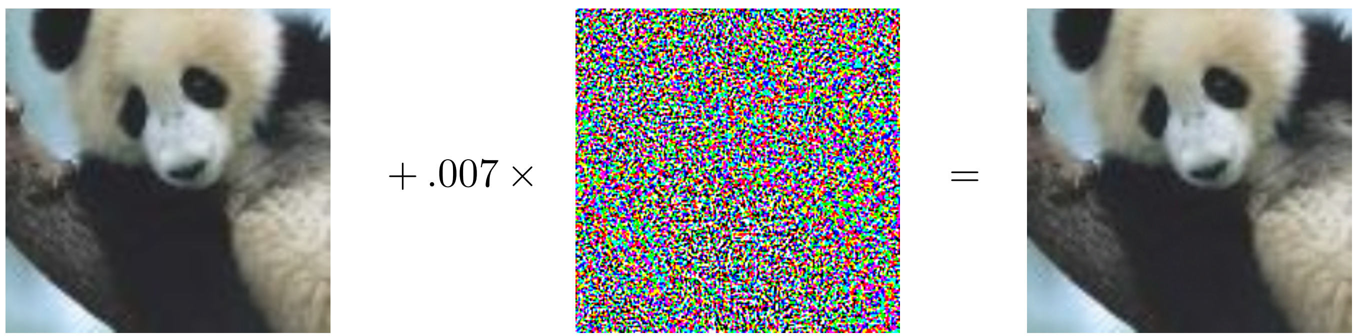
Ensemble Adversarial Training

Stanford Security Lunch
May 17th 2017

Florian Tramèr

Joint work with Alexey Kurakin, Nicolas Papernot, Dan Boneh & Patrick McDaniel

Adversarial Examples in ML



$+ .007 \times$ $=$

$\boldsymbol{x}$ "panda" 57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ "nematode" 8.2% confidence

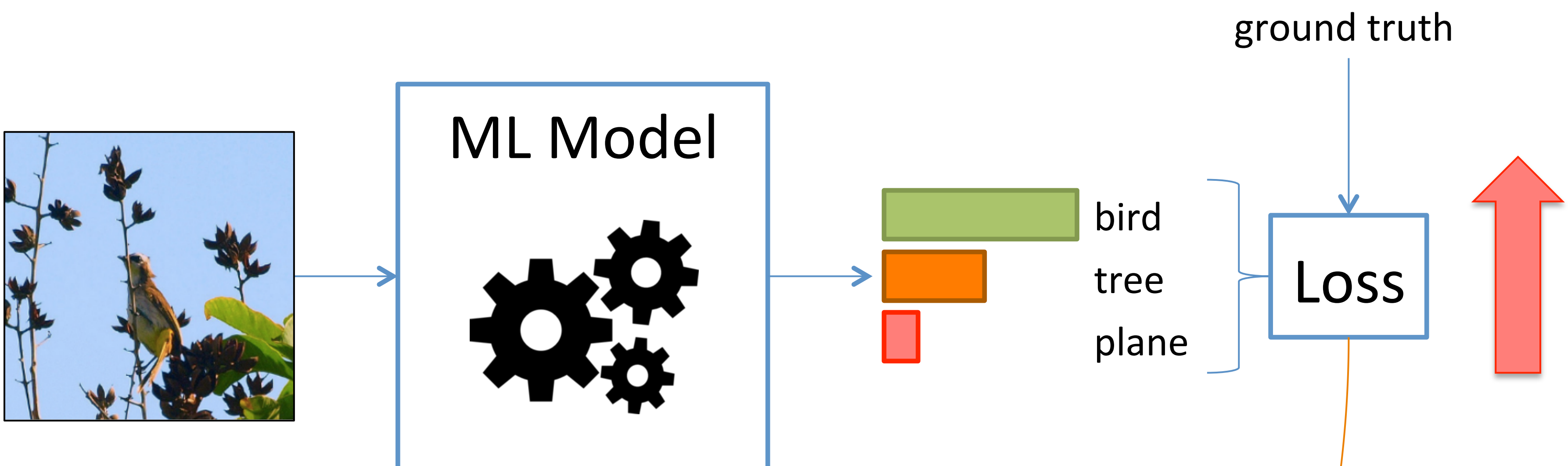
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ "gibbon" 99.3 % confidence

(Goodfellow et al. 2015)

Adversarial Examples in ML

- Images
  Szegedy et al. 2013, Nguyen et al. 2015, Goodfellow et al. 2015, Papernot et al. 2016, Liu et al. 2016, Kurakin et al. 2016, …
- Physical-World Attacks
  Sharif et al. 2016, Kurakin et al. 2017
- Malware
  Šrndić & Laskov 2014, Xu et al. 2016, Grosse et al. 2016, Hu et al. 2017
- Text Understanding
  Papernot et al. 2016
- Reinforcement Learning
  Huang et al. 2017, Lin et al. 2017, Behzadan & Munir 2017

Threat Model: White-Box Attacks

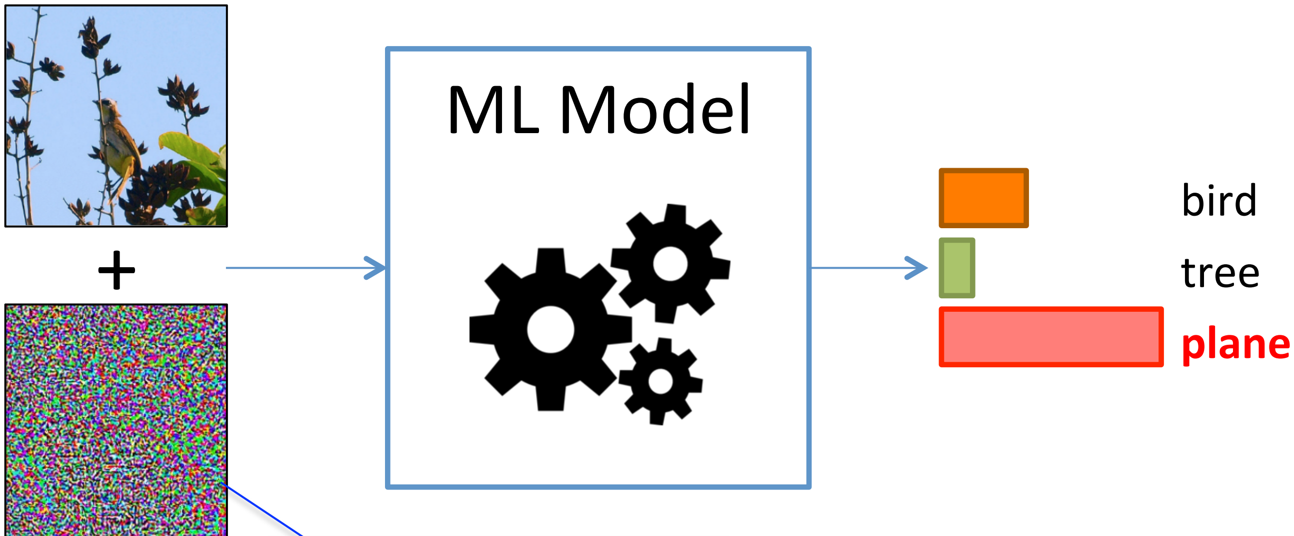


Take gradient
of the loss

"Fast Gradient Sign Method" (FGSM)

$$r = \epsilon \cdot \mathtt{sign}\left(\nabla_x J(x, y, \theta)\right)$$

Threat Model: White-Box Attacks

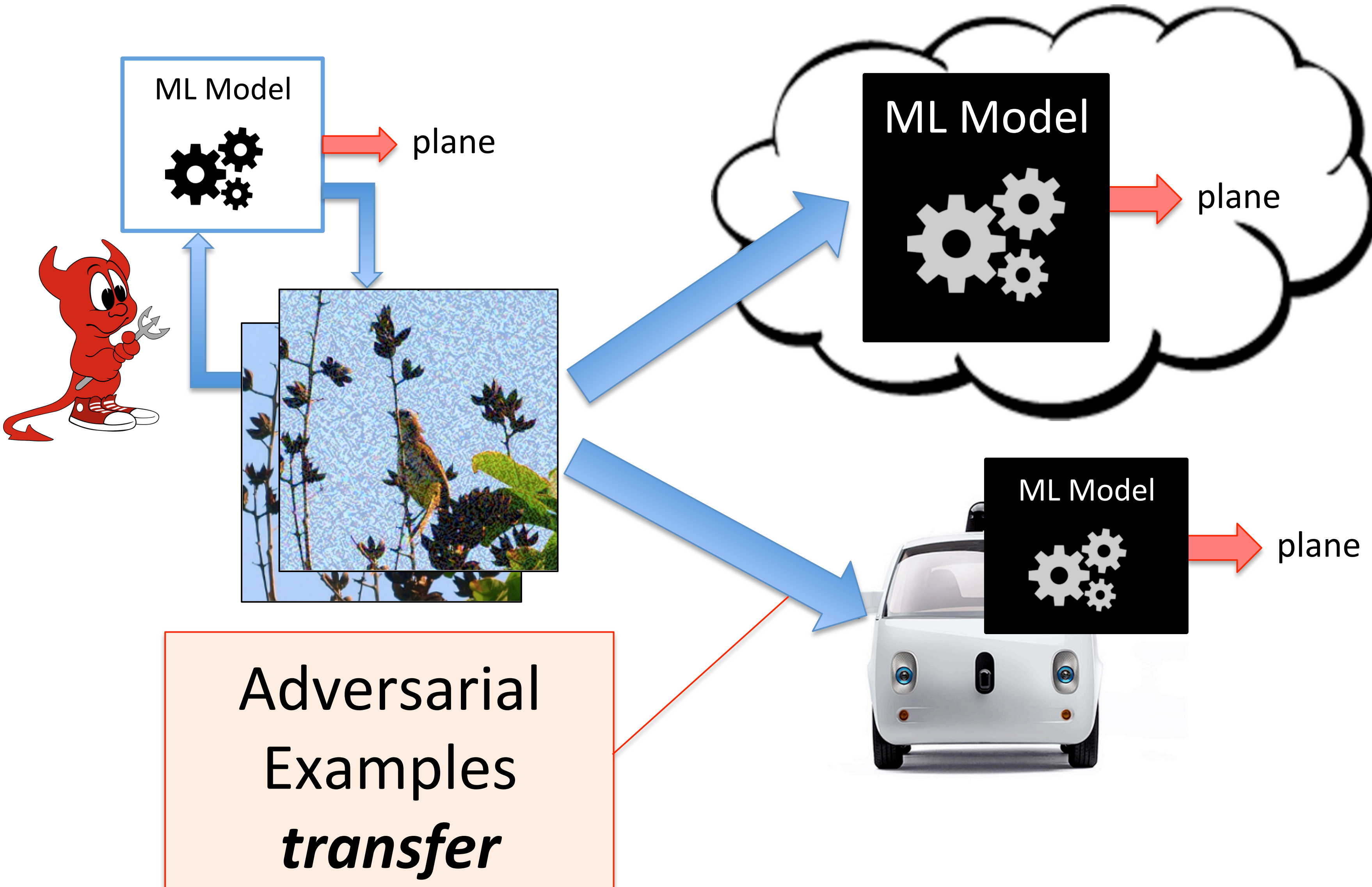


$$\|r\|_\infty = \epsilon$$

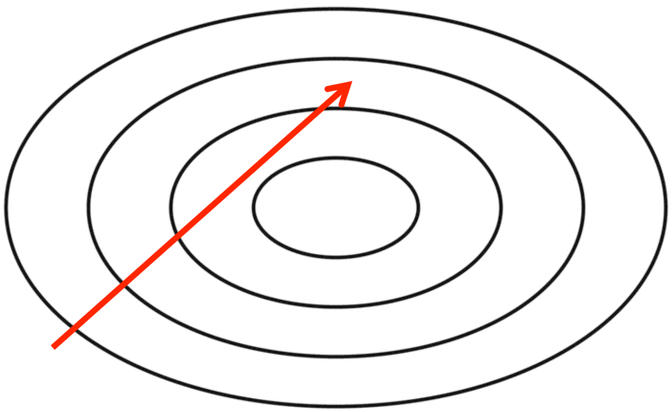
"Fast Gradient Sign Method" (FGSM)

$$r = \epsilon \cdot \texttt{sign}\left(\nabla_x J(x, y, \theta)\right)$$

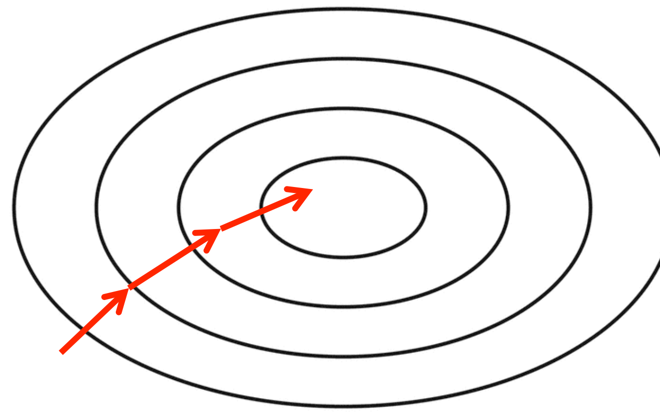
Threat Model: Black-Box Attacks



Iterative Attacks



“One-Shot” Attacks



“Iterative” Attacks

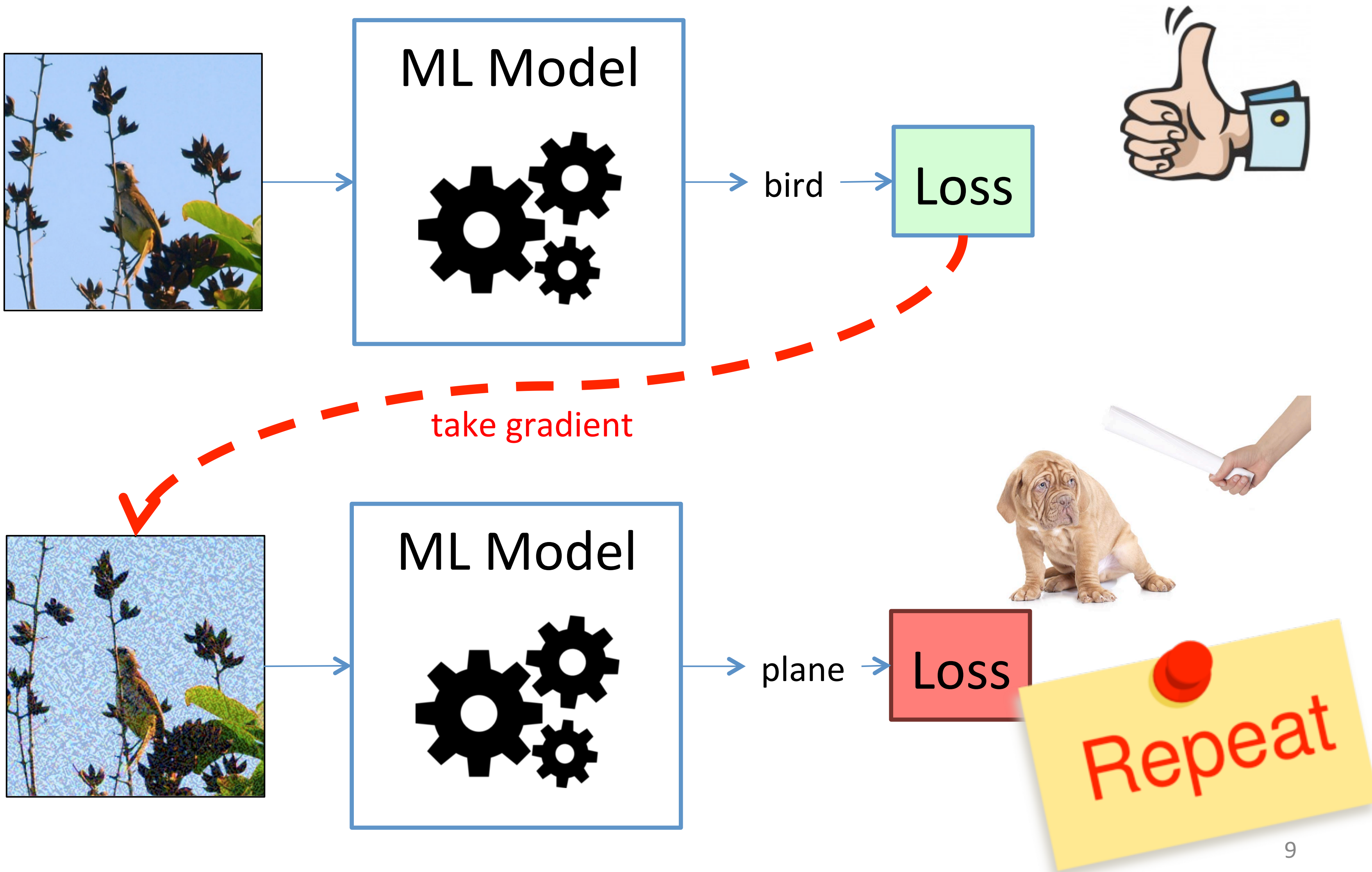
- Computationally **efficient**
- Weaker white-box attacks
- **Transfers with high probability**, strong black-box attacks!

- More Expensive
- **Close to 100% success rate** for imperceptible perturbations
- ***Overfits*** to model’s parameters / doesn’t transfer very well

Defenses?

- Ensembles? ✗
- Distillation? ✗
- Generative modeling? ✗
- Adversarial training? Lets see… ✗

Adversarial Training



Does it Work?

Adversarial Training	White-Box Attacks	Black-Box Attacks
One-Shot		
Iterative		

Does it Work?

Adversarial Training	White-Box Attacks	Black-Box Attacks
One-Shot	Mostly yes!	
Iterative		

Does it Work?

Adversarial Training	White-Box Attacks	Black-Box Attacks
One-Shot	Mostly yes!	
Iterative	Not really	

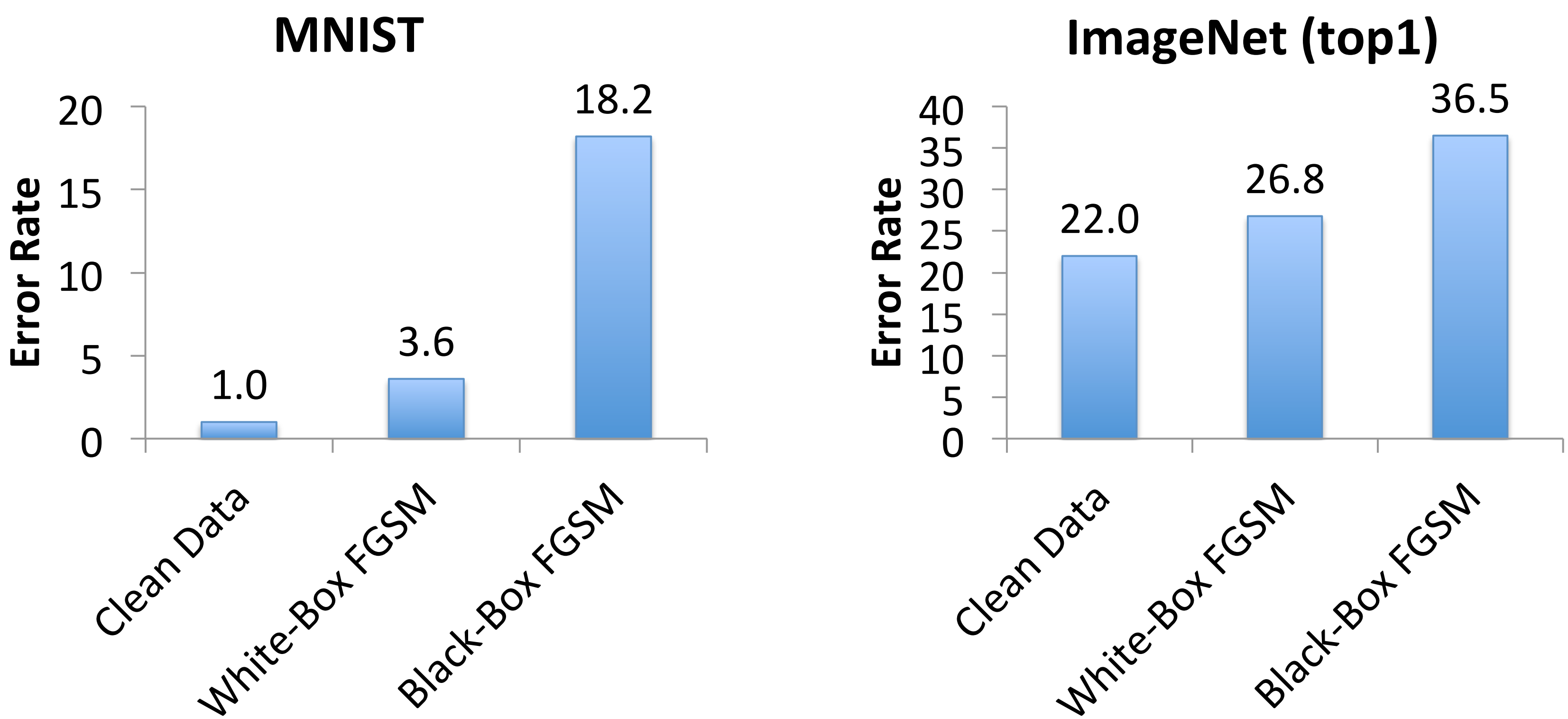
Does it Work?

Adversarial Training	White-Box Attacks	Black-Box Attacks
One-Shot	Mostly yes!	
Iterative	Not really	But they don't transfer much

Does it Work?

Adversarial Training	White-Box Attacks	Black-Box Attacks
One-Shot	Mostly yes!	Not really!
Iterative	Not really	But they don't transfer much

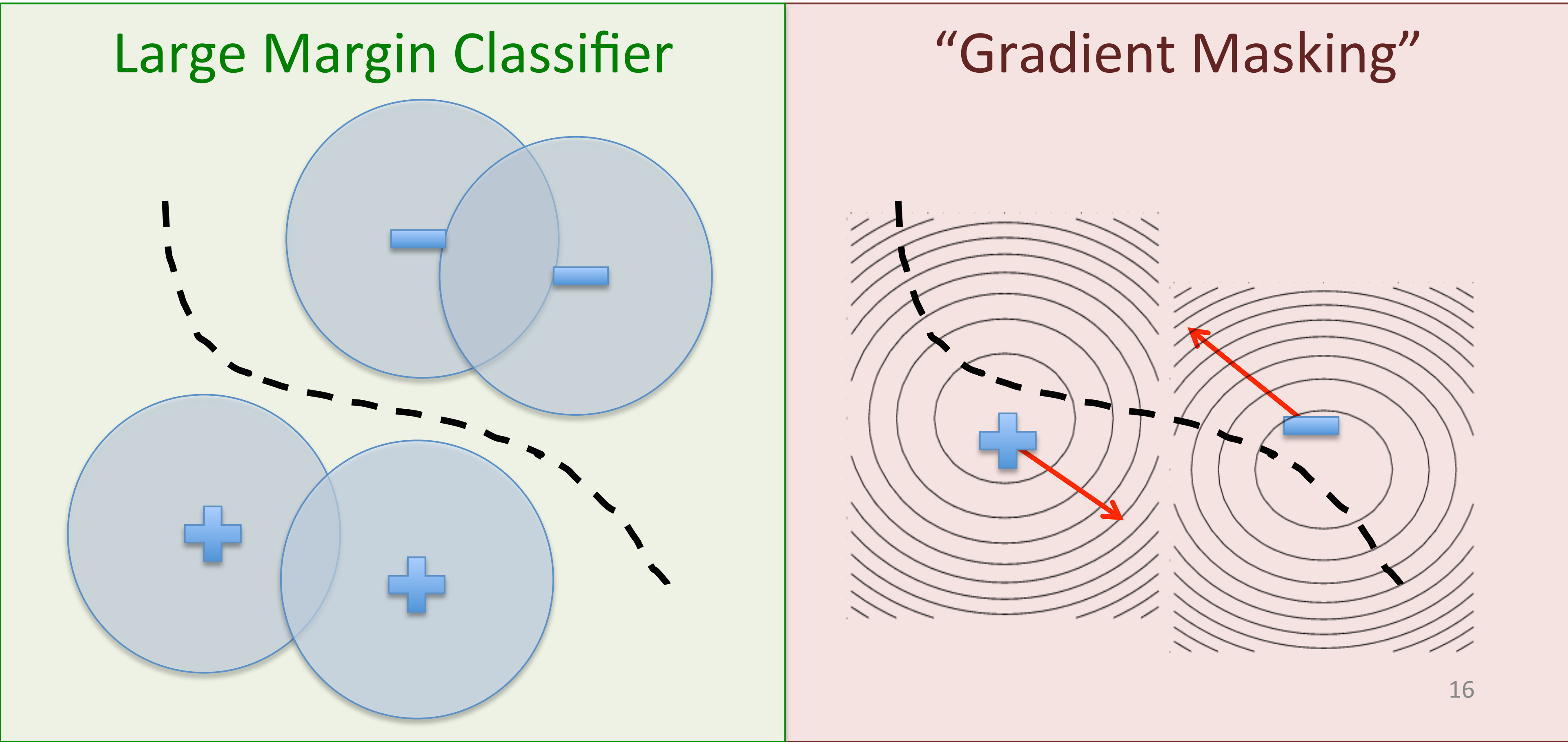
Attacks on Adversarial Training



Adversarial examples transferred from another model

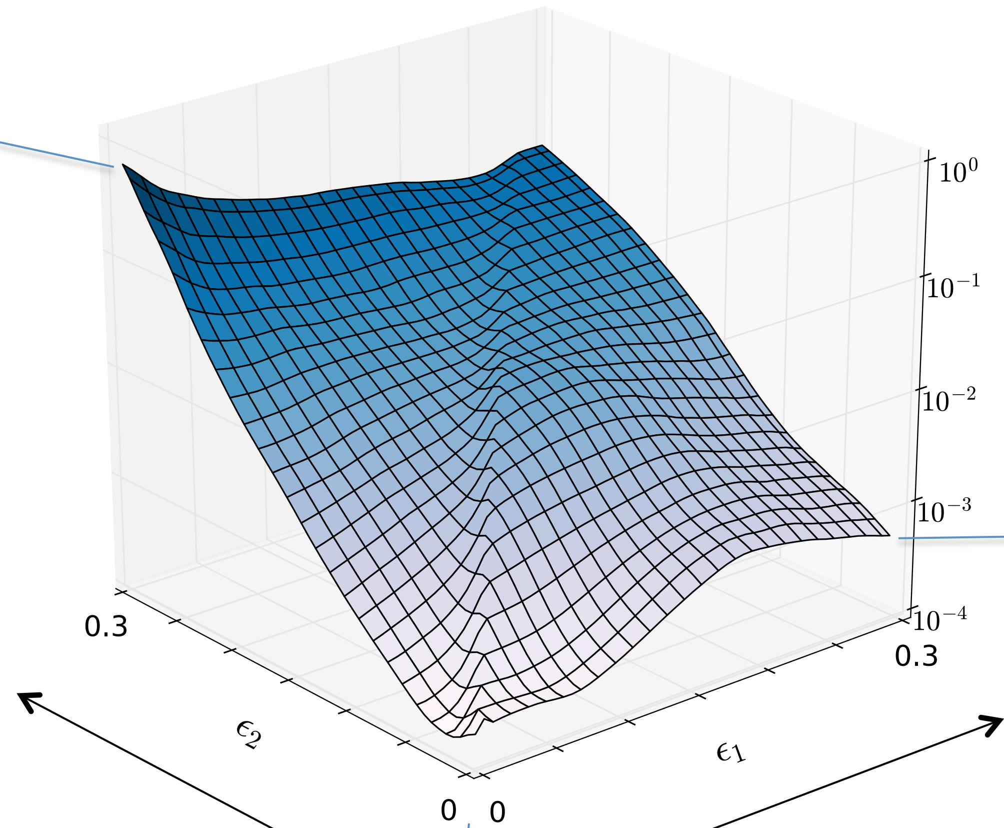
Gradient Masking

- How to get robustness to FGSM-style attacks?



Loss of Adversarially Trained Model

Adversarial Example



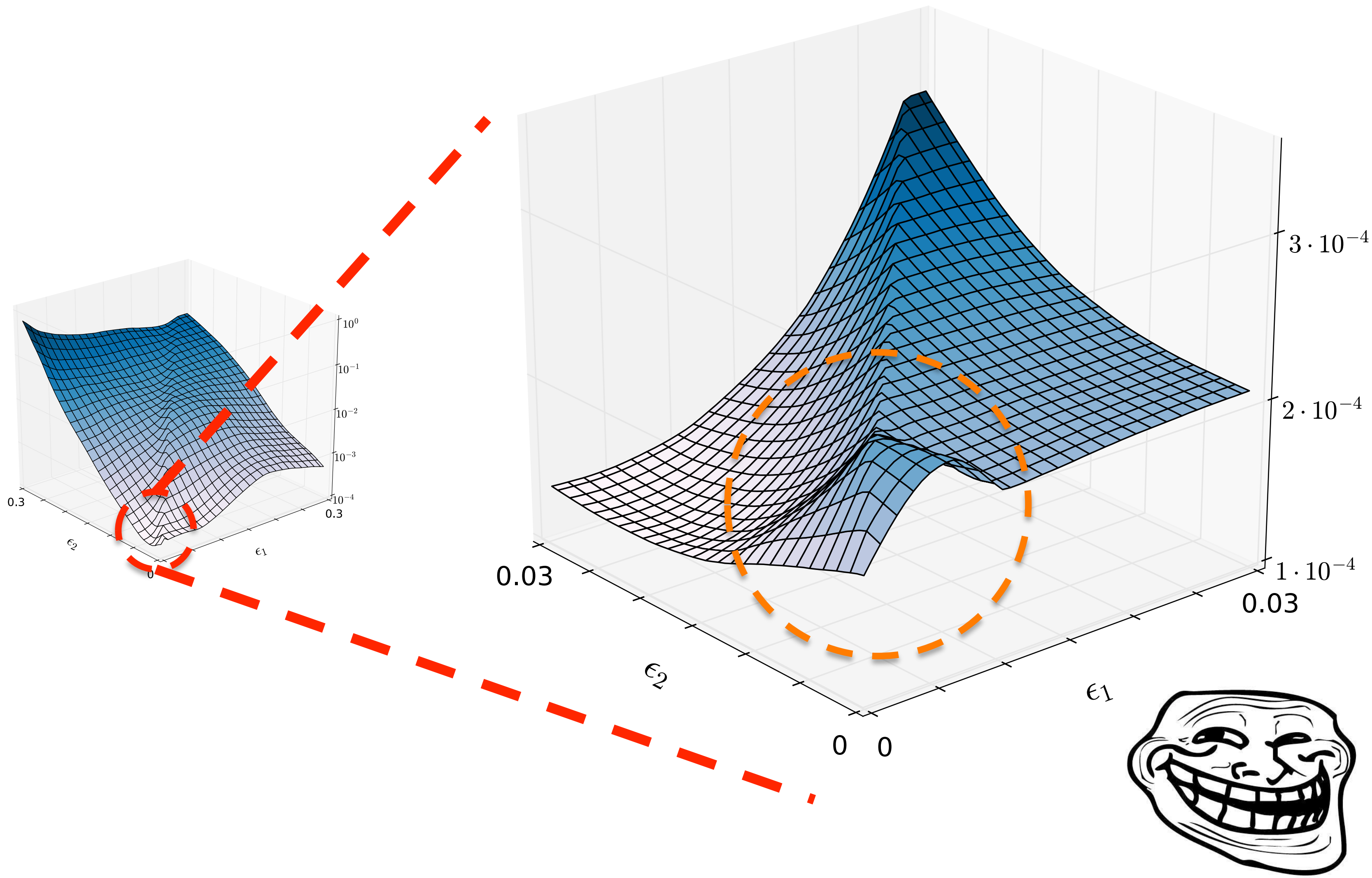
Non-Adversarial Example

Move in direction of another model's gradient (black-box attack)

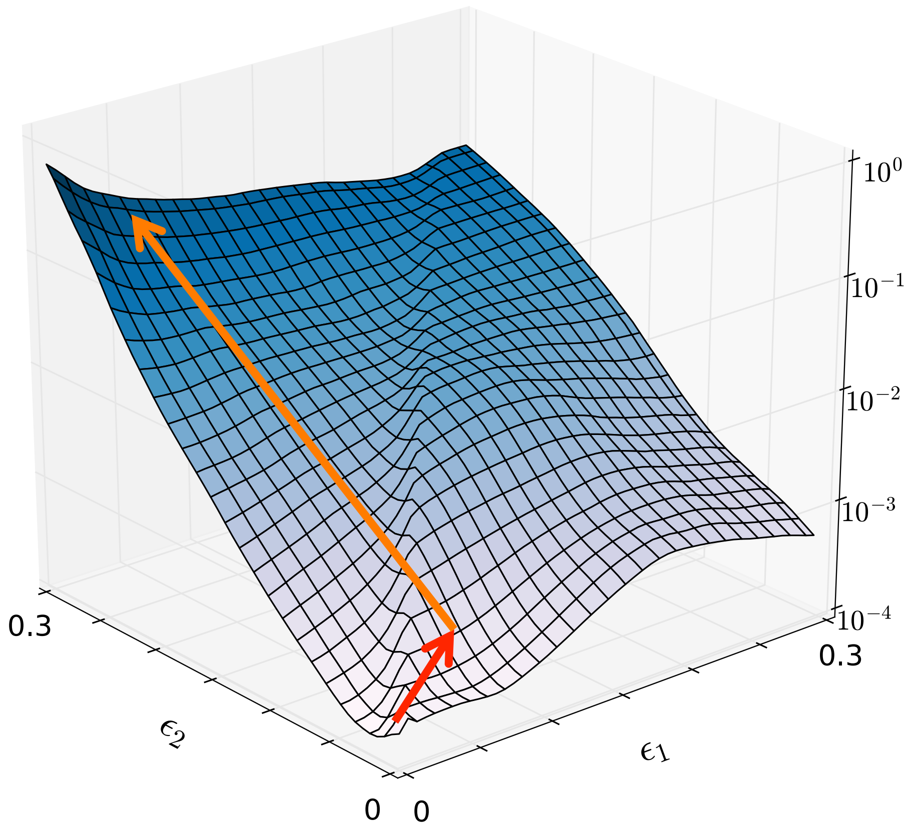
Move in direction of model's gradient (white-box attack)

Data Point

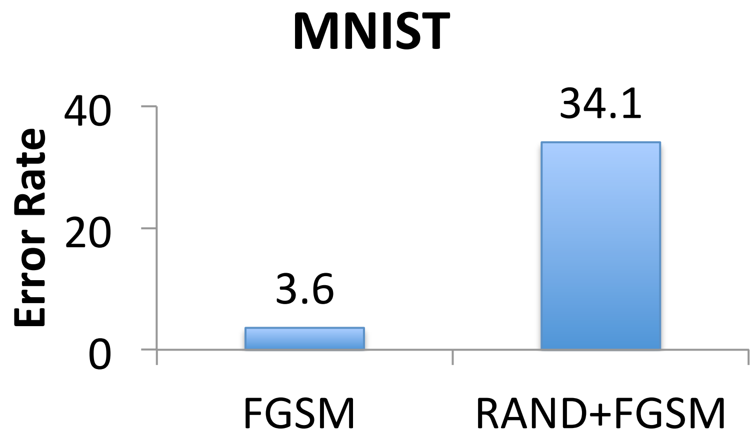
Loss of Adversarially Trained Model

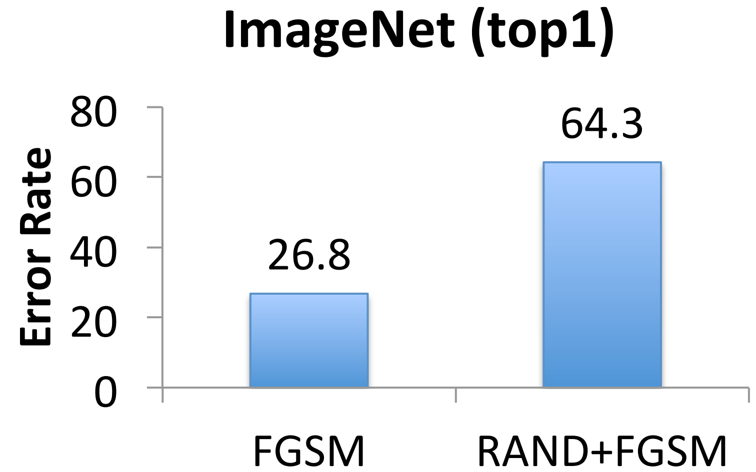


Simple Attack: RAND+FGSM



1. Small random step
2. Step in direction of gradient





Does it Work? (Before)

Adversarial Training	White-Box Attacks	Black-Box Attacks
One-Shot	Mostly yes!	Not really!
Iterative	Not really	But they don't transfer much

Does it Work? (Now)

Adversarial Training	White-Box Attacks	Black-Box Attacks
One-Shot	Not really!	Not really!
Iterative	Not really	But they don't transfer much

Security against white-box attacks seems out-of-reach.
Black-box security might be sufficient. Can we do better?

What's wrong with Adversarial Training?

- Minimize

$$\mathrm{loss}(x, y) + \underbrace{\mathrm{loss}(x + \epsilon \cdot \mathtt{sign}(\mathrm{grad}), y)}$$

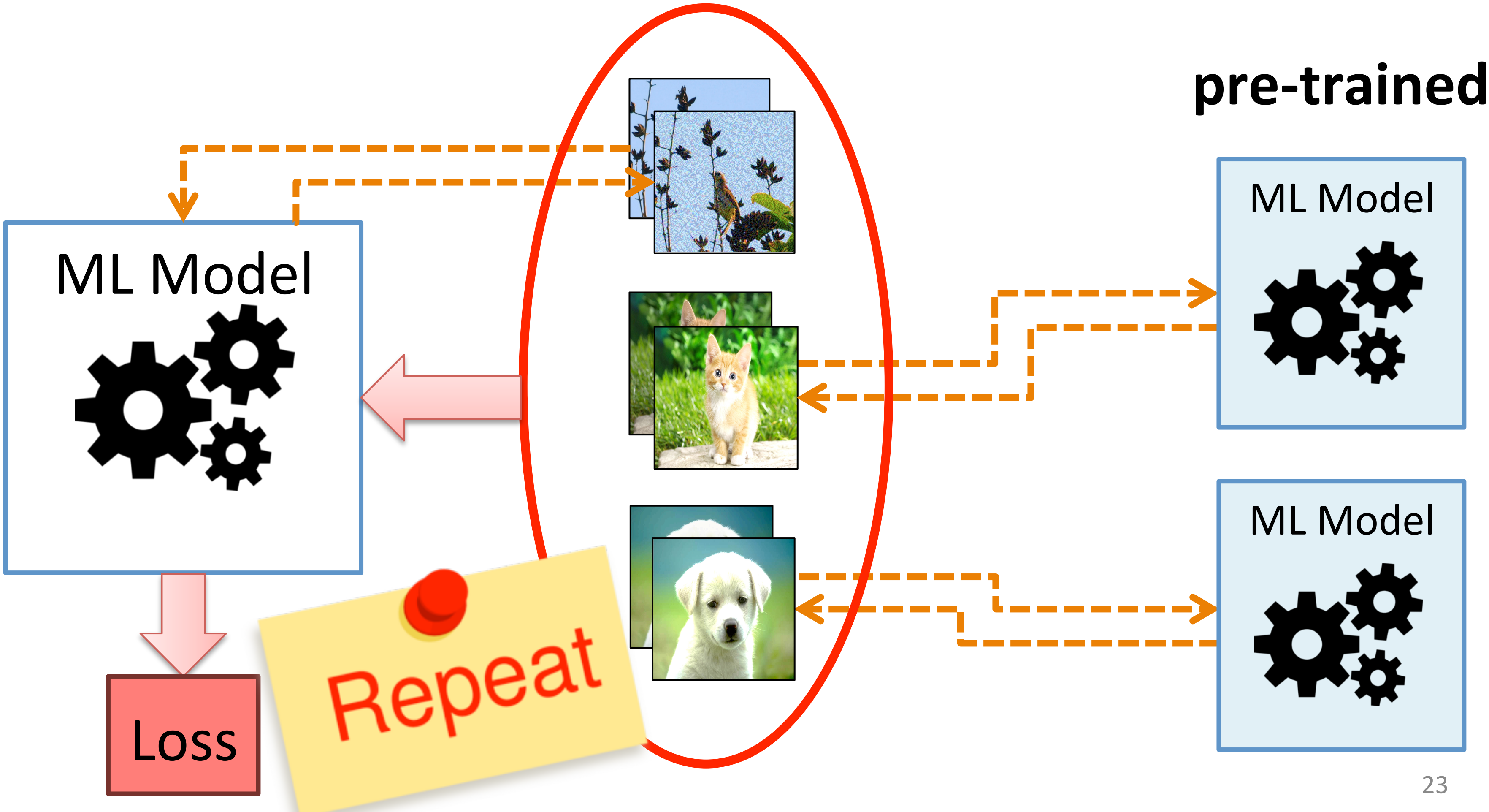
Small if:

1. The model is actually robust
2. *Or, the gradient points in a direction that is not adversarial*

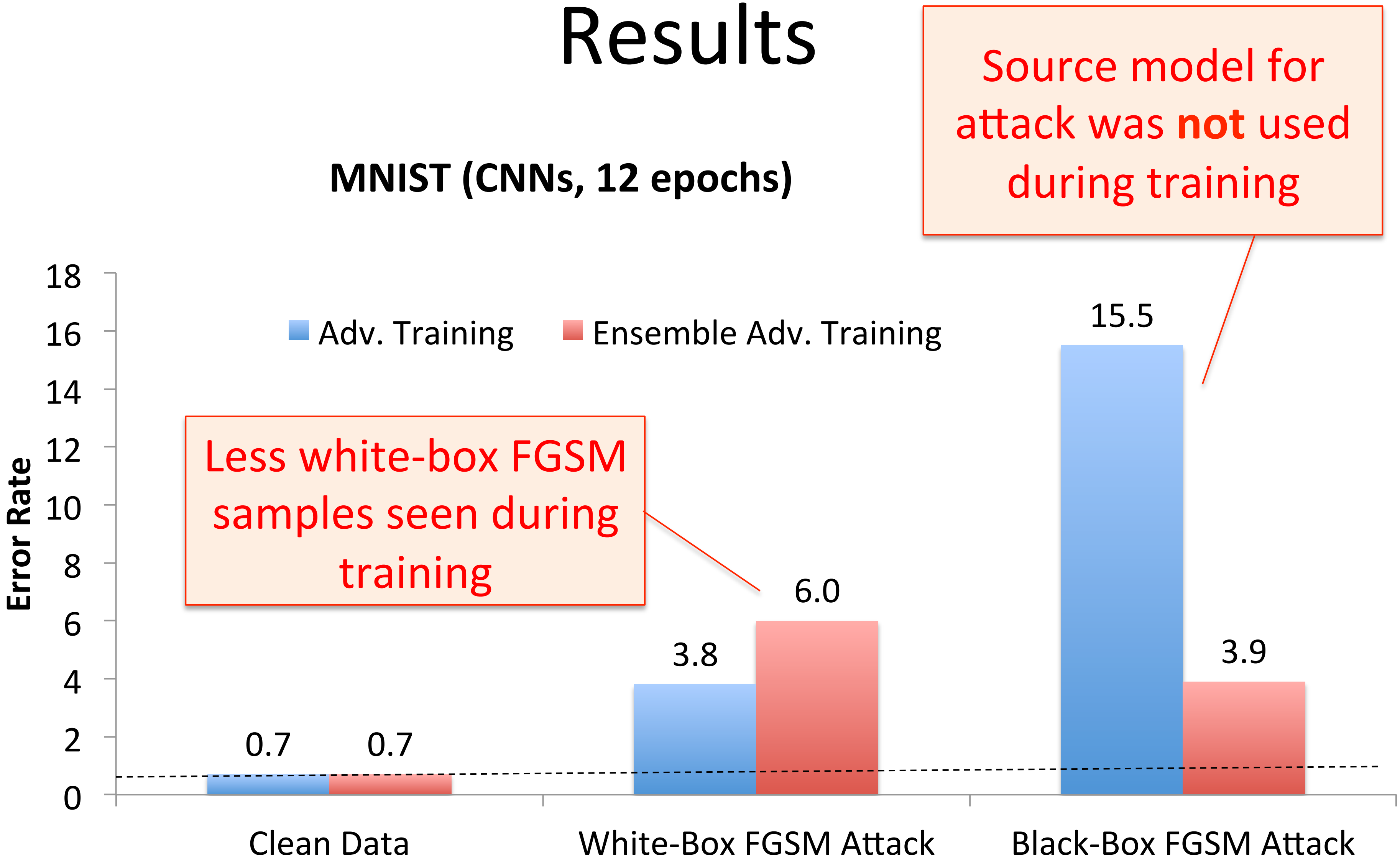
Degenerate Minimum

Ensemble Adversarial Training

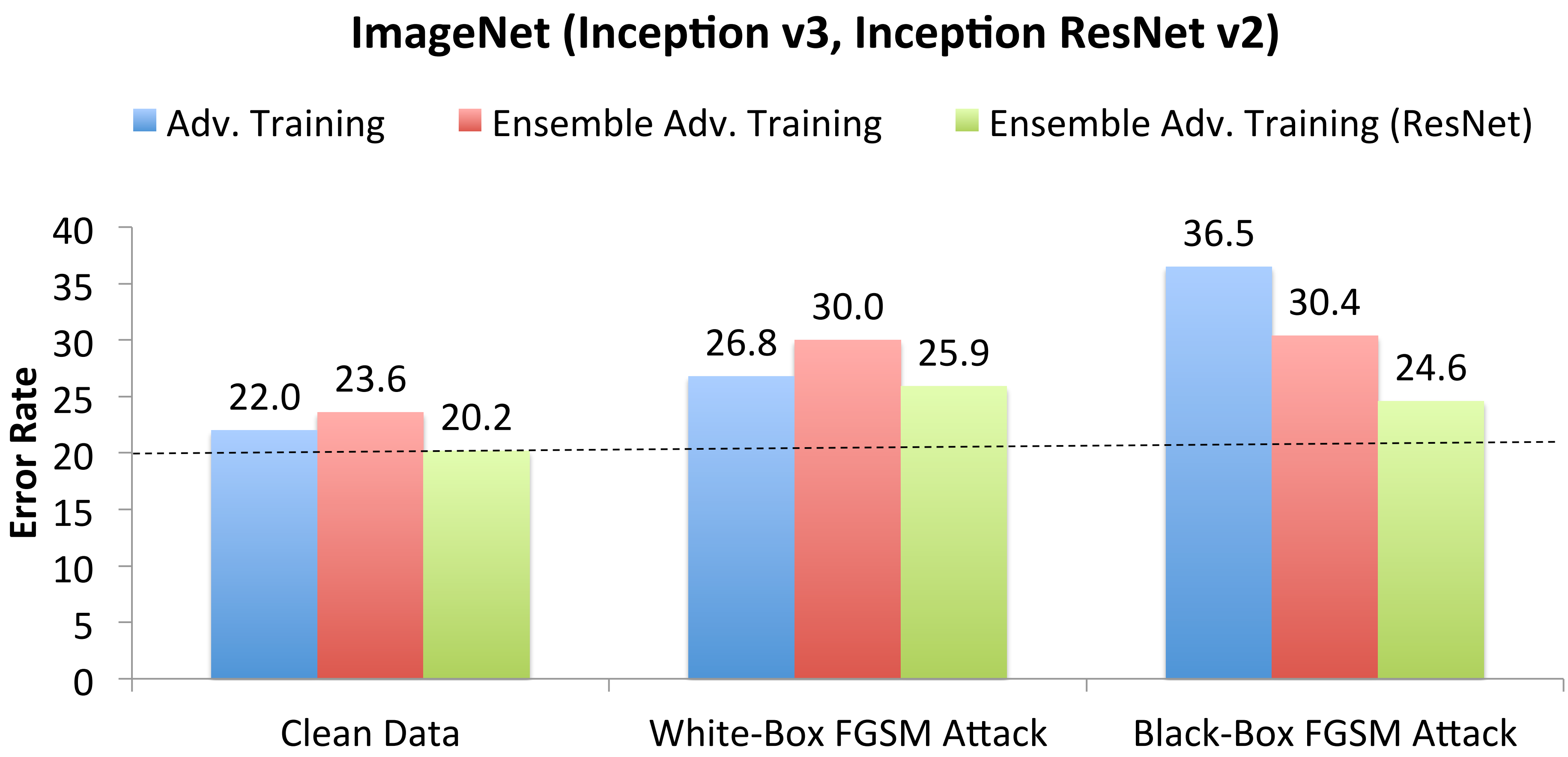
- How do we avoid these degenerate minima?



Results

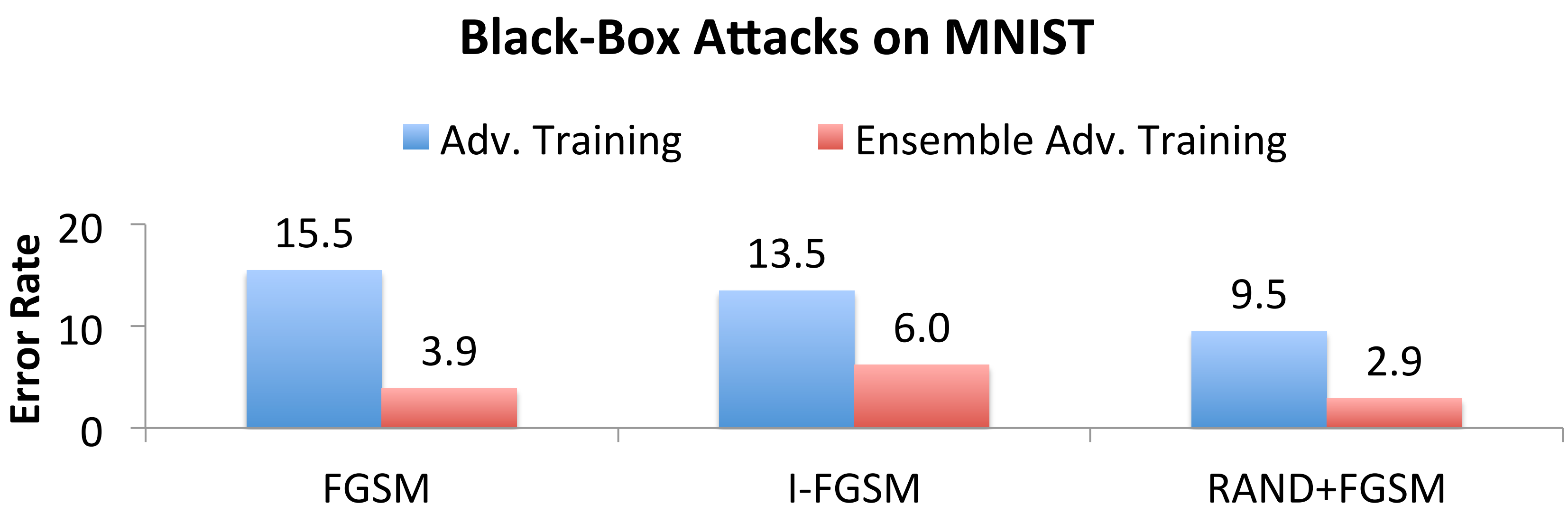

Source model for attack was **not** used during training
MNIST (CNNs, 12 epochs)
Error Rate
18
16
14
12
10
8
6
4
2
0
Adv. Training
Ensemble Adv. Training
Less white-box FGSM samples seen during training
0.7
0.7
3.8
6.0
15.5
3.9
Clean Data
White-Box FGSM Attack
Black-Box FGSM Attack

Results

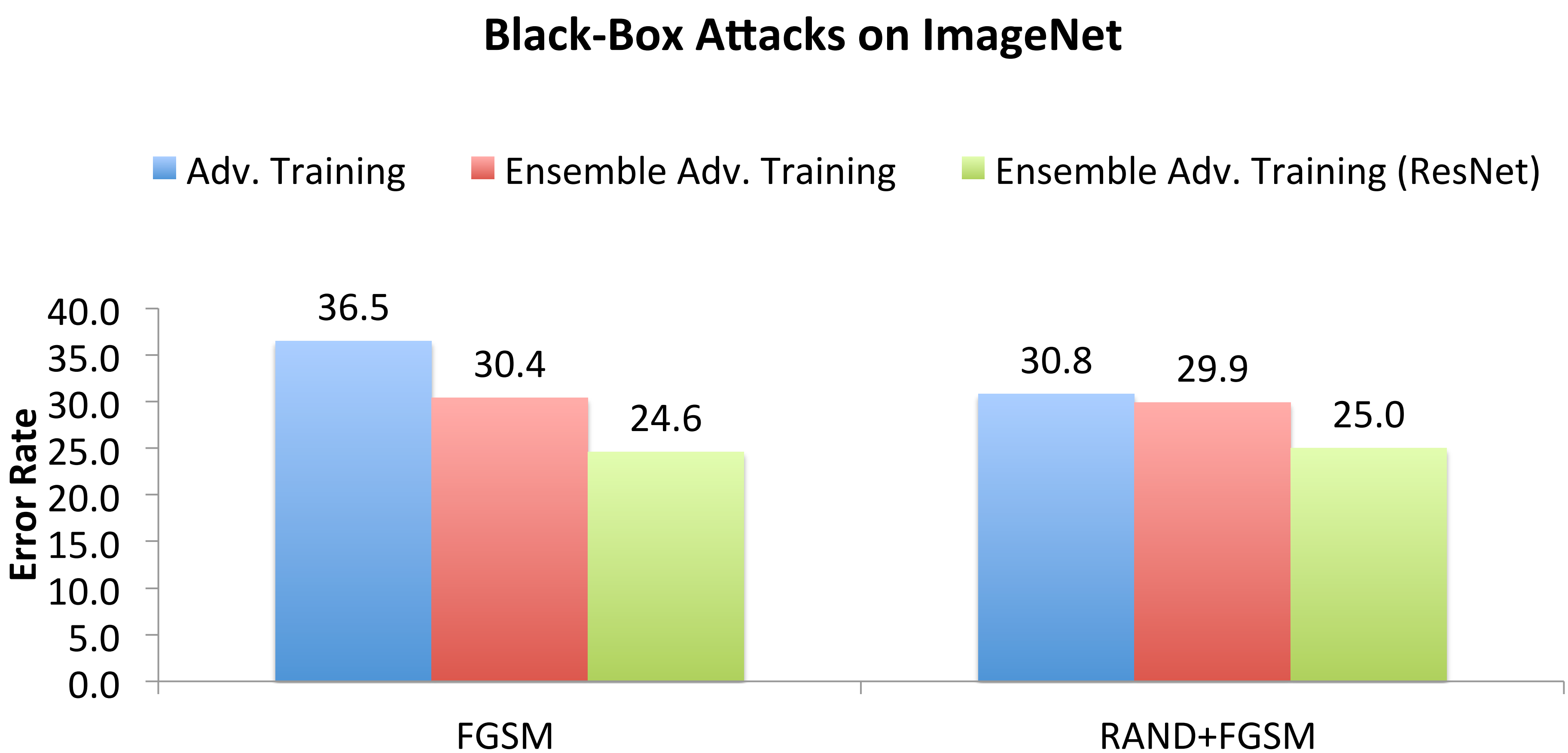

ImageNet (Inception v3, Inception ResNet v2)
Adv. Training
Ensemble Adv. Training
Ensemble Adv. Training (ResNet)
Error Rate
40
35
30
25
20
15
10
5
0
22.0
23.6
20.2
26.8
30.0
25.9
36.5
30.4
24.6
Clean Data
White-Box FGSM Attack
Black-Box FGSM Attack

What about stronger attacks?

- Little to no improvement on **white-box** iterative and RAND+FGSM attacks!
- But, **these attacks don't transfer** well!



What about stronger attacks?



Efficiency of Ensemble Adversarial Training

- **Pre-compute gradients** for pre-trained models
  - Lower per-batch cost than with adversarial training

- **Randomize source model** in each batch
  - If `num_models % num_batches = 0`, we see the same adversarial examples in each epoch if we just rotate

- **Convergence can be *much* slower**

Standard Inception v3:	~150 epochs
Adversarial training:	~190 epochs
Ensemble adversarial training:	**~280 epochs**

Maybe because the task is actually hard?...

Takeaways

- Test defenses on black-box attacks!
  - Distillation (Papernot et al. 2016, attack by Carlini et al. 2016)
  - Biologically Inspired Networks
    (Nayebi & Ganguli **27 Mar. 2017**, attack by Brendel & Bethge **5 Apr. 2017**)
  - Adversarial Training, and probably many others…
- 

  « If you don't know where to go, just move at random. »
  — *Morgan Freeman* — (*or Dan Boneh*)
- Ensemble Adversarial Training vastly improves robustness to black-box attacks

Open Problems

- Better black-box attacks?
  - How much does ***oracle access*** to the model help?

- More efficient ensemble adversarial training?

- Can we say **anything** formal (and useful) about adversarial examples?

THANK YOU