CS 499/579: Trustworthy ML

Preliminaries on adversarial examples

Tu/Th 4:00 – 5:50 pm

Instructor: **Sanghyun Hong**

sanghyun.hong@oregonstate.edu

Oregon State University

SAIL

Secure AI Systems Lab

Notes

- Call for actions
  - In-class presentation sign-ups
  - Term project team-up (by the 10th)

Topics for part I – adversarial examples

- Research questions
  - What are the adversarial examples?
  - How can we find adversarial examples?
  - How can we exploit them in practice?
  - How can we defeat adversarial examples?

How can we train models robust to adversarial inputs?

Towards deep learning models resistant to adversarial attacks, Madry et al., ICLR 2018

How did the research go?

- Many attack proposals
  - FGSM
  - JSMA
  - DeepFool
  - DeepXplore[1]
  - C&W
  - …
- Many defense proposals
  - Regularization … broken
  - Defensive distillation … broken
  - Adversarial training ... but with which attack?
  - …

[1] Pei et al., DeepXplore: Automated Whitebox Testing of Deep Learning Systems, SOSP 2017

How did the research go?

- Main research question
  - How can we train neural networks robust to adversarial examples?

Revisiting the formulation

- Test-time (evasion) attack
  - Suppose
    - A test-time input (x, y)
    - $(x, y) \sim D$, D: data distribution; $x \in R^d$ and $y \in [k]$; $x \in [0, 1]$
    - A NN model f and its parameters θ
    - $L(\theta, x, y)$: a loss function
  - Objective
    - Find an $x^{adv} = x + \delta$ such that $f\left(x^{adv}\right) \neq y$ while $||\delta||_p \leq \varepsilon$

Revisiting the formulation

- Test-time (evasion) attack
  - Suppose
    - A test-time input (x, y)
    - $(x, y) \sim D$, D: data distribution; $x \in R^d$ and $y \in [k]$; $x \in [0, 1]$
    - A NN model f and its parameters θ
    - $L(\theta, x, y)$: a loss function
  - Attacker's objective
    - Find an $x^{adv} = x + \delta$ such that $\max_{\delta \in S} L\left(\theta, x^{adv}, y\right)$ while $||\delta||_p \leq \varepsilon$

Revisiting the formulation

- Test-time (evasion) attack
  - Suppose
    - A test-time input (x, y)
    - $(x, y) \sim D$, D: data distribution; $x \in R^d$ and $y \in [k]$; $x \in [0, 1]$
    - A NN model f and its parameters θ
    - $L(\theta, x, y)$: a loss function
  - Attacker's objective
    - Find an $x^{adv} = x + \delta$ such that $\max_{\delta \in S} L(\theta, x^{adv}, y)$ while $||\delta||_p \leq \varepsilon$
  - Defender's objective
    - Train a neural network f robust to adversarial attacks
    - Find θ such that $\min_{\theta} \rho(\theta)$ where $\rho(\theta) = \mathrm{E}_{(x,y)\sim D}\left[L\left(\theta, x^{adv}, y\right)\right]$

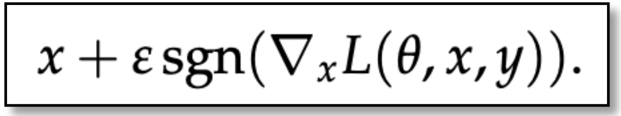
Putting all together

- (Models resilient to) test-time (evasion) attack
  - Suppose
    - A test-time input (x, y)
    - $(x, y) \sim D$, D: data distribution; $x \in R^d$ and $y \in [k]$; $x \in [0, 1]$
    - A NN model f and its parameters θ
    - $L(\theta, x, y)$: a loss function
  - Min-max optimization (between attacker's and defender's objectives)
    - Find $\min_{\theta} \rho(\theta)$ where $\rho(\theta) = \mathrm{E}_{(x,y)\sim D}\left[\max_{\delta \in S} L(\theta, x + \delta, y)\right]$ while $||\delta||_p \leq \varepsilon$
    - s: a set of test-time samples

Saddle point problem: inner maximization and outer minimization

Inner maximization using the first-order adversary

- Revisit FGSM (Fast Gradient Sign Method)



$$x + \varepsilon \operatorname{sgn}(\nabla_x L(\theta, x, y)).$$

  - FGSM can be viewed as a simple one-step toward maximizing the loss (inner part)

Inner maximization

- Revisit FGSM (Fast Gradient Sign Method)

$$\boxed{x + \varepsilon\,\mathrm{sgn}(\nabla_x L(\theta, x, y)).}$$

  - FGSM can be viewed as a simple one-step toward maximizing the loss (inner part)

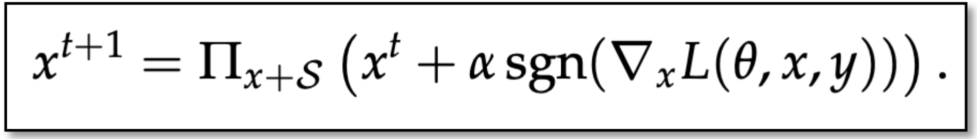
- PGD (Projected Gradient Descent)

$$\boxed{x^{t+1} = \Pi_{x+\mathcal{S}} \underbrace{\left(x^t + \alpha\,\mathrm{sgn}(\nabla_x L(\theta, x, y))\right)}_{\text{FGSM}}.}$$

  - Multi-step adversary; much stronger than FGSM attack

Inner maximization

- PGD (Projected Gradient Descent)



$$x^{t+1} = \Pi_{x+\mathcal{S}} \left(x^t + \alpha \, \mathrm{sgn}(\nabla_x L(\theta, x, y)) \right).$$

  - Multi-step adversary; much stronger than FGSM attack
  - Hyper-parameters
    - t: number of iterations
    - α: step-size
    - ε: perturbation bound $|x^* - x|_p$
  - Notation: PGD-t, bounded by ε, used the step-size of α

Outer minimization

- PGD (Projected Gradient Descent)

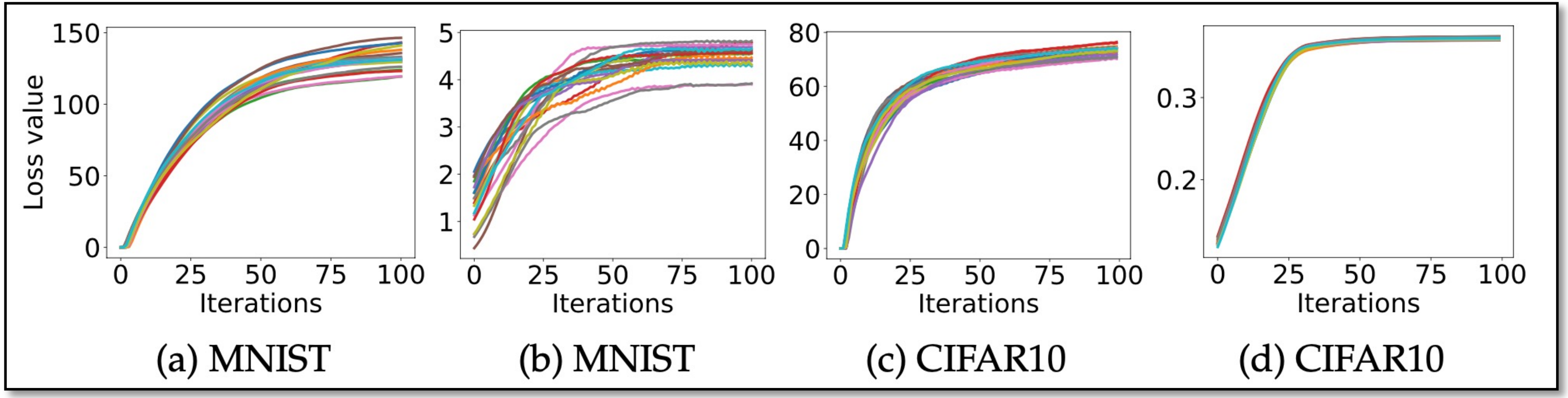
$$x^{t+1} = \Pi_{x+\mathcal{S}} \left(x^t + \alpha \operatorname{sgn}(\nabla_x L(\theta, x, y)) \right) .$$

  - Multi-step adversary; much stronger than FGSM attack

- Adversarial training
  - Make a model do correct prediction on adversarial examples
  - Training procedure
    - At each iteration of training
    - Craft PGD-t adversarial examples
    - Update the model towards making it correct on those adv examples

EVALUATION

- Findings
  - (1, 3) PGD increases the loss values in a fairly consistent way
  - (2, 4) Models trained with PGD attacks are resilient to the same attacks



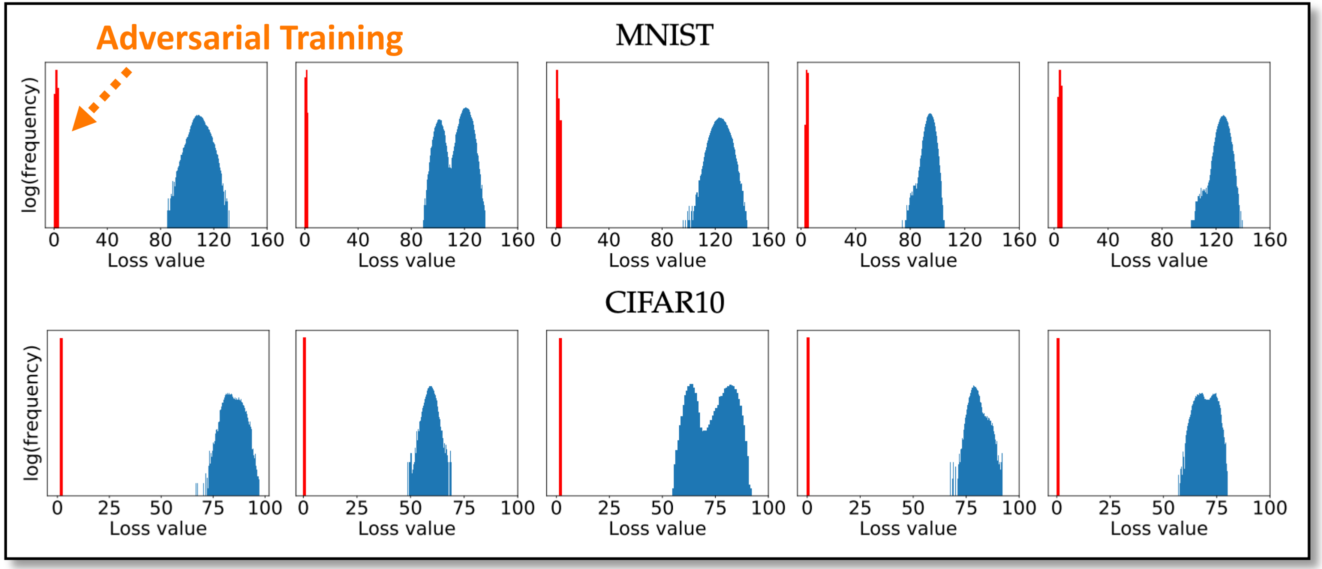
(a) MNIST (b) MNIST (c) CIFAR10 (d) CIFAR10

Adversarial Training

Adversarial Training

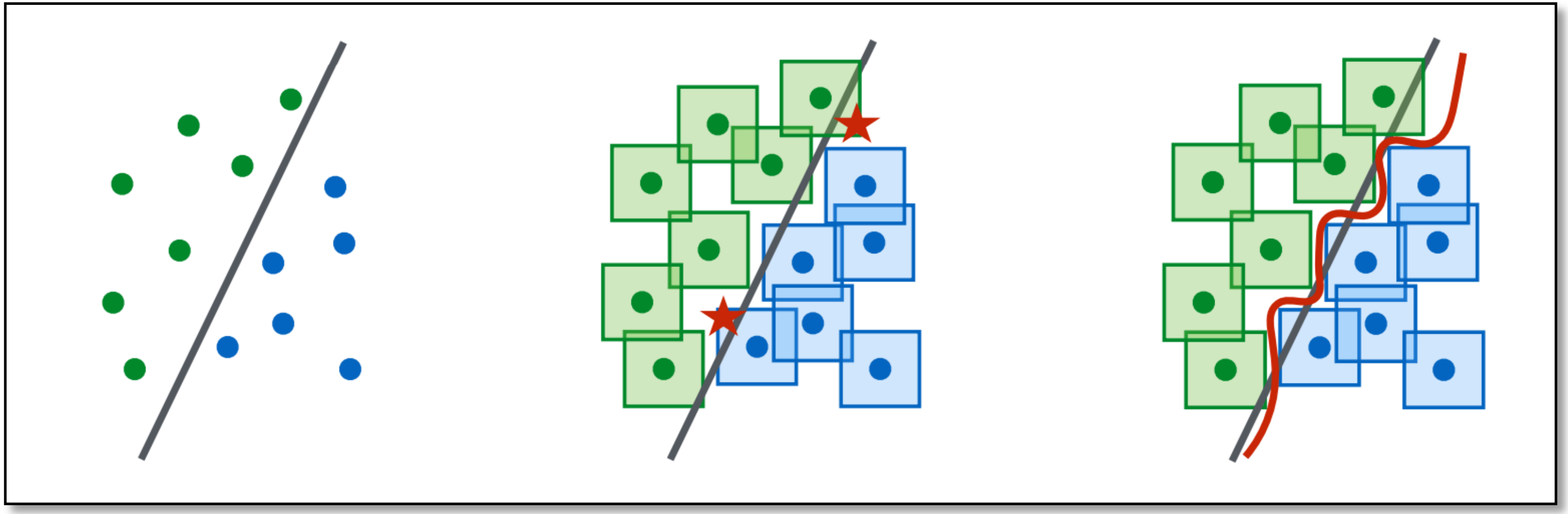
Evaluation

- Findings
  - PGD increases the loss values in a fairly consistent way
  - Models trained with PGD attacks are resilient to the same attacks
  - Final loss of PGD attacks are concentrated (both for defended/undefended models)



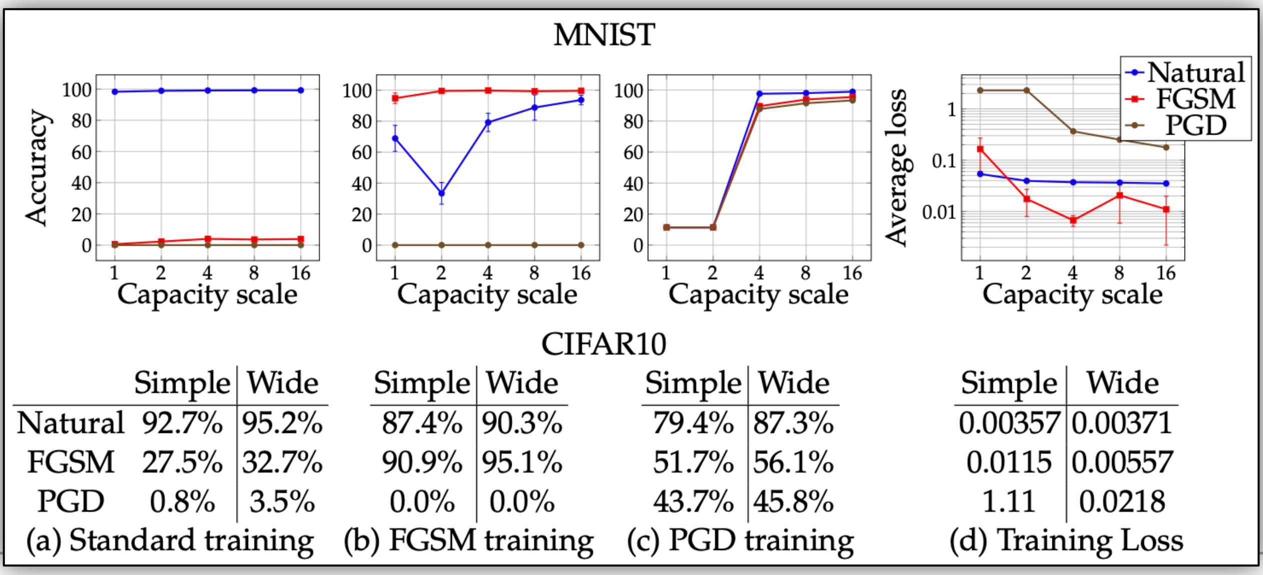
Evaluation

- Why adversarial training (AT) works?
  - Capacity is crucial for the robustness: robust models need complex decision boundary
  - Capacity alone helps: high-capacity models show more robustness w/o AT



Evaluation

- … Cont'd
  - Capacity is crucial for the robustness: robust models need complex decision boundary
  - Capacity alone helps: high-capacity models show more robustness w/o AT
  - AT with weak attacks (like FGSM) can't defeat a strong one like PGD
  - (optional) Robustness may be at odds with accuracy



MNIST

CIFAR10

	Simple	Wide	Simple	Wide	Simple	Wide	Simple	Wide
Natural	92.7%	95.2%	87.4%	90.3%	79.4%	87.3%	0.00357	0.00371
FGSM	27.5%	32.7%	90.9%	95.1%	51.7%	56.1%	0.0115	0.00557
PGD	0.8%	3.5%	0.0%	0.0%	43.7%	45.8%	1.11	0.0218

(a) Standard training (b) FGSM training (c) PGD training (d) Training Loss

Summary

- Bottom-line
  - PGD is a strong attack we can use
  - Training a model with PGD can make it resilient to the first-order adversary
  - To achieve such robustness, we need sufficient model complexity

CS 499/579: Trustworthy ML

Adversarial attacks: transferability

Tu/Th 4:00 – 5:50 pm

Instructor: **Sanghyun Hong**

sanghyun.hong@oregonstate.edu

Oregon State University

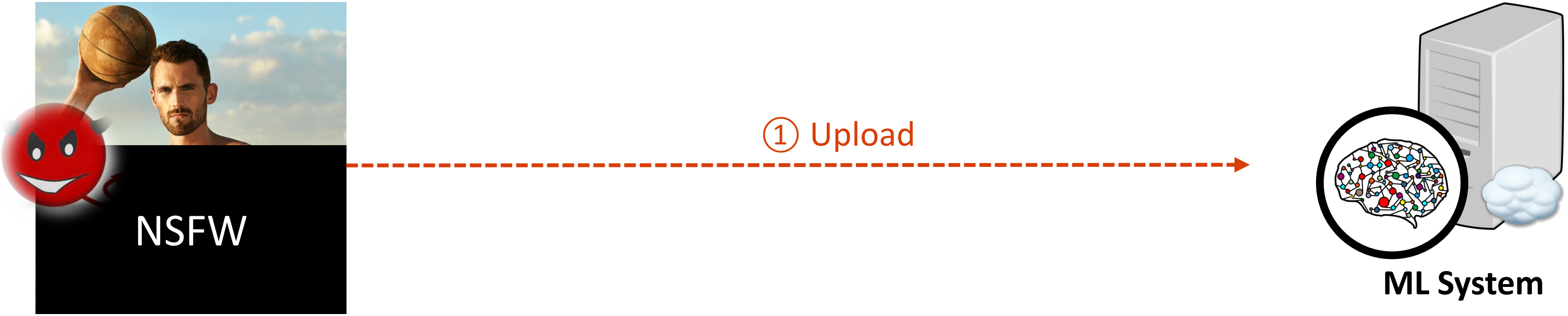
SAIL
Secure AI Systems Lab

ADVERSARIAL ~~EXAMPLES~~ ATTACKS

- Test-time (evasion) attack
  - Given a test-time sample x
  - Craft an adversarial example x^* that fools the target neural network

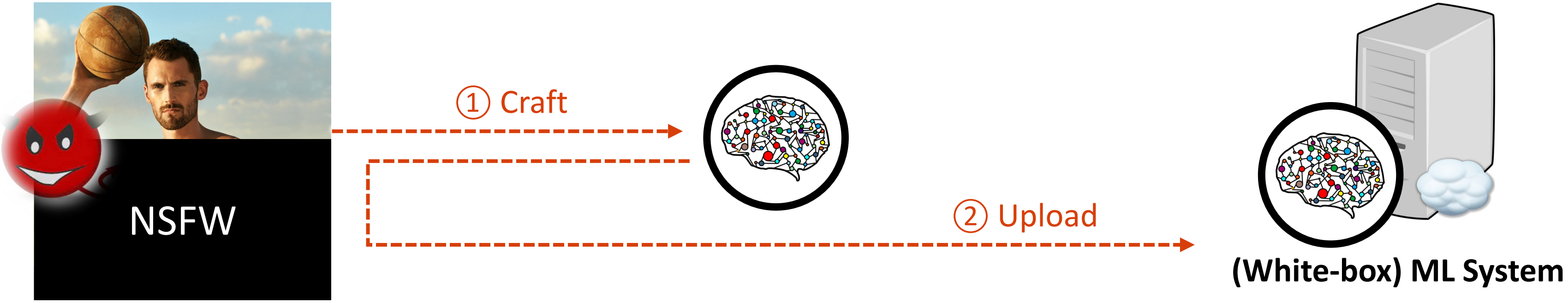
Adversarial attacks

- Example: An adversary wants to upload NSFW image to the cloud



White-box adversarial attacks

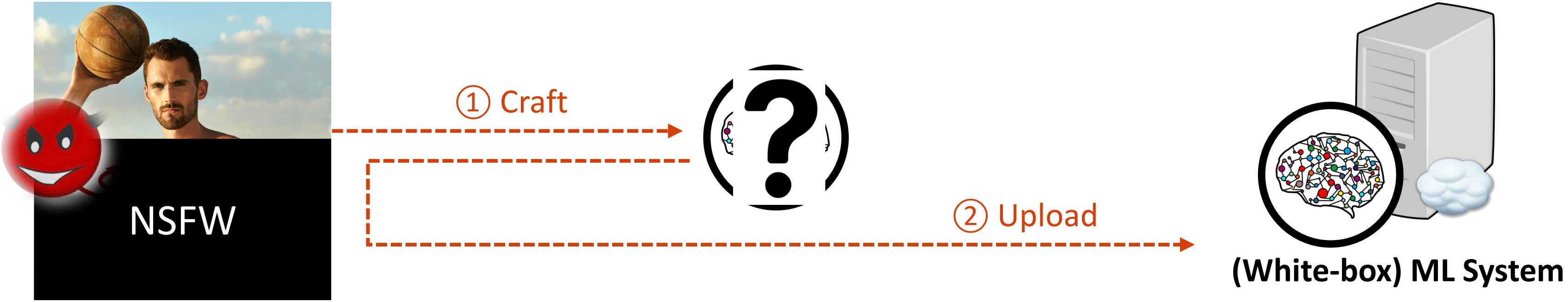
- Example: An adversary wants to upload NSFW image to the cloud



  - **FGSM, C&W, PGD, …:** the attacker has *complete* access to the target model

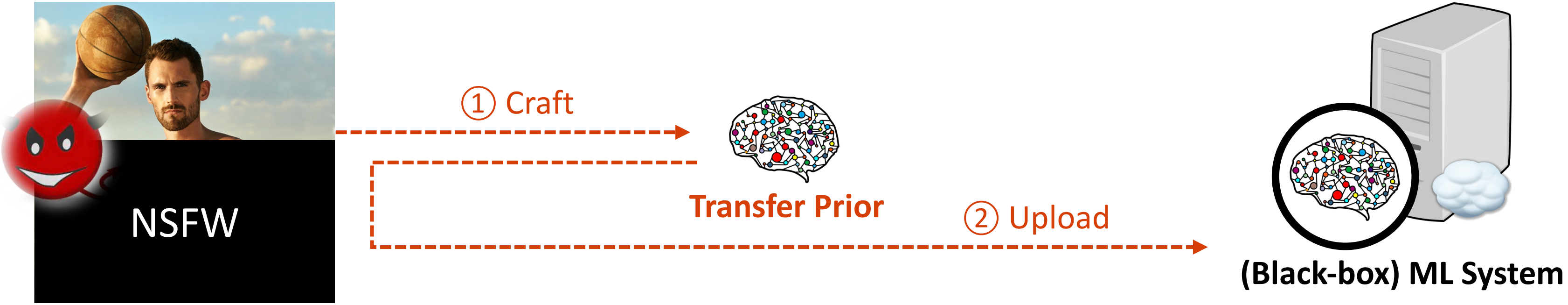
Black-box adversarial attacks

- Example: An adversary wants to upload NSFW image to the cloud



(Transfer-based) black-box adversarial attack

- Example: An adversary wants to upload NSFW image to the cloud

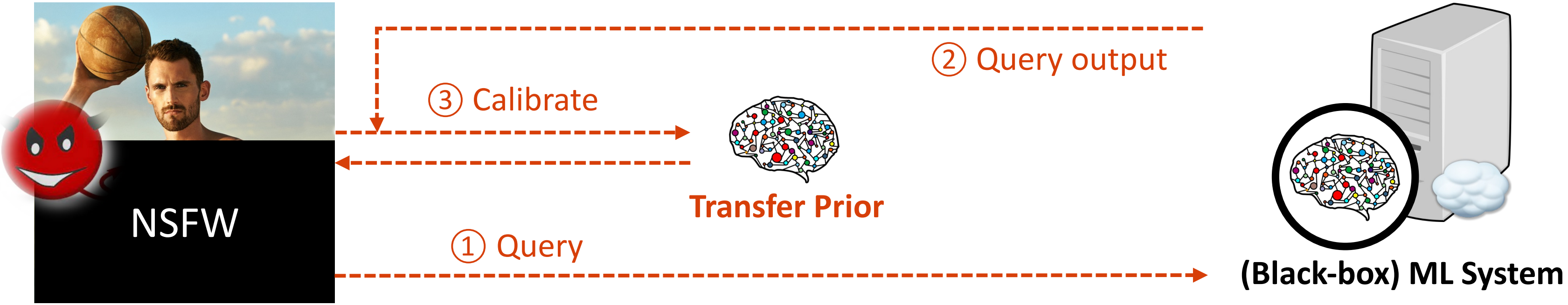


  - **Transfer-based attacks**[12] : craft adv. examples on a transfer prior

[1] Goodfellow *et al.*, *Explaining and Harnessing Adversarial Examples*, ICLR 2015
[2] Madry *et al.*, *Towards Deep Learning Models Resistant to Adversarial Attacks*, ICLR 2018

(Optimization-based) black-box adversarial attack

- Example: An adversary wants to upload NSFW image to the cloud



  - **Transfer-based attacks**[1][2] : craft adv. examples on a transfer prior
  - **Optimization-based attacks**[3] : craft them iteratively with query outputs and a transfer prior

[1] Goodfellow *et al.*, *Explaining and Harnessing Adversarial Examples*, ICLR 2015

[2] Madry *et al.*, *Towards Deep Learning Models Resistant to Adversarial Attacks*, ICLR 2018

[3] Cheng *et al.*, *Improving Black-box Adversarial Attacks with a Transfer-based Prior*, NeurIPS 2019

Today we talk about transfer-based attacks

Delving into transferable adversarial examples and black-box attacks, Liu et al., ICLR 2017

Transfer-based adversarial attacks

- Research questions
  - How well do adversarial examples transfer?
  - How practical are the transfer-based attacks?
  - What factors influence the transferability?
  - How can we reduce the transferability?

How well do adversarial examples transfer?

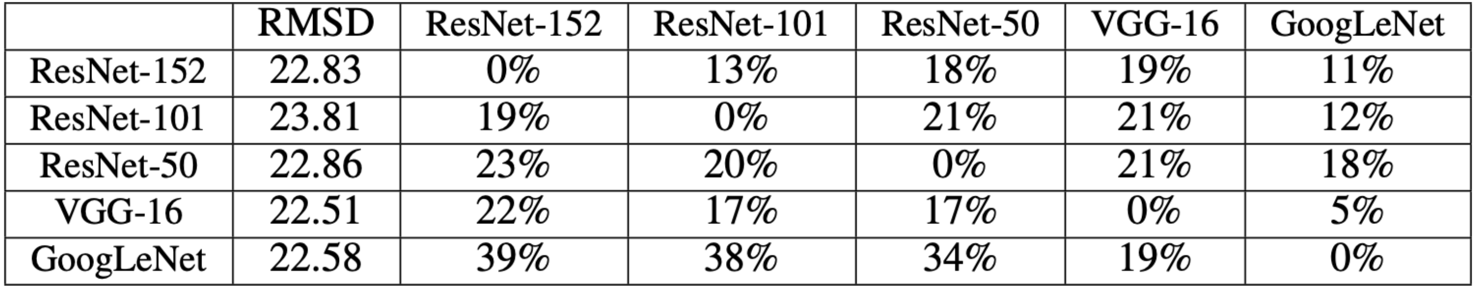
- Empirical evaluation
  - Train two models on a dataset
  - Craft adversarial examples on a model A (targeted and non-targeted)
  - Measure the success of these examples on the other model B

- Setup
  - Choose 100 images randomly from the ImageNet test-set
  - Use ResNet-50/-101/-152, GoogleNet, and VGG-16 models
  - Matching rate and distortion (l_2-distance)

- Adversarial attacks
  - Optimization-based approach (similar to C&W)
  - Fast Gradient-based approach (similar to PGD)

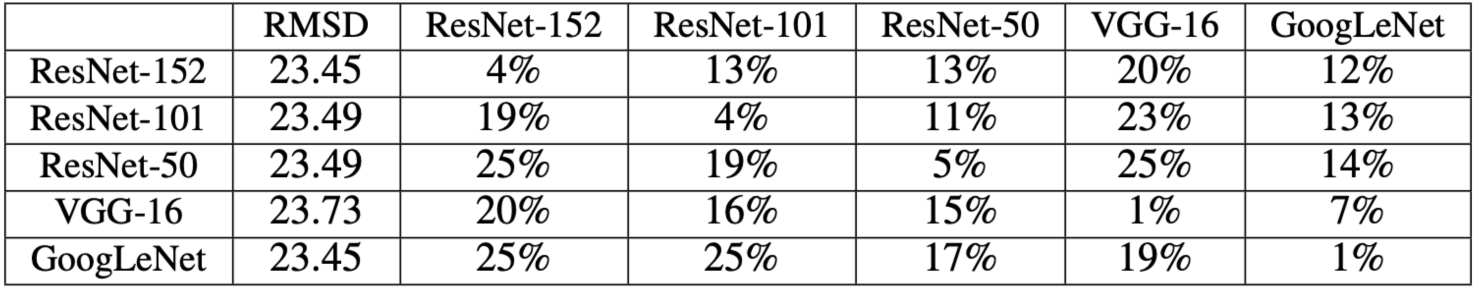
How well do adversarial examples transfer?

- Results from non-targeted attacks (Top-5 acc.)



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

Panel A: Optimization-based approach

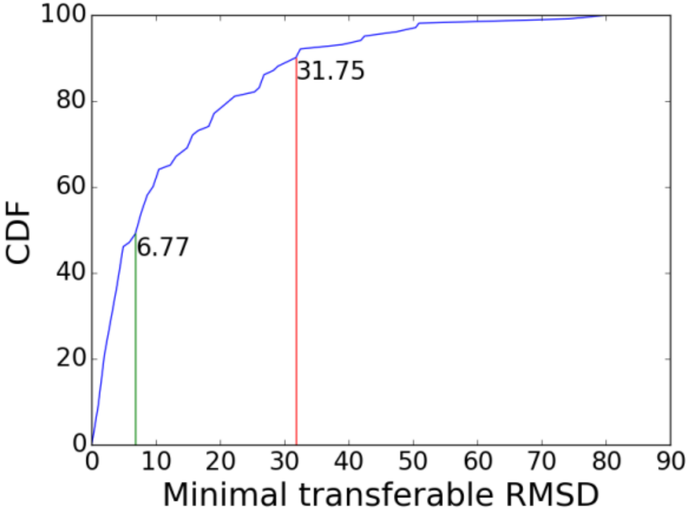


	RMSD	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
ResNet-152	23.45	4%	13%	13%	20%	12%
ResNet-101	23.49	19%	4%	11%	23%	13%
ResNet-50	23.49	25%	19%	5%	25%	14%
VGG-16	23.73	20%	16%	15%	1%	7%
GoogLeNet	23.45	25%	25%	17%	19%	1%

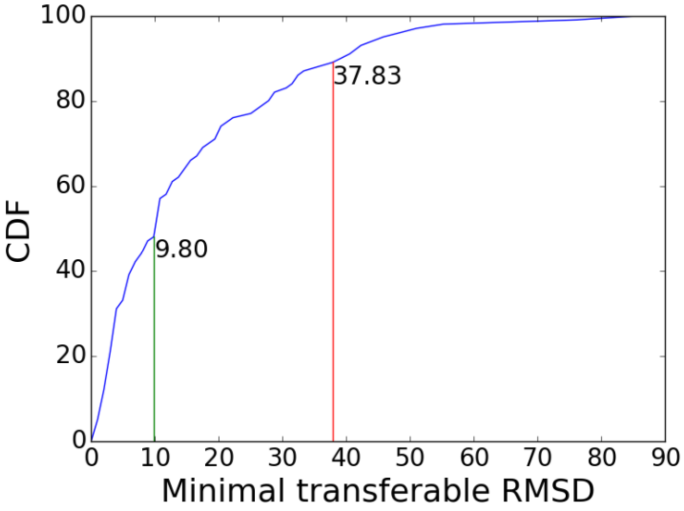
Panel B: Fast gradient approach

How well do adversarial examples transfer?

- More distortion leads to successful attacks?
  - Setup: VGG-16 to ResNet-152



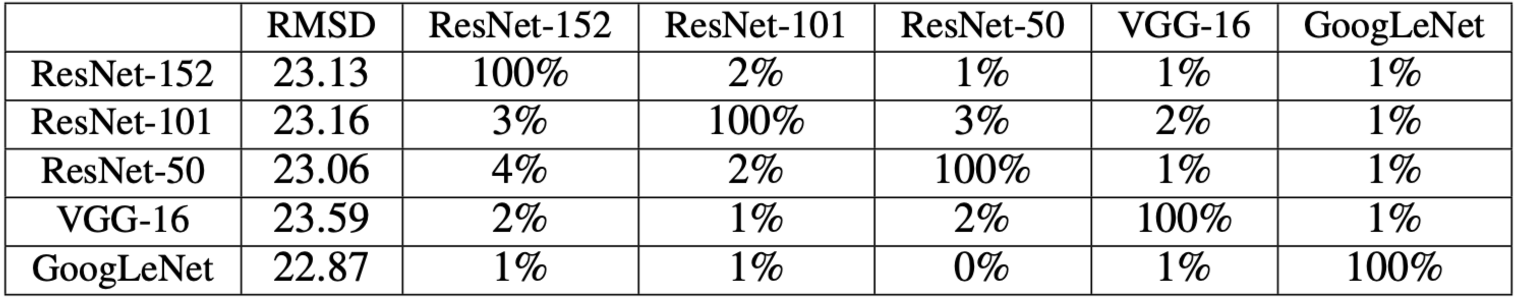
(a) Fast Gradient



(b) Fast Gradient Sign

How well do adversarial examples transfer?

- Results from targeted attacks (Matching rate)



	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%

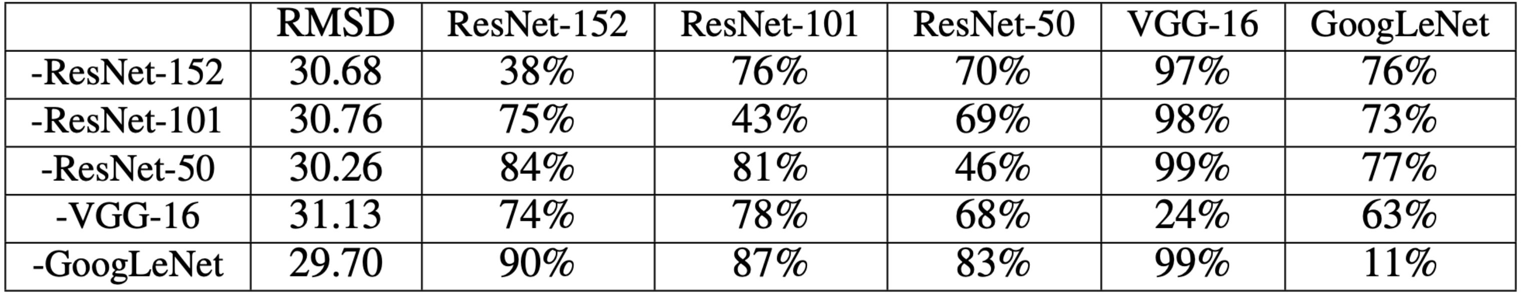
  - What if we use just random perturbations? Does *not* transfer

How well do adversarial examples transfer?

- Take aways
  - Non-targeted adversarial attacks transfer
  - Targeted adversarial attacks does not transfer well
  - Sub-research question: How we can make targeted attacks transferable?

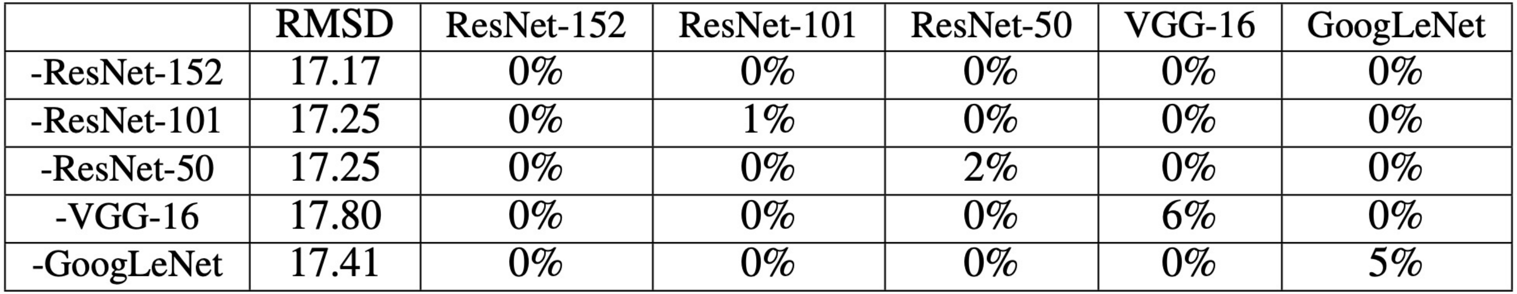
Improving transferability of targeted attacks

- “Ensemble” (Used optimization-based attacks)



	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%

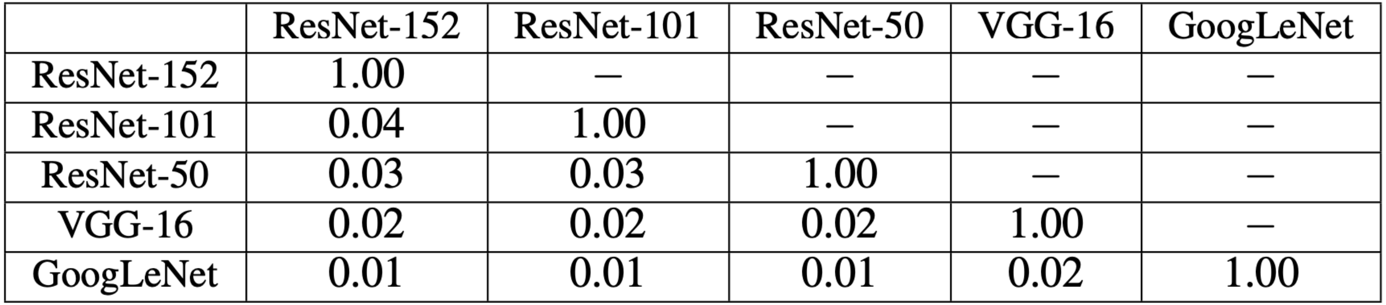
  - What about non-targeted attacks?



	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%

Improving transferability of targeted attacks

- Why does ensemble work?
  - Hypothesis: it makes computed gradients are aligned to that of the target model
  - Evaluation approach
    - Compute the gradients of inputs from the models
    - Compute the cosine similarity between the gradients from two different models
  - Results



	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
ResNet-152	1.00	—	—	—	—
ResNet-101	0.04	1.00	—	—	—
ResNet-50	0.03	0.03	1.00	—	—
VGG-16	0.02	0.02	0.02	1.00	—
GoogLeNet	0.01	0.01	0.01	0.02	1.00

How practical are the transfer-based attacks?

- Method
  - Craft adversarial examples on ImageNet models
  - Use them to fool the object recognition service in Clarifai.com (~~You can do as well~~)

- Setup
  - Choose 100 images randomly from the ImageNet test-set
  - Use models: ResNet-50/-101, GoogleNet and VGG-16
  - Matching rate

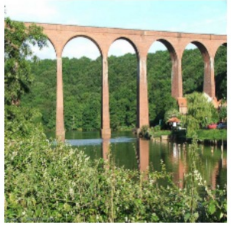
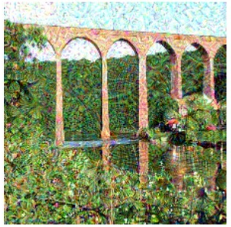
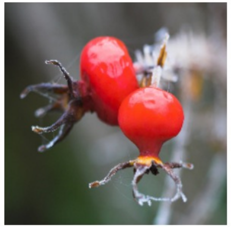
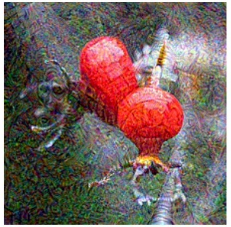
- Attacks
  - Optimization-based approach (similar to C&W)

How practical are the transfer-based attacks?

- Transfer attack results
    - Non-targeted:
        - Most attacks transfer (= fooled Clarifai.com)
            - 57% AEs crafted on VGG-16 transfer
            - 76% AEs crafted on the ensemble transfer
    - Targeted:
        - Misclassification **towards a target label**
            - 2% AEs crafted on VGG-16 transfer
            - 18% AEs crafted on the ensemble transfer

How practical are the transfer-based attacks?

- Transfer attack results

original image	true label	Clarifai.com results of original image	target label	targeted adversarial example	Clarifai.com results of targeted adversarial example
	viaduct	bridge, sight, arch, river, sky	window screen		window, wall, old, decoration, design
	hip, rose hip, rosehip	fruit, fall, food, little, wildlife	stupa, tope		Buddha, gold, temple, celebration, artistic

Transfer-based adversarial attacks

- Research questions
  - How well do adversarial examples transfer?
  - How practical are the transfer-based attacks?
  - What factors influence the transferability?
  - How can we reduce the transferability?

Why do adversarial attacks transfer?

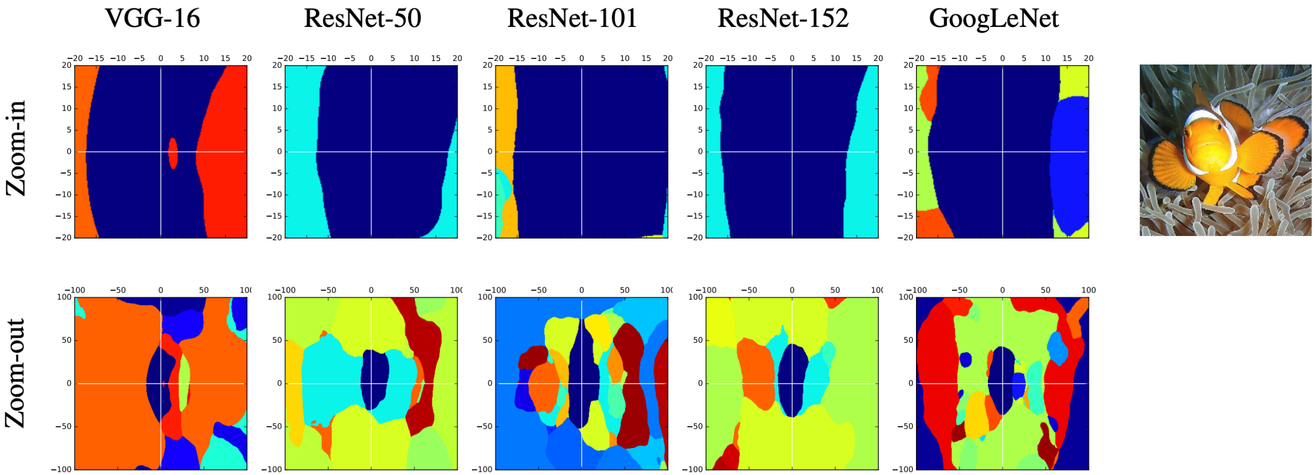
The space of transferable adversarial examples, Tramer et al.
Why do adversarial attacks transfer, Demontis et al., USENIX Security 2019

Why do adversarial attacks transfer?

- How to answer this question?
  - Inspect a model's decision boundary (Liu et al., Tramer et al.)
  - Inspect the data distribution (Tramer et al.)
  - Comprehensive empirical evaluation (Demotis et al.)
  - …

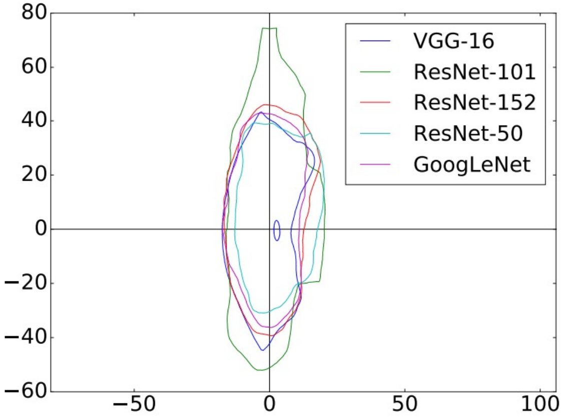
Why do adversarial examples transfer?

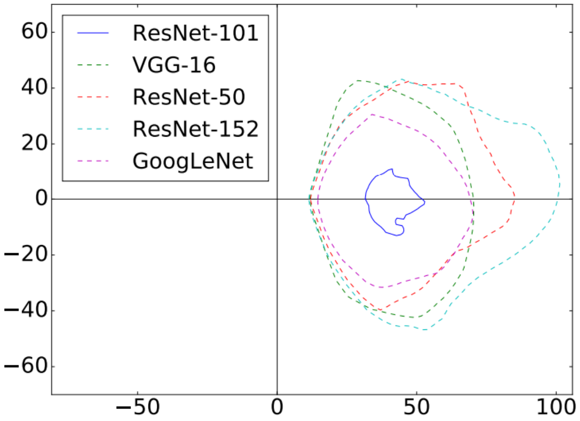
- Inspect a model's decision boundary
  - Setup:
    - Take a sample image, and two orthogonal gradient directions
    - Perturb the sample along each direction and measure the labels
  - Results



Why do adversarial examples transfer?

- Inspect a model’s decision boundary: ensemble
  - Setup:
    - Take a sample image, and two orthogonal gradient directions
    - Perturb the sample along each direction and measure the labels
  - Results





Why do adversarial examples transfer?

- Inspect a model's decision boundary: subspace
  - Setup:
    - Take a sample image, and *multiple* orthogonal gradient directions
    - Perturb the sample along each direction and measure the loss
  - Results

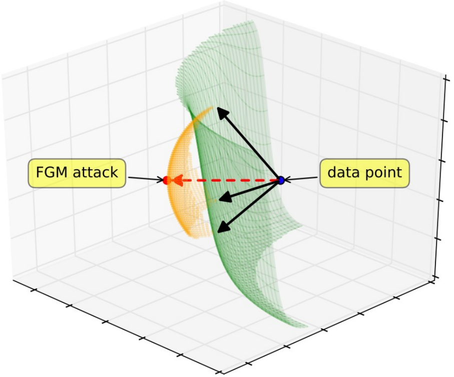


Figure 1: Illustration of the Gradient Aligned Adversarial Subspace (GAAS). The gradient aligned attack (red arrow) crosses the decision boundary. The black arrows are orthogonal vectors aligned with the gradient that span a subspace of potential adversarial inputs (orange).

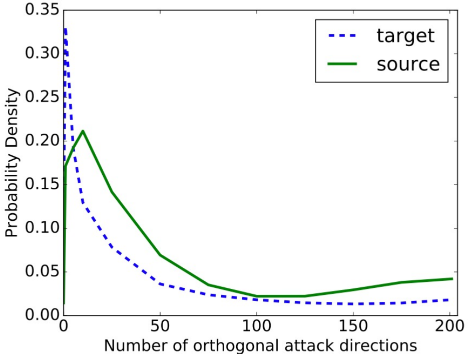


Figure 2: Probability density function of the number of successful orthogonal adversarial perturbations found by the GAAS method on the source DNN model, and of the number of perturbations that transfer to the target DNN model.

Thank You!

Tu/Th 4:00 – 5:50 pm

Instructor: **Sanghyun Hong**

https://secure-ai.systems/courses/MLSec/F23

Oregon State University

SAIL
Secure AI Systems Lab