \mathbf{T})\right)$
  - Single step attack: $\mathbf{x} = \mathbf{x_0} - \epsilon \cdot \text{sign}\left(FD\left(Z(x)_{y'} - Z(x)_T, h\right)\right)$
  - Iterative attack: $\mathbf{x}^{t+1} = \Pi\left(\mathbf{x}_0^t + \alpha \cdot \text{sign}\left(FD\left(Z(x)_{y'} - Z(x)_T, h\right)\right)\right)$
- ZOO attack applies optimization
  - Similar to C&W targeted white-box attack

$$\text{minimize } \|\mathbf{x} - \mathbf{x_0}\|_2^2 + c \cdot \left(Z(x)_{y'} - Z(x)_T\right)$$
$$\text{subject to } \mathbf{x} \in [0,1]$$

  - ZOO solves the optimization problem with the estimated loss (with a slight abuse of the notation) based on:

$$\text{minimize } \|\mathbf{x} - \mathbf{x_0}\|_2^2 + c \cdot FD\left(Z(x)_{y'} - Z(x)_T, h\right)$$
$$\text{subject to } \mathbf{x} \in [0,1]$$

  - ***Adam optimization*** is used to solve the problem

Adam Optimization Attack

ZOO Attack

- Algorithm

Algorithm 2 ZOO-ADAM: Zeroth Order Stochastic Coordinate Descent with Coordinate-wise ADAM

Require: Step size η, ADAM states $M \in \mathbb{R}^p, v \in \mathbb{R}^p, T \in \mathbb{Z}^p$, ADAM hyper-parameters $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$

$M \leftarrow 0, v \leftarrow 0, T \leftarrow 0$
while not converged **do**
  Randomly pick a coordinate $i \in \{1, \cdots, p\}$
  Estimate $\hat{g}_i$ using (6)
  $T_i \leftarrow T_i + 1$
  $M_i \leftarrow \beta_1 M_i + (1 - \beta_1)\hat{g}_i, \quad v_i \leftarrow \beta_2 v_i + (1 - \beta_2)\hat{g}_i^2$
  $\hat{M}_i = M_i/(1 - \beta_1^{T_i}), \quad \hat{v}_i = v_i/(1 - \beta_2^{T_i})$
  $\delta^* = -\eta \frac{\hat{M}_i}{\sqrt{\hat{v}_i + \epsilon}}$
  Update $\mathbf{x}_i \leftarrow \mathbf{x}_i + \delta^*$
end while

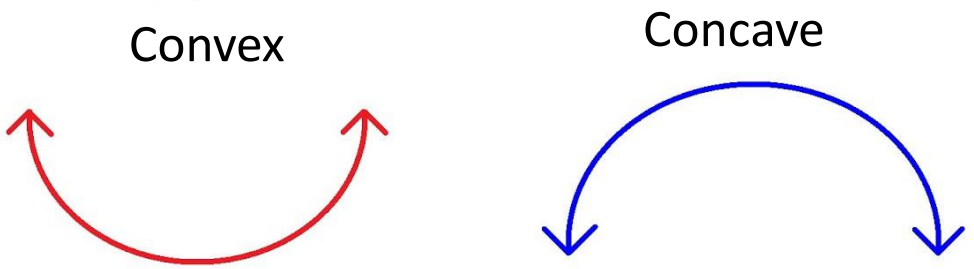
Newton Optimization Attack

ZOO Attack

- The paper proposed one more similar approach, that instead of Adam optimization uses ***Newton optimization*** method
  - Newton optimization method finds a minimum of $f(x)$ by performing the following iterations: $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$
- The approximation of the Hessian matrix of the model is estimated based on

$$\boldsymbol{h} = \frac{\partial^2}{\partial \boldsymbol{x}^2}\boldsymbol{f}(\mathbf{x}) \approx \frac{f(\mathbf{x}+h)-\mathbf{2}f(\mathbf{x})+\boldsymbol{f}(\mathbf{x}-h)}{\boldsymbol{h}^2}$$

  - If $\boldsymbol{h} > \mathbf{0}$, then the loss function is convex, update is based on $\boldsymbol{g}/\boldsymbol{h}$
  - If $\boldsymbol{h} \leq \mathbf{0}$, then the loss function is concave, update is based only on the gradient $\boldsymbol{g}$



$$\frac{\partial^2 \boldsymbol{f}(\mathbf{x})}{\partial \boldsymbol{x}^2} > 0 \qquad \frac{\partial^2 \boldsymbol{f}(\mathbf{x})}{\partial \boldsymbol{x}^2} < 0$$

Newton Optimization Attack

ZOO Attack

- Algorithm for the Newton optimization attack

Algorithm 3 ZOO-Newton: Zeroth Order Stochastic Coordinate Descent with Coordinate-wise Newton's Method

Require: Step size η

while not converged **do**
  Randomly pick a coordinate $i \in \{1, \cdots, p\}$
  Estimate $\hat{g}_i$ and $\hat{h}_i$ using (6) and (7)
  if $\hat{h}_i \leq 0$ **then**
    $\delta^* \leftarrow -\eta \hat{g}_i$
  else
    $\delta^* \leftarrow -\eta \frac{\hat{g}_i}{\hat{h}_i}$
  end if
  Update $\mathbf{x}_i \leftarrow \mathbf{x}_i + \delta^*$
end while

Experimental Evaluation

ZOO Attack

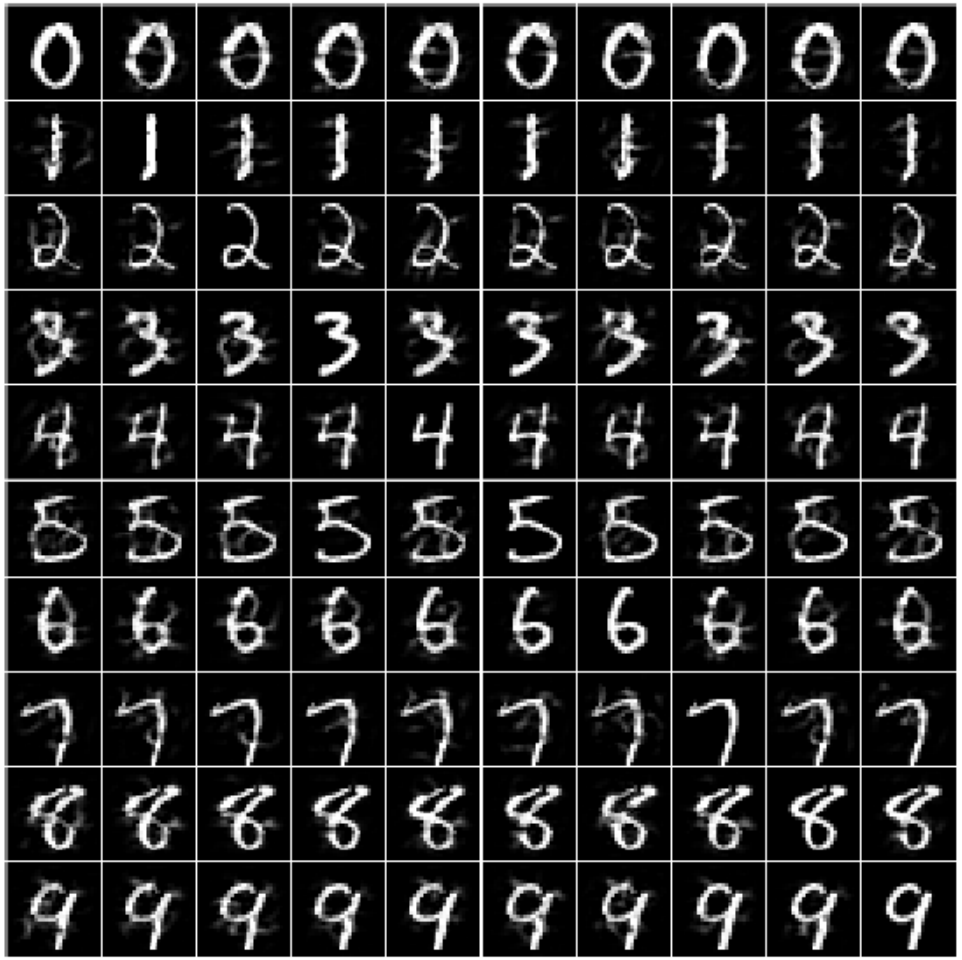
- On MNIST and Cifar-10, ZOO attacks achieve almost 100% success rate
  - The added L_2 perturbations are comparable to C&W white-box attack
  - As expected, the time for generating adversarial samples is longer than white-box attack

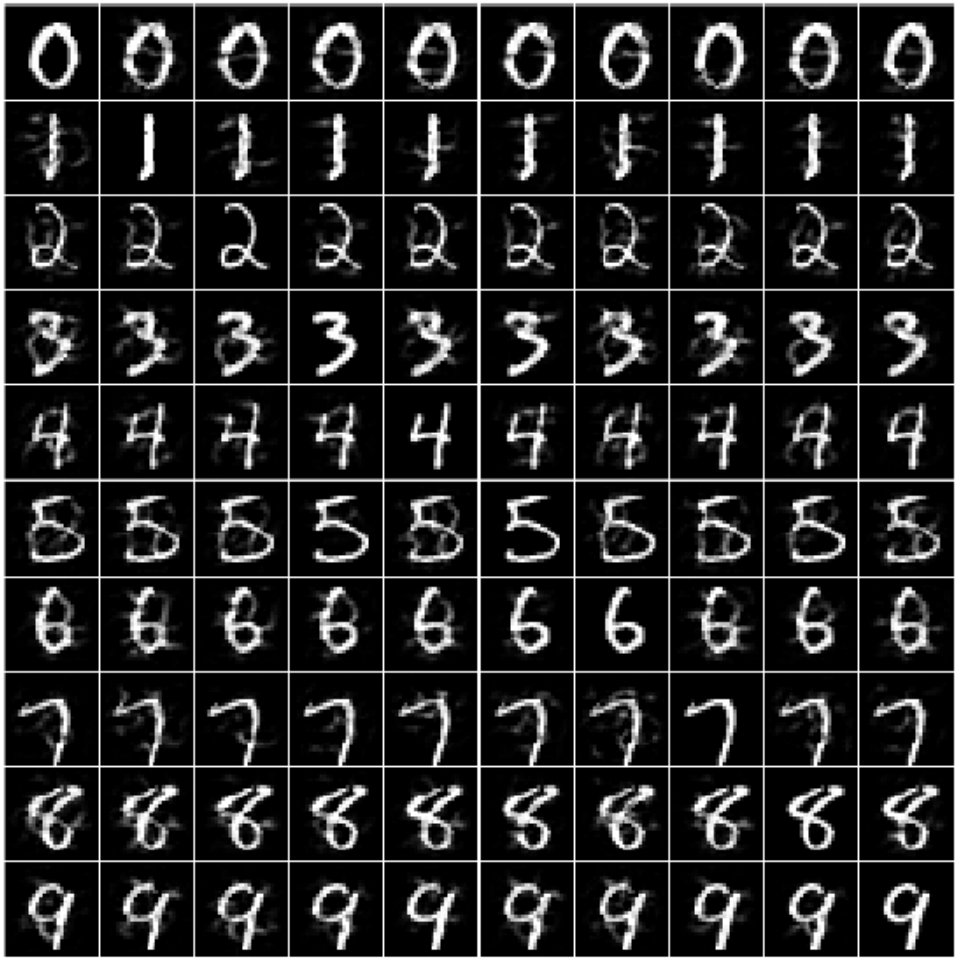
	MNIST					
	Untargeted			Targeted		
	Success Rate	Avg. L_2	Avg. Time (per attack)	Success Rate	Avg. L_2	Avg. Time (per attack)
White-box (C&W)	100 %	1.48066	0.48 min	100 %	2.00661	0.53 min
Black-box (Substitute Model + FGSM)	40.6 %	-	0.002 sec (+ 6.16 min)	7.48 %	-	0.002 sec (+ 6.16 min)
Black-box (Substitute 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

Experimental Evaluation

ZOO Attack

- Comparison between C&W white-box (left) and ZOO attack (right)





Queries Reduction

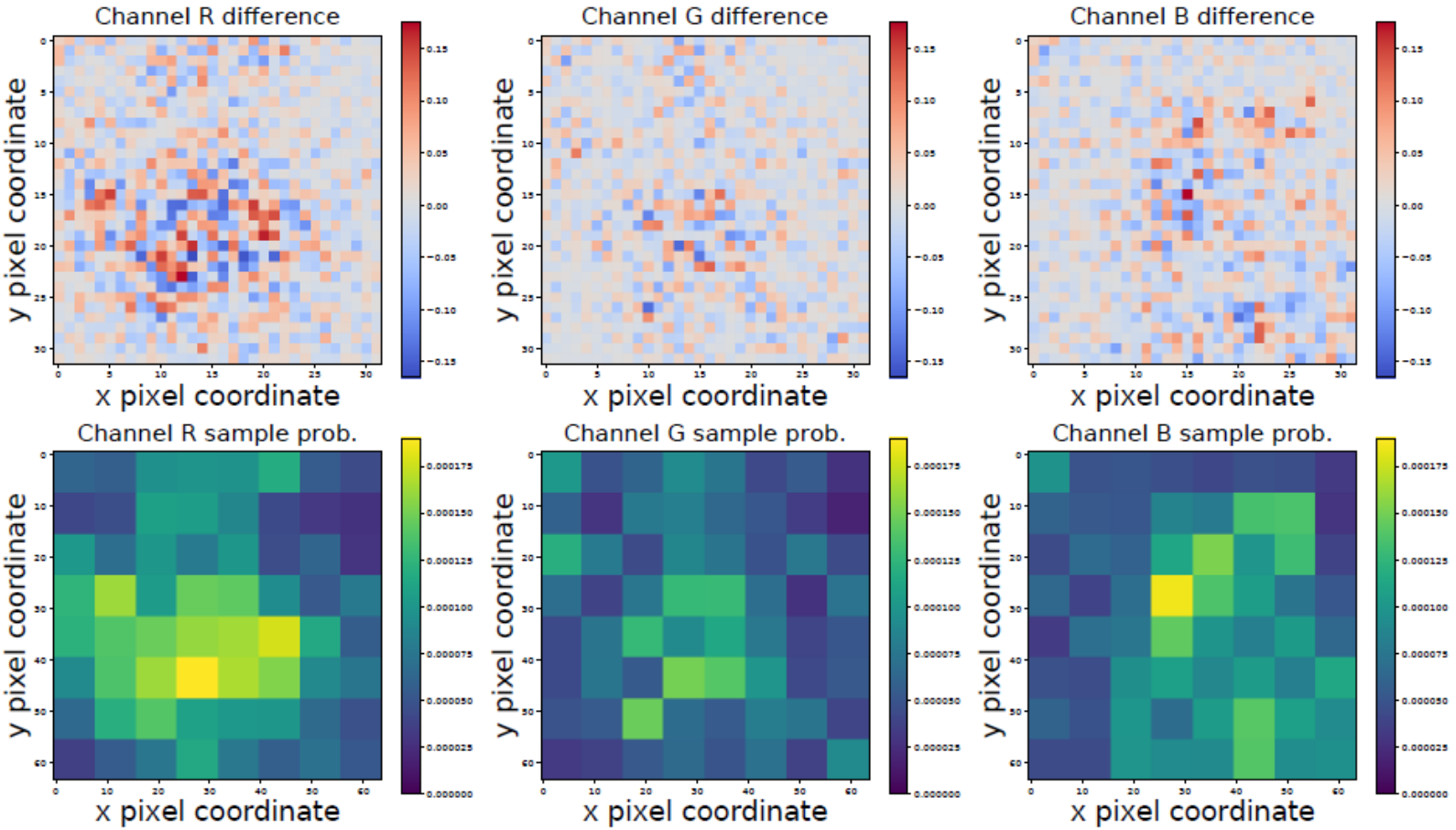
ZOO Attack

- Goal: reduce the number of queries
  - For a 28×28 pixels, we need $2 \cdot 784 = 1{,}568$ queries to estimate the gradient
  - Recall that PCA and random sets of pixels used in Gradient Estimation attack
- The proposed approach starts with reduced resolution, and the resolution is progressively increased (referred to as hierarchical attack)
  - E.g., an image of a size 299×299 pixels is used
  - Divide the image into 8×8 regions
    - Make only 64 queries to estimate the gradients
    - Optimize until the loss start decreasing
  - Increase to 16×16 regions
    - Make queries and optimize until the loss start decreasing
  - Increase to 32×32 regions
    - Make queries and optimize until the loss start decreasing
  - …

Queries Reduction

ZOO Attack

- Also, apply ***importance sampling***
  - Estimate the gradient only for the most important regions in an image
  - E.g., corner pixels are less important for this image



Experimental Evaluation

ZOO Attack

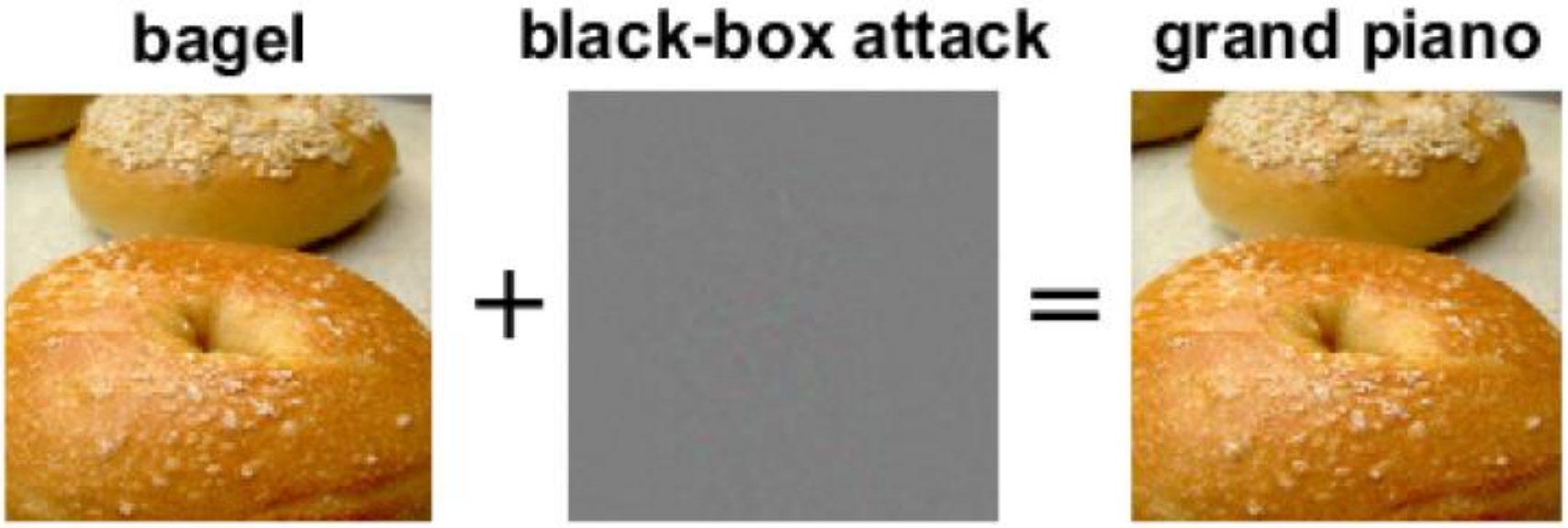
- ImageNet untargeted attack
  - Recall that there are 1,000 classes in ImageNet
  - InceptionV3 model used
  - Required about 192,000 queries per image, 20 minutes per image

	Success Rate	Avg. L_2
White-box (C&W)	100 %	0.37310
Proposed black-box (ZOO-ADAM)	88.9 %	1.19916
Black-box (Substitute Model)	N.A.	N.A.

Examples

ZOO Attack

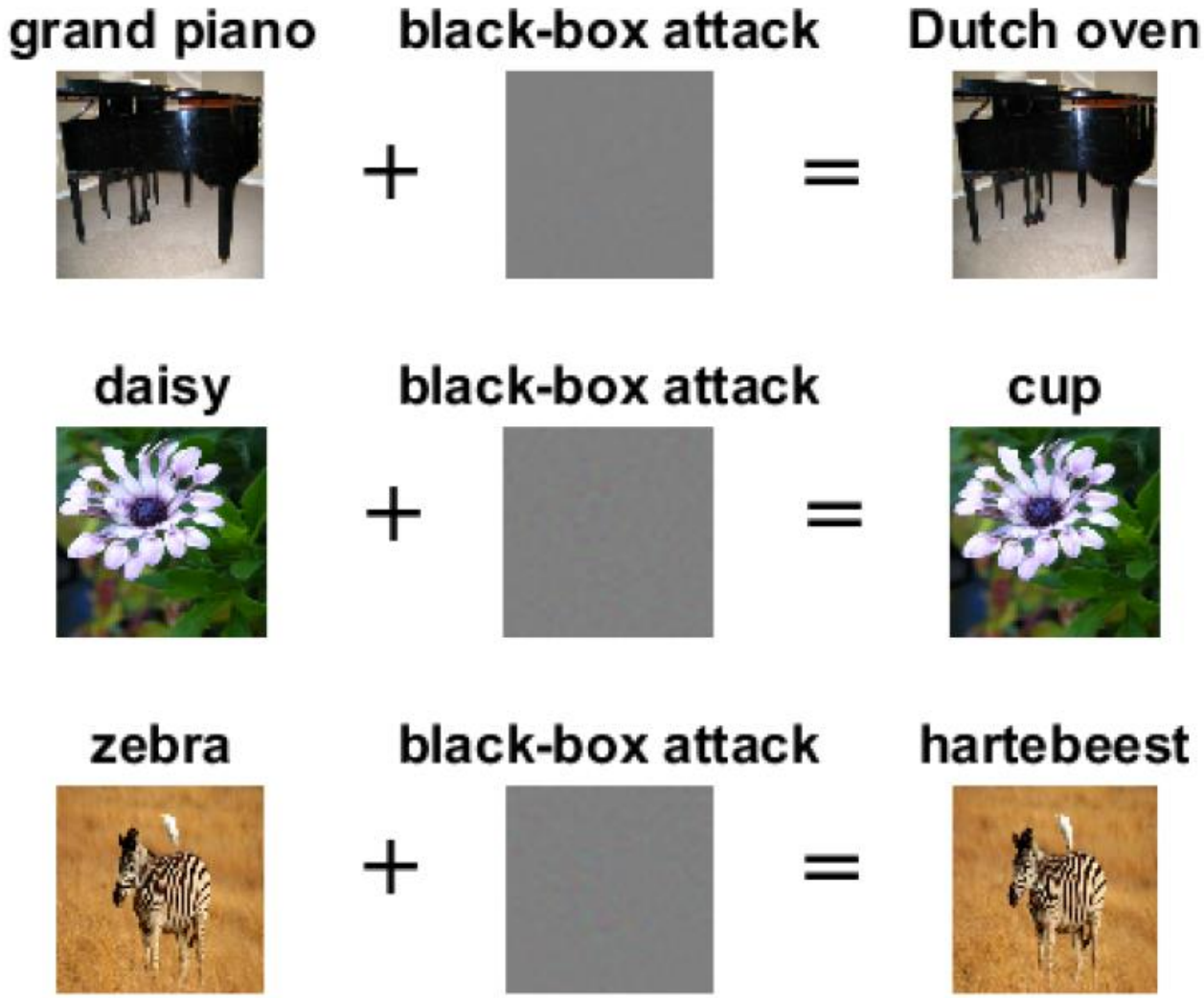
- Targeted attack
  - The added perturbations are imperceptible



Examples

ZOO Attack

- Untargeted attack



Transfer-based Attacks

- ***Transfer-based attacks*** (or ***transferability attacks***)
  - The adversary does not query the model
- ***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 for attacking a substitute model, and afterward transfer the generated adversarial samples to the target model
- ***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

Substitute Model Attack

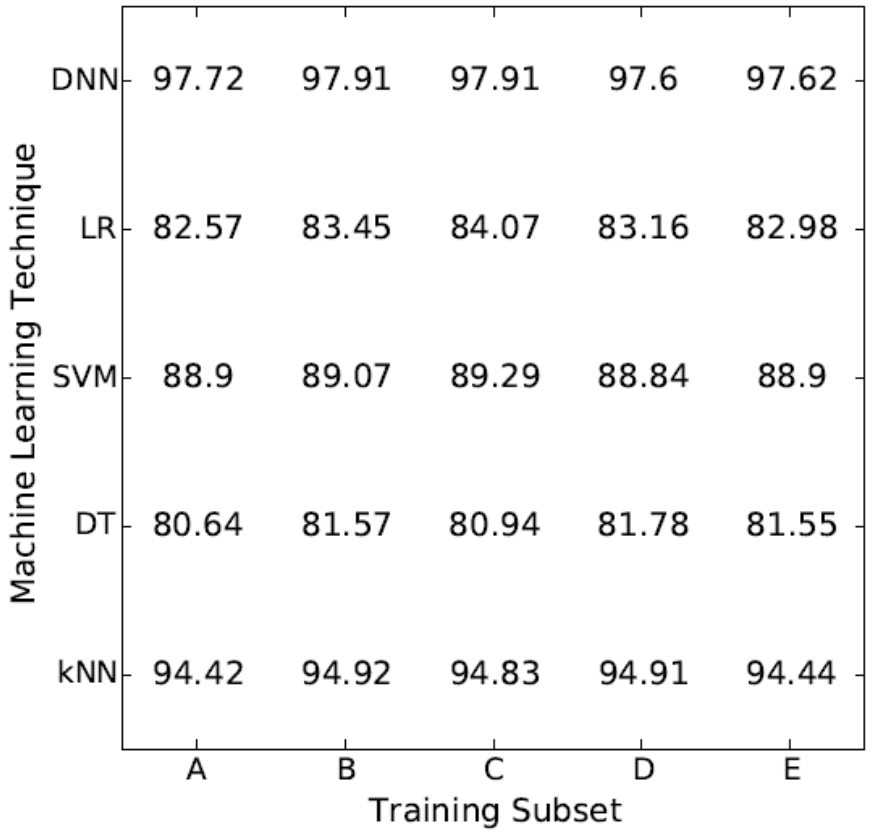
Substitute Model Attack

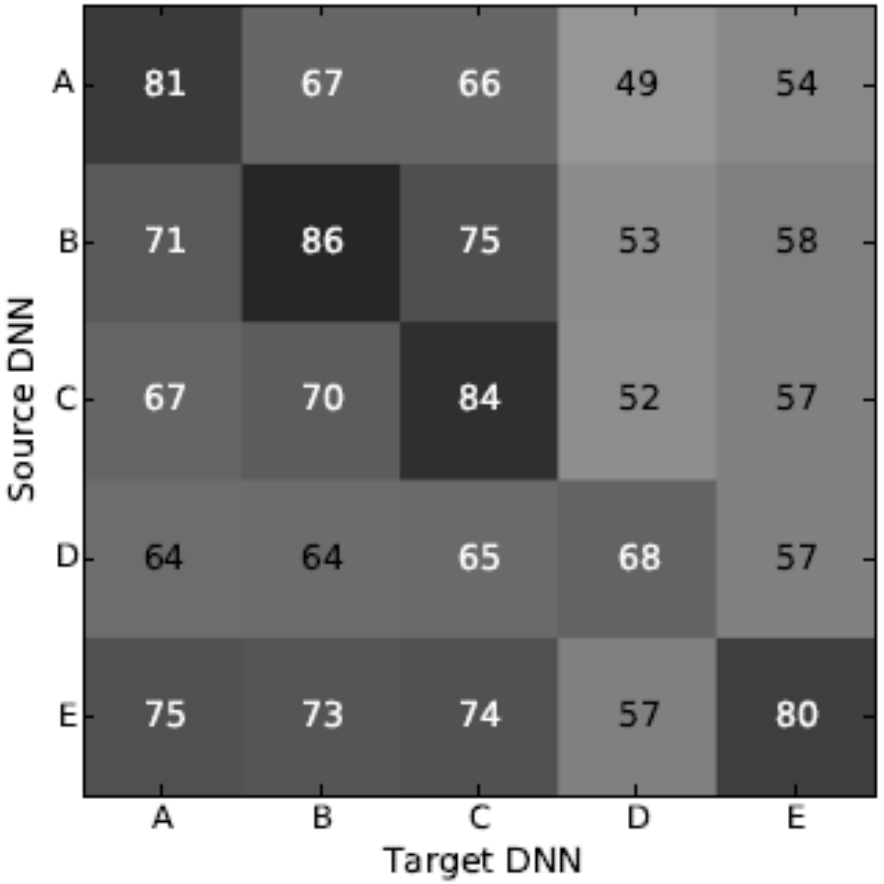
- Transferability between the following ML models is explored:
  - Deep neural networks (DNNs)
  - Logistic regression (LR)
  - Support vector machines (SVM)
  - Decision trees (DT)
  - *k*-Nearest neighbors (kNN)
  - Ensembles (Ens)
- Evaluated on MNIST

Substitute Model Attack

Substitute Model Attack

- ***Intra-technique variability***
  - E.g., adversarial examples created by one DNN are transferred to another DNN
  - Model accuracies (left) and attack success rate (right)

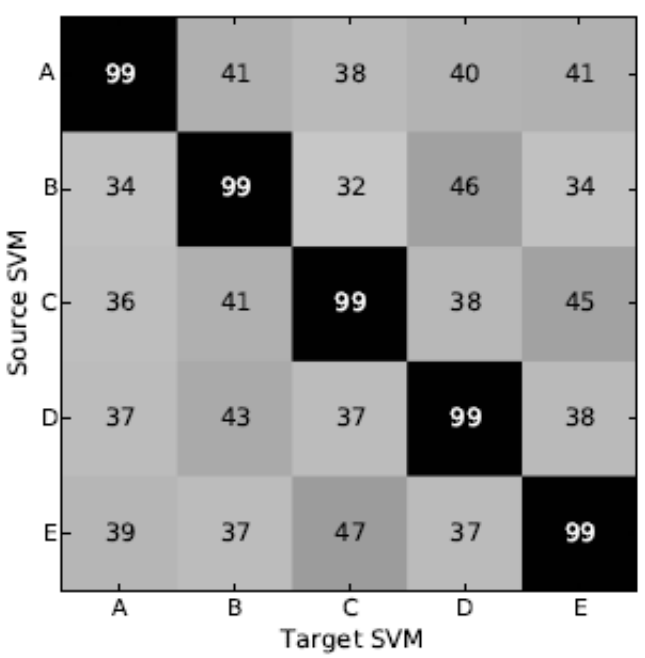


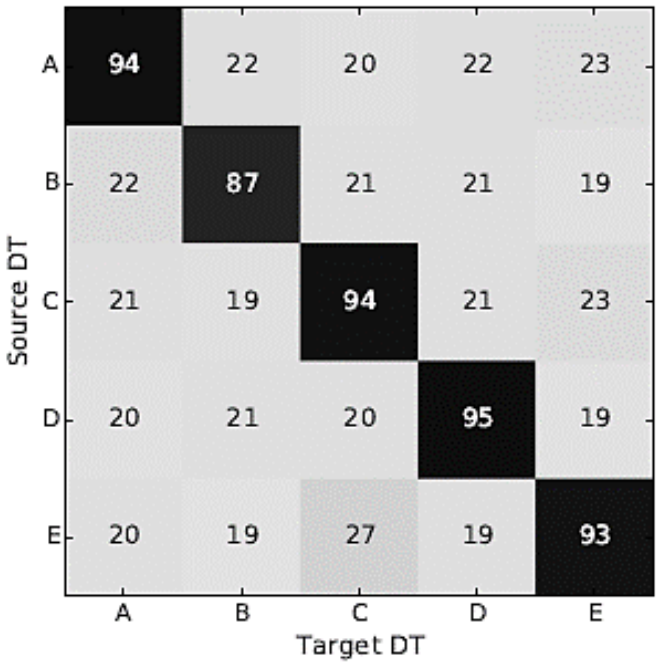


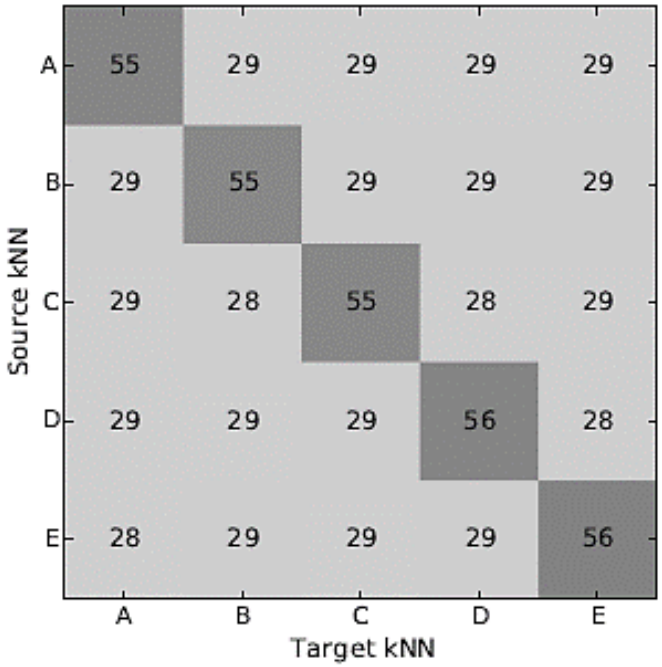
Substitute Model Attack

Substitute Model Attack

- Intra-technique variability
  - Differentiable models like DNNs and LR are more vulnerable to intra-technique transferability than non-differentiable models like SVMs, DTs, and kNNs



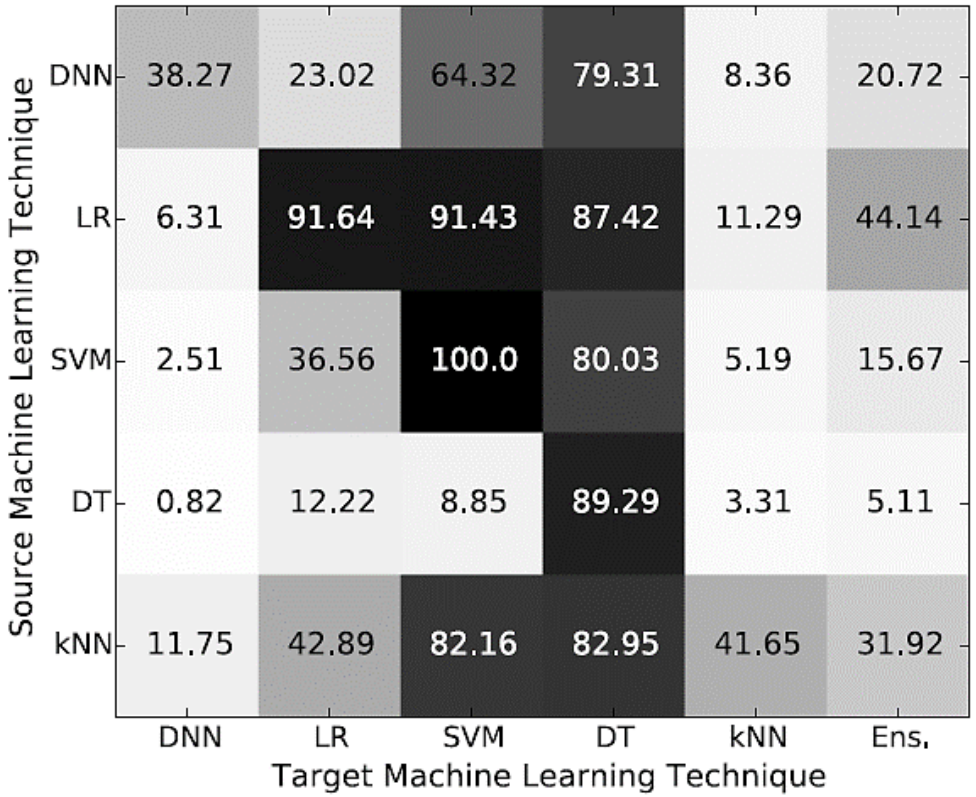




Substitute Model Attack

Substitute Model Attack

- ***Cross-technique variability***
  - E.g., adversarial examples created by DNN are transferred to other ML models
  - The most vulnerable model is DTs: misclassification rates from 79.31% to 89.29%
  - The most resilient are DNNs: misclassification between 0.82% and 38.27%



Substitute Model Attack

Substitute Model Attack

- Attacks on Amazon ML Web Services (upper table) and Google Cloud ML Prediction Oracle (lower table)

Substitute type	DNN	LR
$\rho = 3$ (800 queries)	87.44%	96.19%
$\rho = 6$ (6,400 queries)	96.78 %	96.43%
$\rho = 6$ (PSS + RS) (2,000 queries)	95.68%	95.83%

Table 3: Misclassification rates of the Amazon oracle on adversarial samples ($\varepsilon = 0.3$) produced with DNN and LR substitutes after $\rho = \{3, 6\}$ augmentation iterations. Substitutes are trained without and with refinements from Section 4: periodic step size (PSS) and reservoir sampling (RS).

Substitute type	DNN	LR
$\rho = 3$ (800 queries)	84.50%	88.94%
$\rho = 6$ (6,400 queries)	97.17%	92.05%
$\rho = 6$ (PSS + RS) (2,000 queries)	91.57%	97.72%

Table 4: Misclassification rates of the Google oracle on adversarial samples ($\varepsilon = 0.3$) produced with DNN and LR substitutes after $\rho = \{3, 6\}$ augmentation iterations.. Substitutes are trained without and with refinements from Section 4: periodic step size (PSS) and reservoir sampling (RS).

Ensemble of Local Models Attack

Ensemble of Local Models Attack

- ***Ensemble of local models attack***
  - Liu et al. (2017) Delving into Transferable Adversarial Examples and Black-box Attacks
- Observations regarding transferability
  - Transferable non-targeted adversarial examples are easy to find
  - However, targeted adversarial examples rarely transfer with their target labels
- The proposed approach allows transferring targeted adversarial examples

Ensemble of Local Models Attack

Ensemble of Local Models Attack

- On ImageNet, targeted examples do not transfer across models
  - Only a small percentage of adversarial images retain the target label when transferred to other models
  - RMSD is the average perturbation of the used adversarial images

	RMSD	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
ResNet-152	23.13	100%	2%	1%	1%	1%
ResNet-101	23.16	3%	100%	3%	2%	1%
ResNet-50	23.06	4%	2%	100%	1%	1%
VGG-16	23.59	2%	1%	2%	100%	1%
GoogLeNet	22.87	1%	1%	0%	1%	100%

- On the other hand, untargeted examples transfer well

	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%

Ensemble of Local Models Attack

Ensemble of Local Models Attack

- Hypothesis: if an adversarial image remains adversarial for multiple models, it is more likely to transfer to other models as well
- Approach: solve the following optimization problem (for targeted attack):

$$\mathbf{argmin}_{x\star} - \log\left((\sum_{i=1}^{k} \alpha_i J_i(x^\star)) \cdot \mathbf{1}_{y^\star}\right) + \lambda d(x, x^\star)$$

- The problem is similar to C&W
  - x is a clean image
  - x^* is an adversarial image
  - $d(x, x^*)$ is distance function
  - $J_1, J_2, \ldots, J_k$ are white-box models in the ensemble
  - $\alpha_1, \alpha_2, \ldots, \alpha_k$ are the ensemble weights
  - $-\log(\alpha_1 J_1 \cdot \mathbf{1}_{y^*})$ is the cross-entropy loss between the prediction by model J_1 and the one-hot vector for the target class $\mathbf{1}_{y^*}$

Targeted Attack Evaluation

Ensemble of Local Models Attack

- Targeted attack using the ensemble attack

	RMSD	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
-ResNet-152	30.68	38%	76%	70%	97%	76%
-ResNet-101	30.76	75%	43%	69%	98%	73%
-ResNet-50	30.26	84%	81%	46%	99%	77%
-VGG-16	31.13	74%	78%	68%	24%	63%
-GoogLeNet	29.70	90%	87%	83%	99%	11%

Non-targeted Attack Evaluation

Ensemble of Local Models Attack

- Non-targeted ensemble attack results

	RMSD	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
-ResNet-152	17.17	0%	0%	0%	0%	0%
-ResNet-101	17.25	0%	1%	0%	0%	0%
-ResNet-50	17.25	0%	0%	2%	0%	0%
-VGG-16	17.80	0%	0%	0%	6%	0%
-GoogLeNet	17.41	0%	0%	0%	0%	5%

HopSkipJump Attack

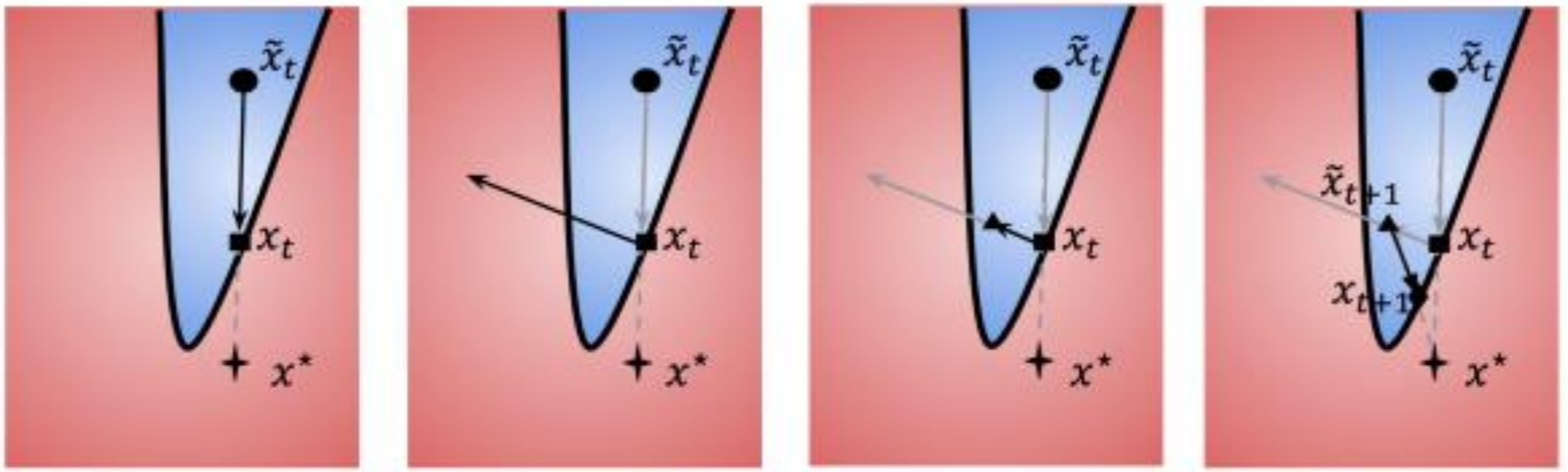
Other Black-box Evasion Attacks

- ***HopSkipJump Attack***
  - Chen and Jordan (2019) HopeSkipJumpAttack: A Query-efficient Decision-based Adversarial Attack
- This attack is an extension of the Boundary Attack
  - I.e., it is a decision-based attack, has access only to the predicted output class
  - Requires fewer queries than the Boundary Attack
  - It includes both untargeted and targeted attacks
  - Proposes a a novel approach for estimation of the gradient direction along the decision boundary

HopSkipJump Attack

Other Black-box Evasion Attacks

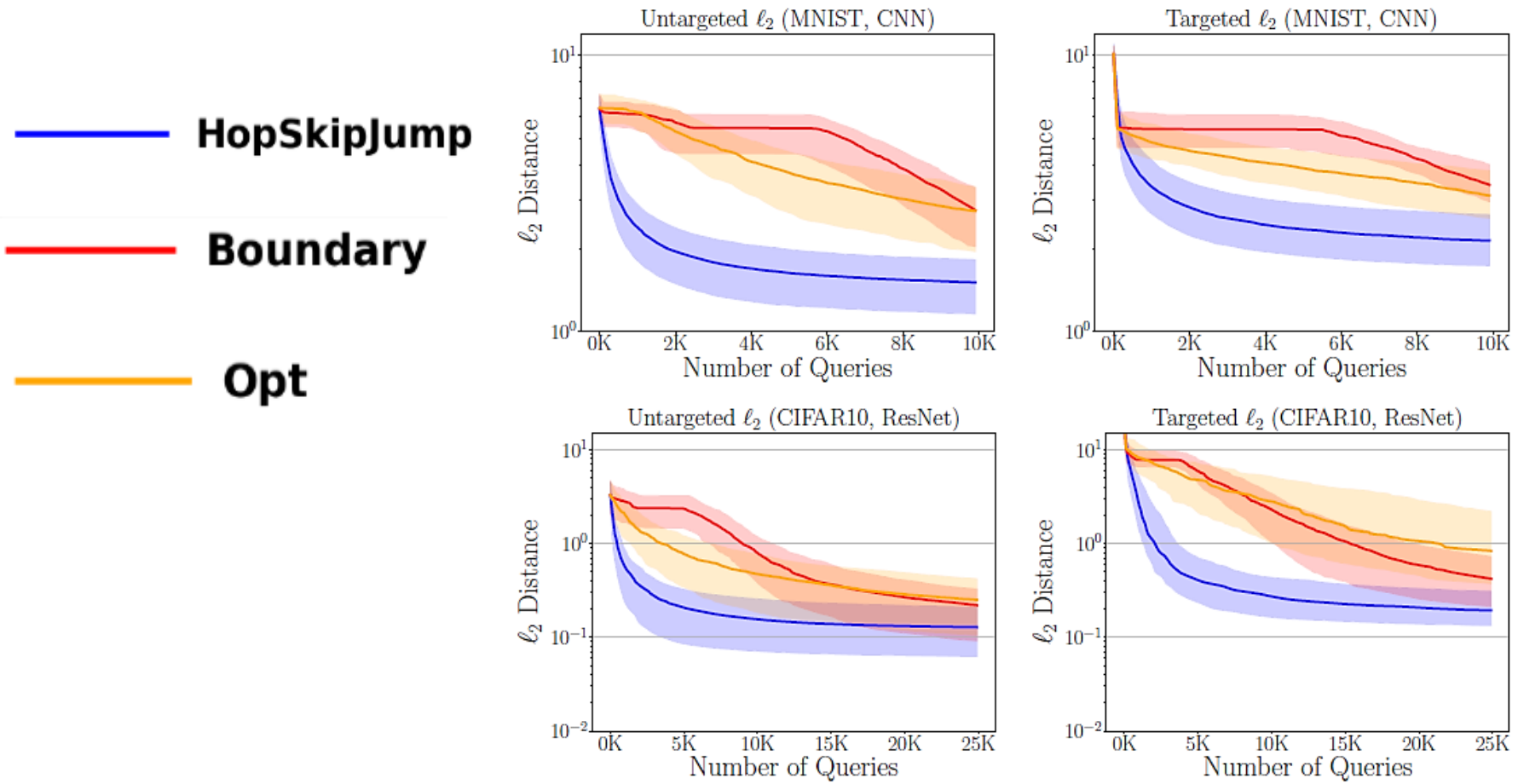
- Start from an adversarial image $\tilde{x}_t$
- Perform a binary search to find the boundary
- Estimate the gradient direction at the boundary point x_t
- Perform a step-size search, and update to the next image $\tilde{x}_{t+1}$
- Search again for the next boundary point x_{t+1}
- Repeat until the closest adversarial image to the original image x^* is found



HopSkipJump Attack

Other Black-box Evasion Attacks

- Experimental evaluation
  - Comparison to Boundary attack and Opt attack on CI-FAR-100
  - Top row (CNN), bottom row (ResNet)

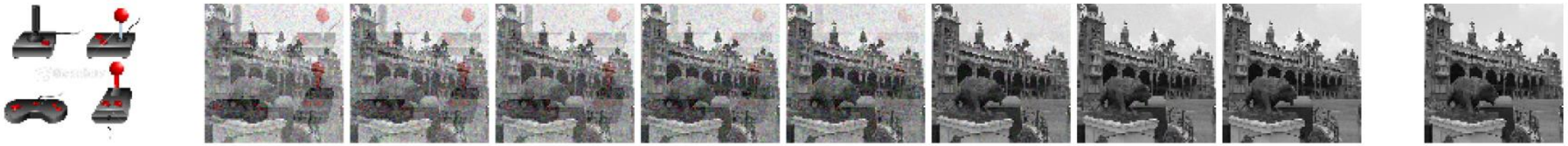


HopSkipJump Attack

Other Black-box Evasion Attacks

- Untargeted attack
  - 2nd to 9th columns: images at 100, 200, 500, 1K, 2K, 5K, 10K, 25K model queries

- Targeted attack



Simple Black-box Attack

Other Black-box Evasion Attacks

- ***Simple Black-box Attack***
  - Guo et al. (2019) Simple Black-box Adversarial Attacks
- A.k.a. SimBA attack
  - Score-based attack
  - Focus on query efficiency
  - Both targeted and untargeted attacks
- Approach:
  - Use random orthonormal perturbations to approach the decision boundary

Simple Black-box Attack

Other Black-box Evasion Attacks

- Steps:
  - Randomly sample perturbation vectors from a predefined orthonormal basis
  - Check the confidence score to find out if it is pointing toward or away from the decision boundary
  - Perturb the image by adding or subtracting the vector
- Goal:
  - Each iteration moves the image away from the original image, and towards the decision boundary

Simple Black-box Attack

Other Black-box Evasion Attacks

- Algorithm

Algorithm 1 SimBA in Pseudocode

procedure SIMBA($\mathbf{x}, y, Q, \epsilon$)
  $\delta = \mathbf{0}$
  $\mathbf{p} = p_h(y \mid \mathbf{x})$
  while $\mathbf{p}_y = \max_{y'} \mathbf{p}_{y'}$ **do**
    Pick randomly without replacement: $\mathbf{q} \in Q$
    for $\alpha \in \{\epsilon, -\epsilon\}$ **do**
      $\mathbf{p}' = p_h(y \mid \mathbf{x} + \delta + \alpha\mathbf{q})$
      if $\mathbf{p}'_y < \mathbf{p}_y$ **then**
        $\delta = \delta + \alpha\mathbf{q}$
        $\mathbf{p} = \mathbf{p}'$
        break
  return δ

Simple Black-box Attack

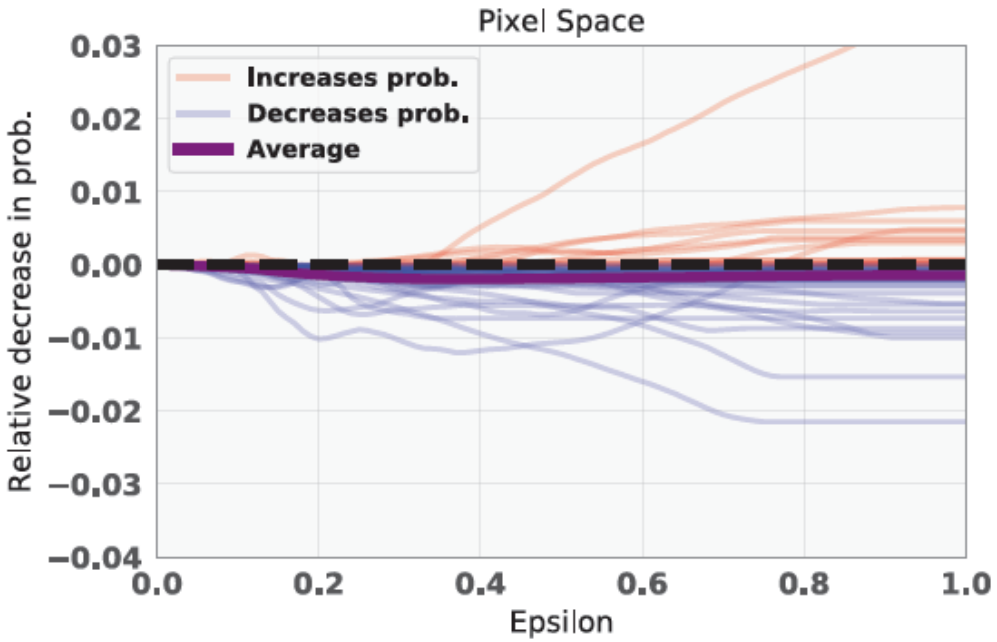
Other Black-box Evasion Attacks

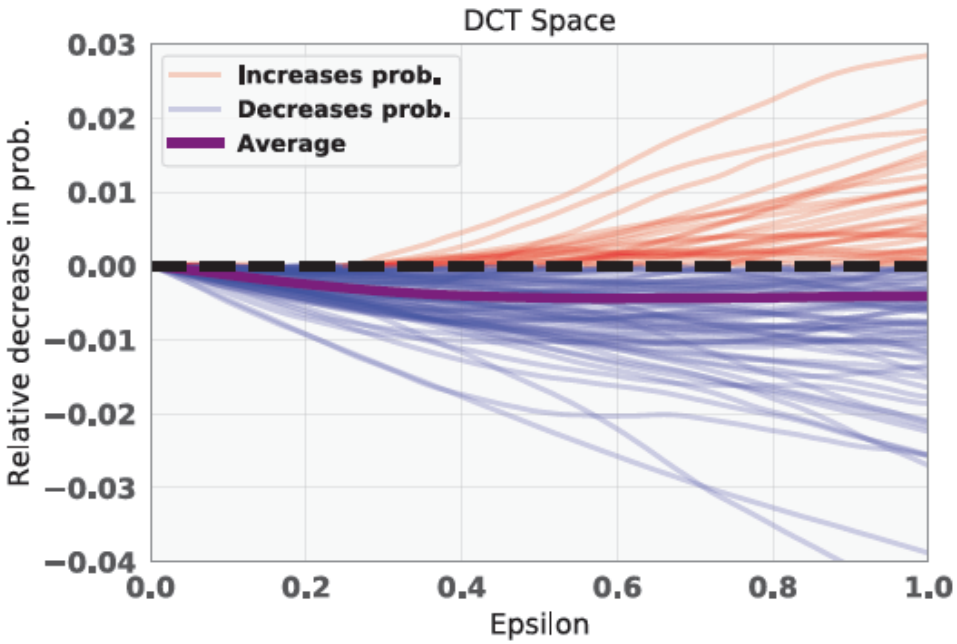
- Perturbation vectors are selected to be orthonormal
  - I.e., the random directions for each pixel do not cancel each other out, or amplify each other
- For orthonormal vectors $\mathbf{x}$ and $\mathbf{y}$, the dot product is $\mathbf{x} \cdot \mathbf{y} = 0$
  - The angle between the vectors is 90 degrees
  - I.e., they are orthogonal
- How to choose orthonormal perturbation vector?
  - One inefficient option are the vectors [1,0,0,…,0], [0,1,0,…,0], [0,0,1,…,0],…,[0,0,0,…,1]
    - I.e., only one pixel is changed at a time
  - The authors propose an approach called Discrete Cosine Transform (DCT)
    - It is based on frequency coefficients that correspond to magnitudes of cosine functions
    - Hence, many pixels will change the direction at each step

Simple Black-box Attack

Other Black-box Evasion Attacks

- The average change of the output scores (probabilities) is larger when the DCT approach is employed
  - In comparison to changing individual pixels

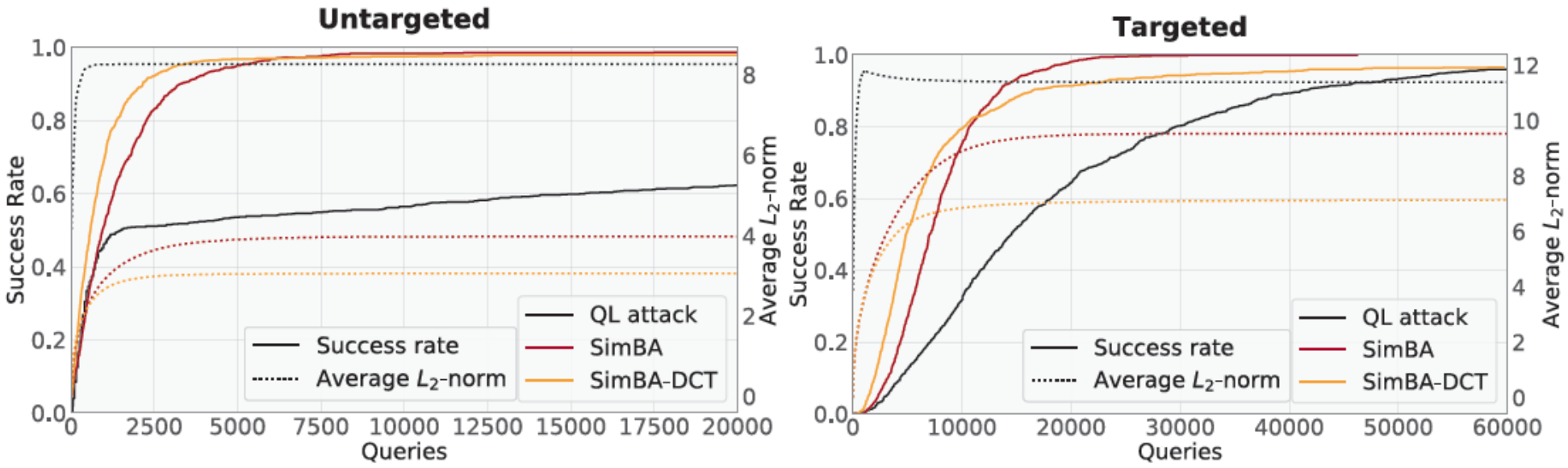




Simple Black-box Attack

Other Black-box Evasion Attacks

- Experimental evaluation
  - High success rate achieved



Simple Black-box Attack

Other Black-box Evasion Attacks

- Experimental evaluation
  - Good query-efficiency

Untargeted

Attack	Average queries	Average L_2	Success rate
Label-only			
Boundary attack	123,407	5.98	100%
Opt-attack	71,100	6.98	100%
LFBA	30,000	6.34	100%
Score-based			
QL-attack	28,174	8.27	85.4%
Bandits-TD	5,251	5.00	80.5%
SimBA	1,665	3.98	98.6%
SimBA-DCT	**1,283**	3.06	97.8%

Targeted

Attack	Average queries	Average L_2	Success rate
Score-based			
QL-attack	20,614	11.39	98.7%
AutoZOOM	13,525	26.74	100%
SimBA	**7,899**	9.53	100%
SimBA-DCT	8,824	7.04	96.5%

Simple Black-box Attack

Other Black-box Evasion Attacks

- Attack on Google Cloud Vision API
  - Checked on 50 random images
  - 70% success rate after 5,000 queries

origin_54.BMP

Camera Accessory	87%
Product	82%
Hardware	67%
Optical Instrument	66%
Camera Lens	61%
Gun	61%
Product	58%
Weapon	53%

after_54.BMP

Weapon	94%
Gun	94%
Firearm	76%
Air Gun	65%
Trigger	63%
Optical Instrument	59%
Airsoft Gun	58%
Rifle	51%

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