

# Adversarial Examples and Adversarial Training

Ian Goodfellow, OpenAI Research Scientist

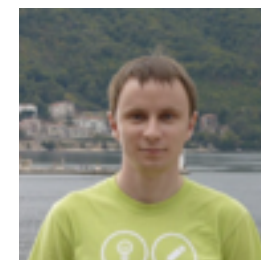
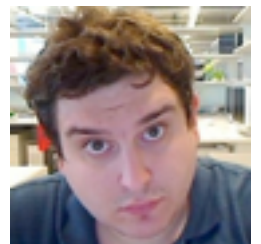
Presentation at HORSE 2016

London, 2016-09-19

OpenAI

# In this presentation

- “Intriguing Properties of Neural Networks” Szegedy et al, 2013
- “Explaining and Harnessing Adversarial Examples” Goodfellow et al 2014
- “Adversarial Perturbations of Deep Neural Networks” Warde-Farley and Goodfellow, 2016
- “Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples” Papernot et al 2016
- “Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples” Papernot et al 2016
- “Adversarial Perturbations Against Deep Neural Networks for Malware Classification” Grosse et al 2016 (**not my own work**)
- “Distributional Smoothing with Virtual Adversarial Training” Miyato et al 2015 (**not my own work**)
- “Virtual Adversarial Training for Semi-Supervised Text Classification” Miyato et al 2016
- “Adversarial Examples in the Physical World” Kurakin et al 2016

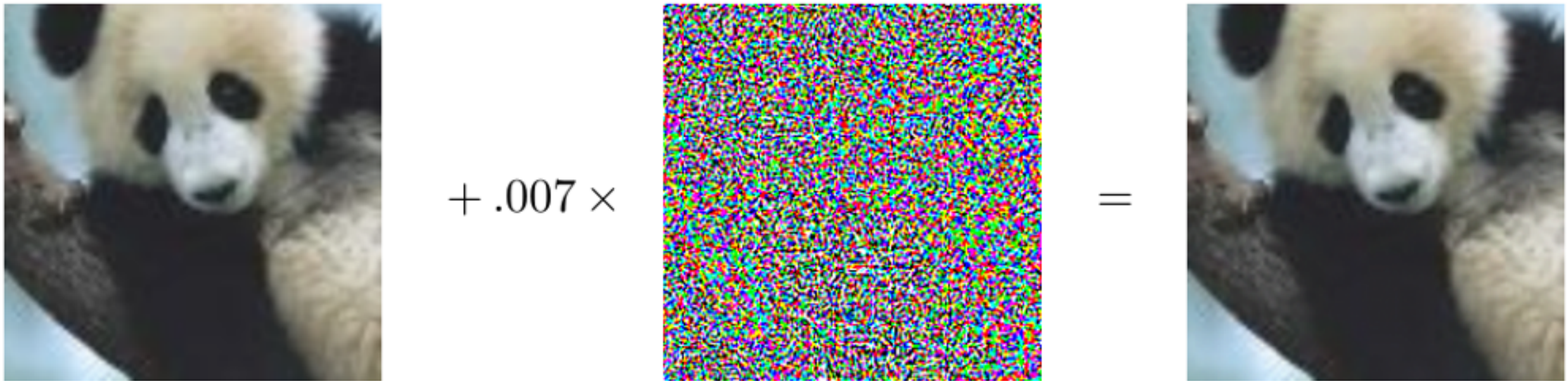


# Overview

- What causes adversarial examples?
- How can they be used to compromise machine learning systems?
- Adversarial training and virtual adversarial training
- New open source adversarial example library:

**cleverhans**

# Adversarial Examples



Timeline:

“Adversarial Classification” Dalvi et al 2004: fool spam filter

“Evasion Attacks Against Machine Learning at Test Time”

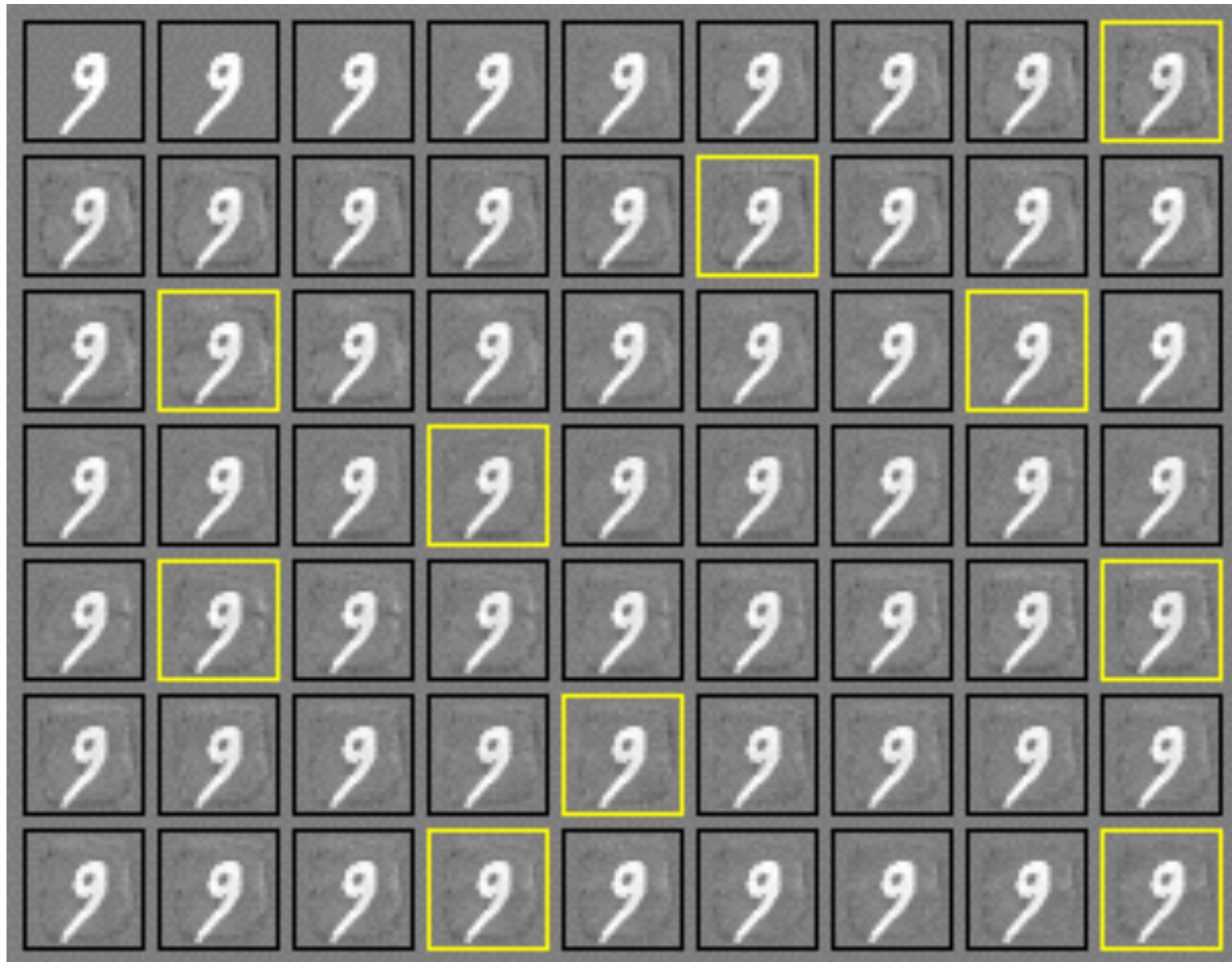
Biggio 2013: fool neural nets

Szegedy et al 2013: fool ImageNet classifiers imperceptibly

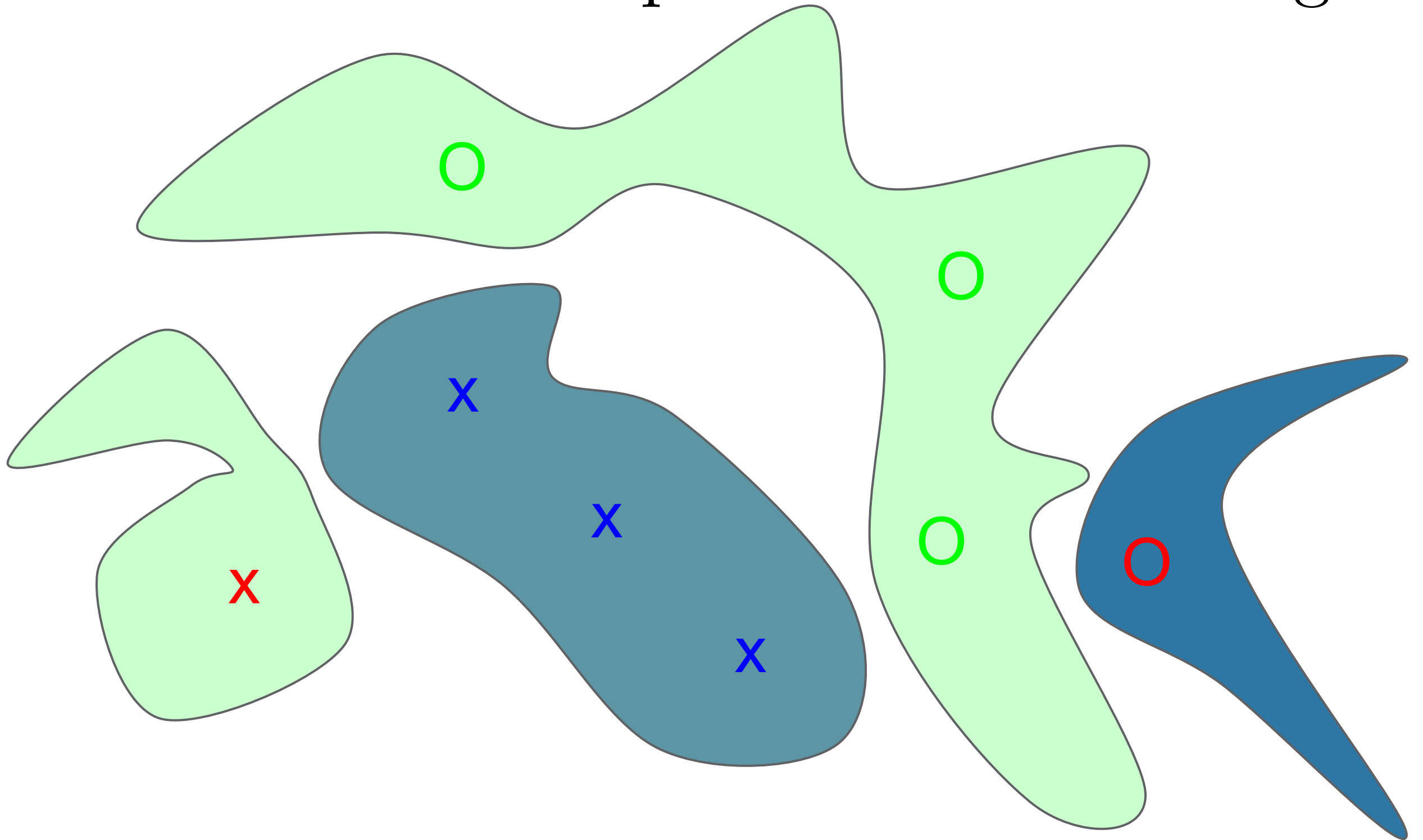
Goodfellow et al 2014: cheap, closed form attack



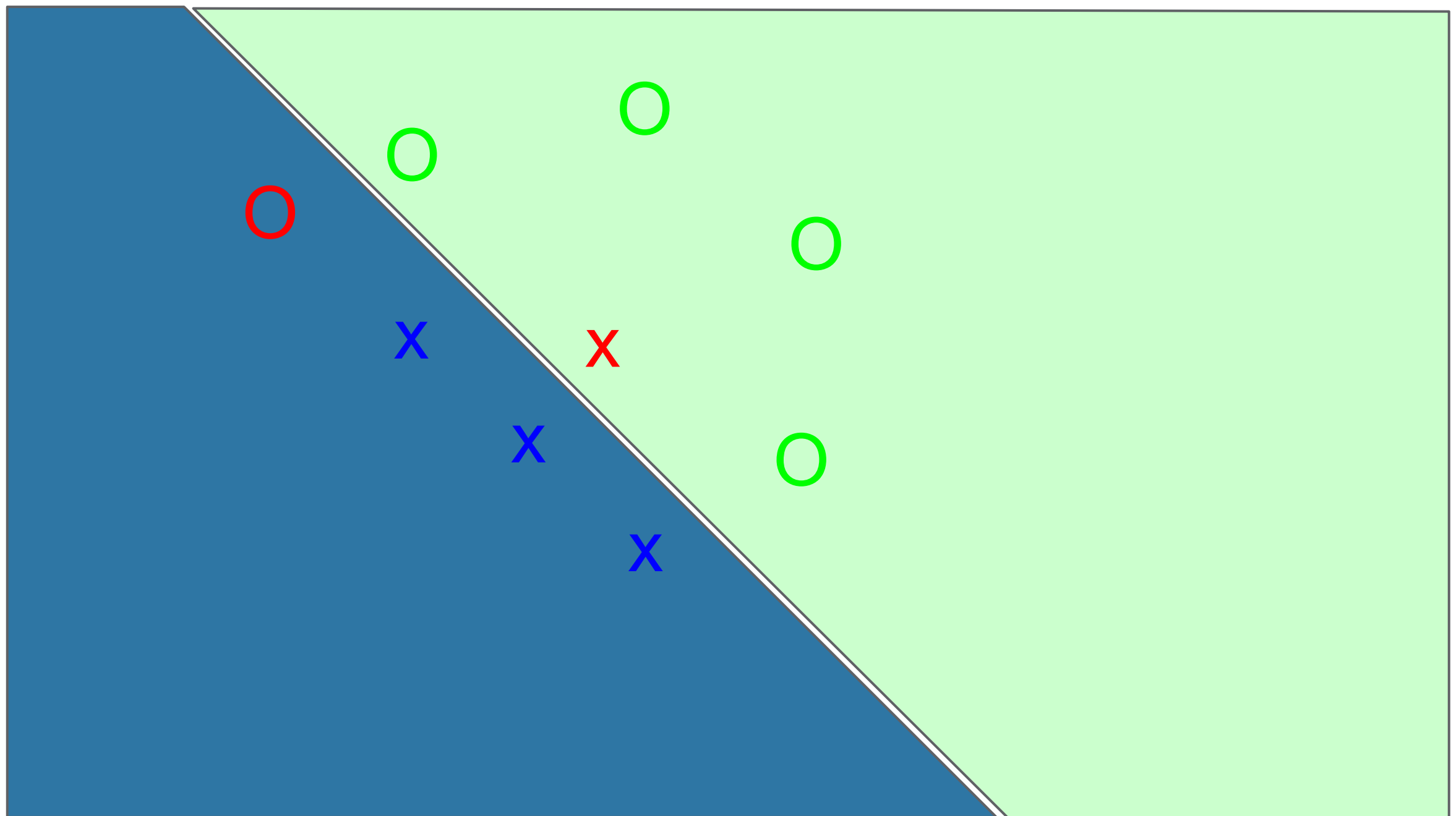
# Attacking a Linear Model



# Adversarial Examples from Overfitting

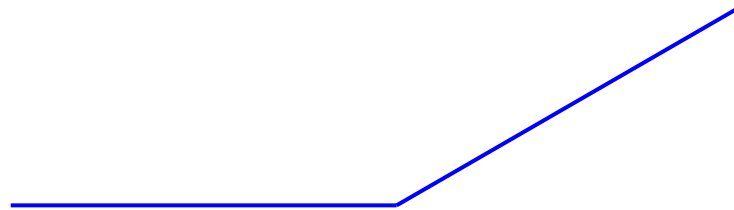


# Adversarial Examples from Excessive Linearity

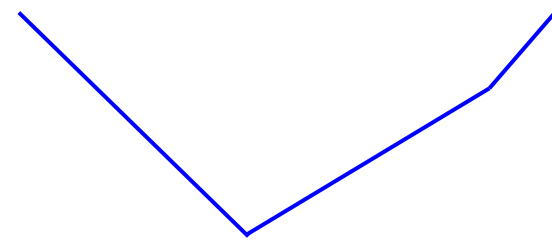


# Modern deep nets are very piecewise linear

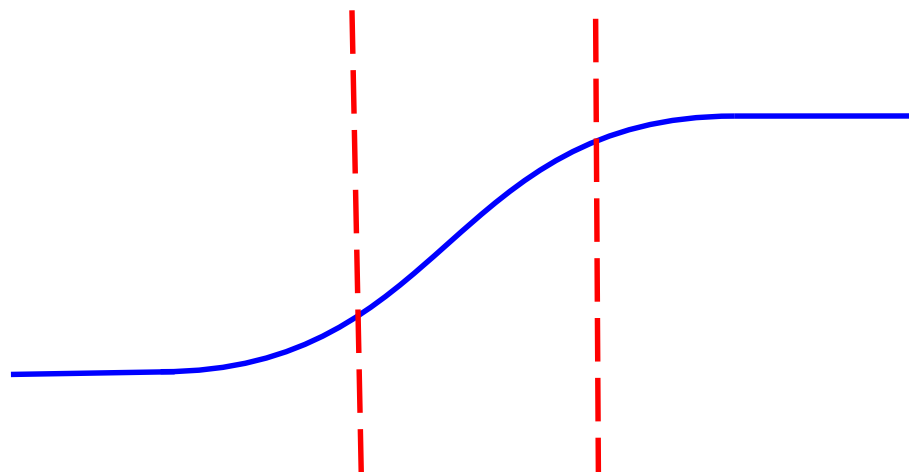
Rectified linear unit



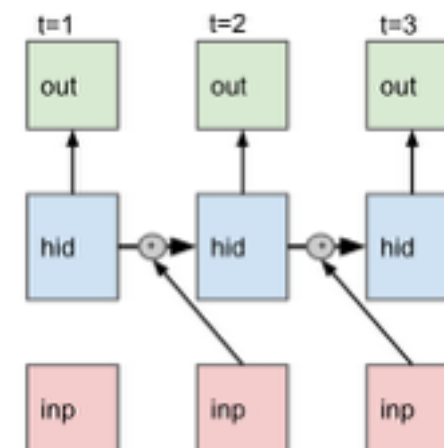
Maxout



Carefully tuned sigmoid

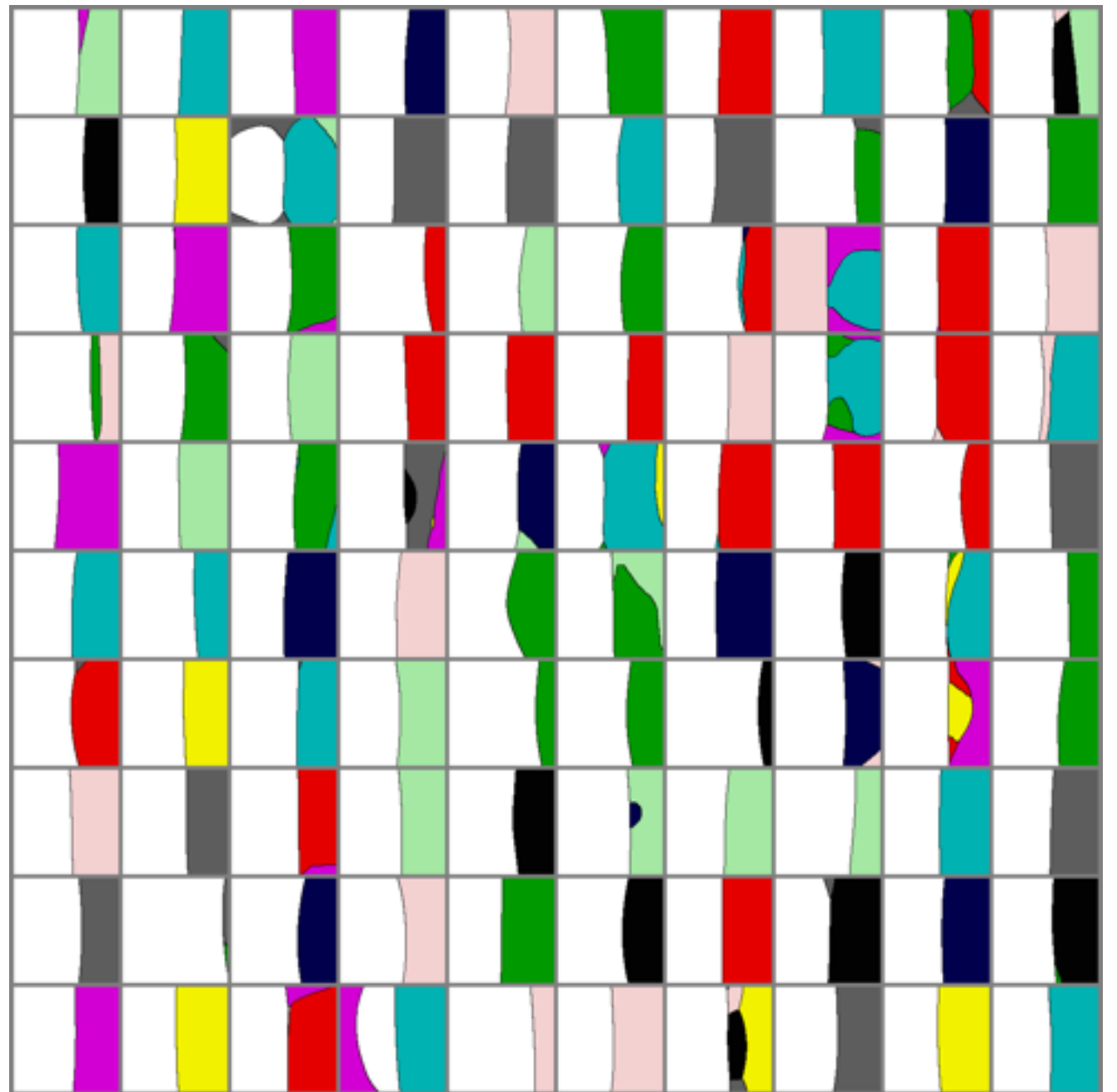
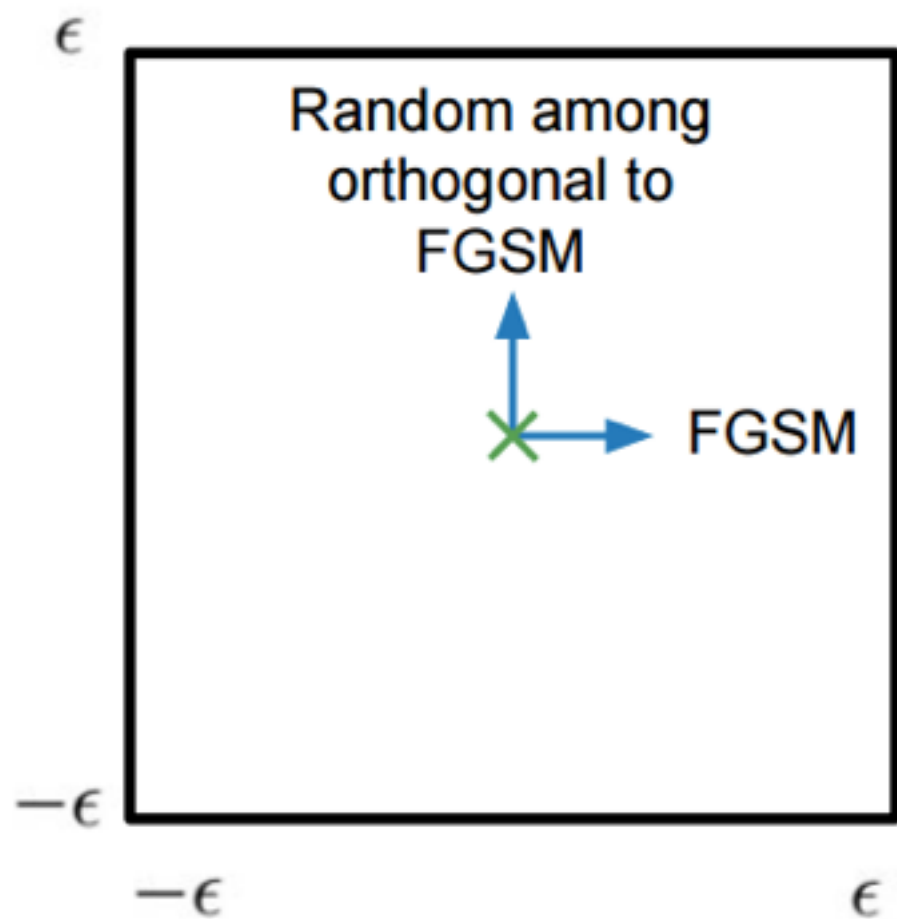


LSTM



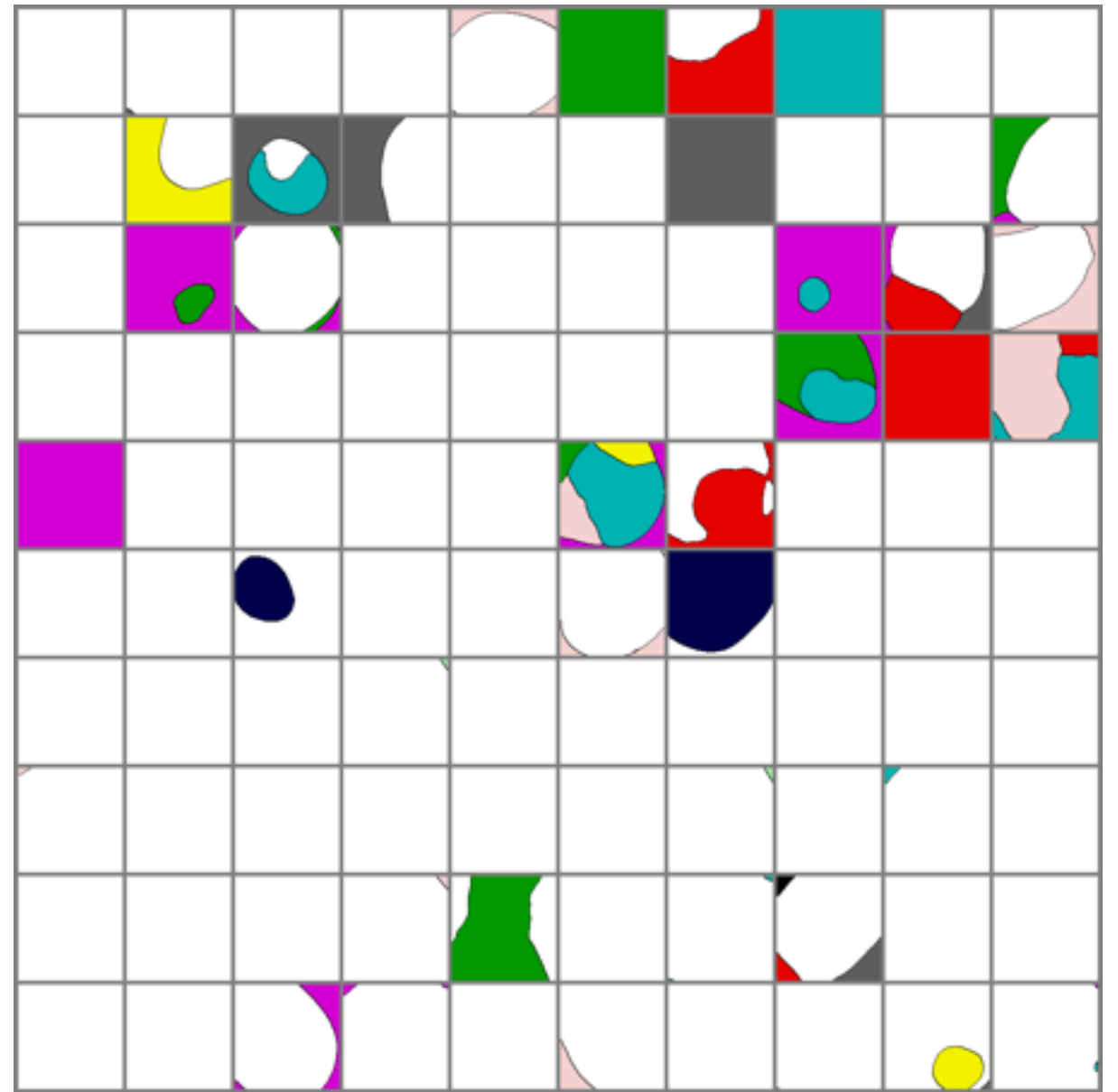
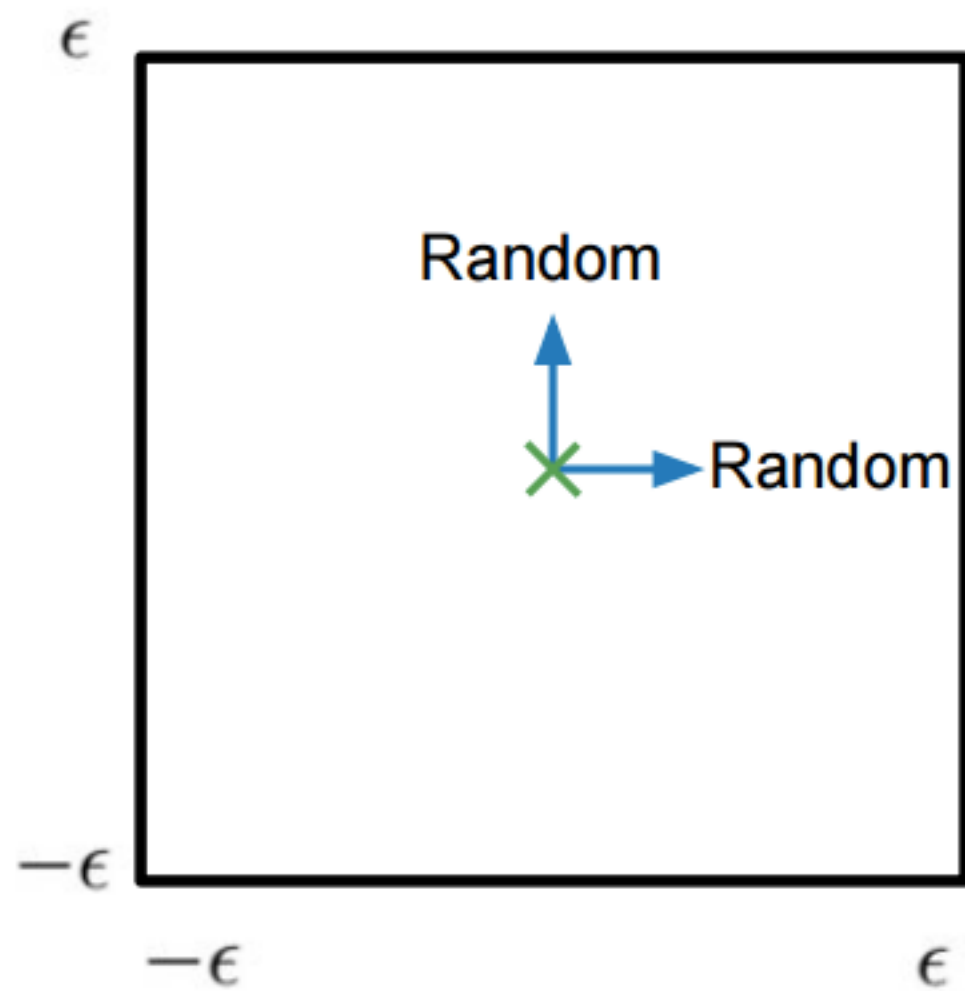


# Maps of Adversarial and Random Cross-Sections



# Maps of Random Cross-Sections

Adversarial examples  
are not noise

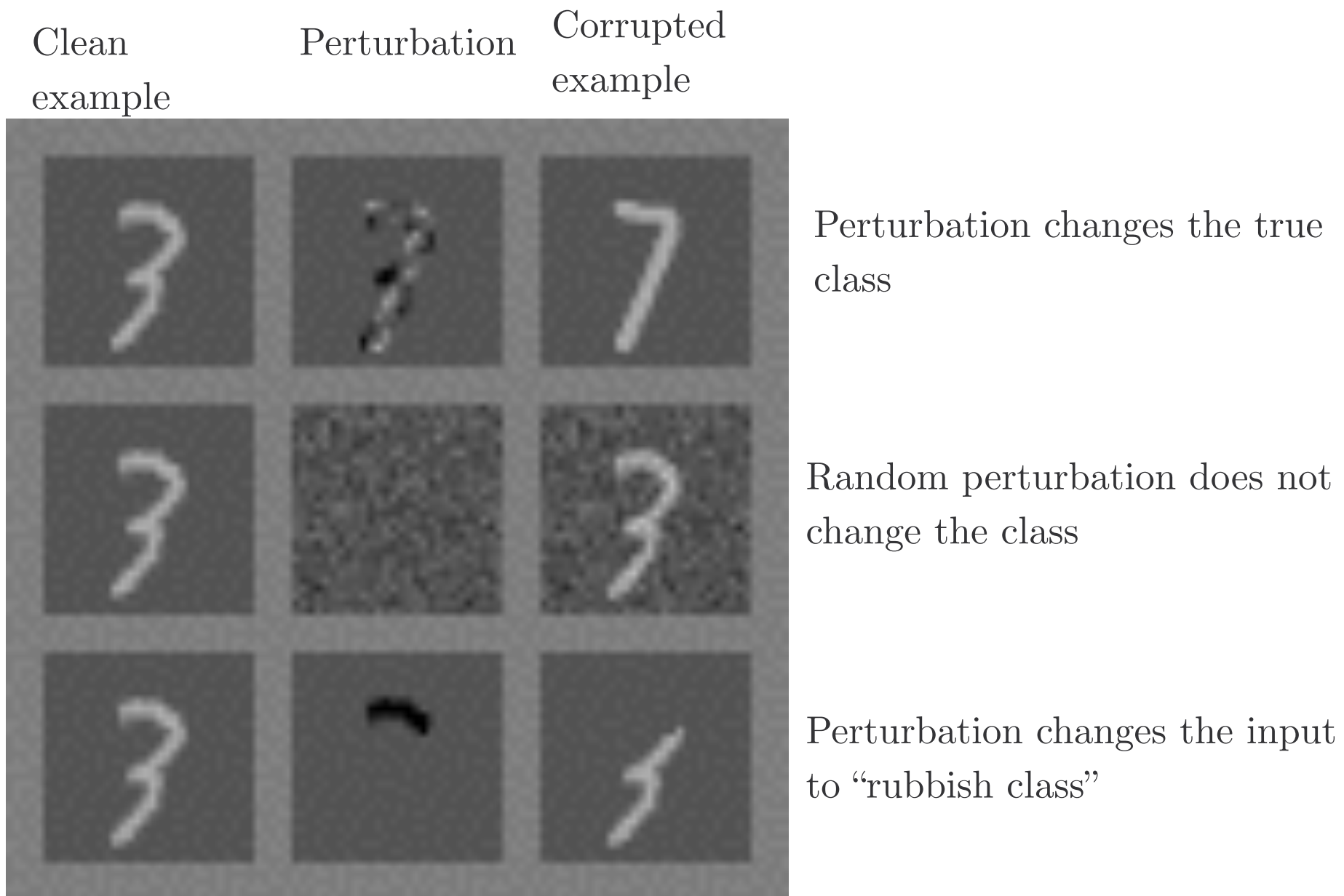


# Clever Hans



(“Clever Hans,  
Clever  
Algorithms,”  
Bob Sturm)

# Small inter-class distances



All three perturbations have L2 norm 3.96

This is actually small. We typically use 7!

# The Fast Gradient Sign Method

$$J(\tilde{\mathbf{x}}, \boldsymbol{\theta}) \approx J(\mathbf{x}, \boldsymbol{\theta}) + (\tilde{\mathbf{x}} - \mathbf{x})^\top \nabla_{\mathbf{x}} J(\mathbf{x}).$$

Maximize

$$J(\mathbf{x}, \boldsymbol{\theta}) + (\tilde{\mathbf{x}} - \mathbf{x})^\top \nabla_{\mathbf{x}} J(\mathbf{x})$$

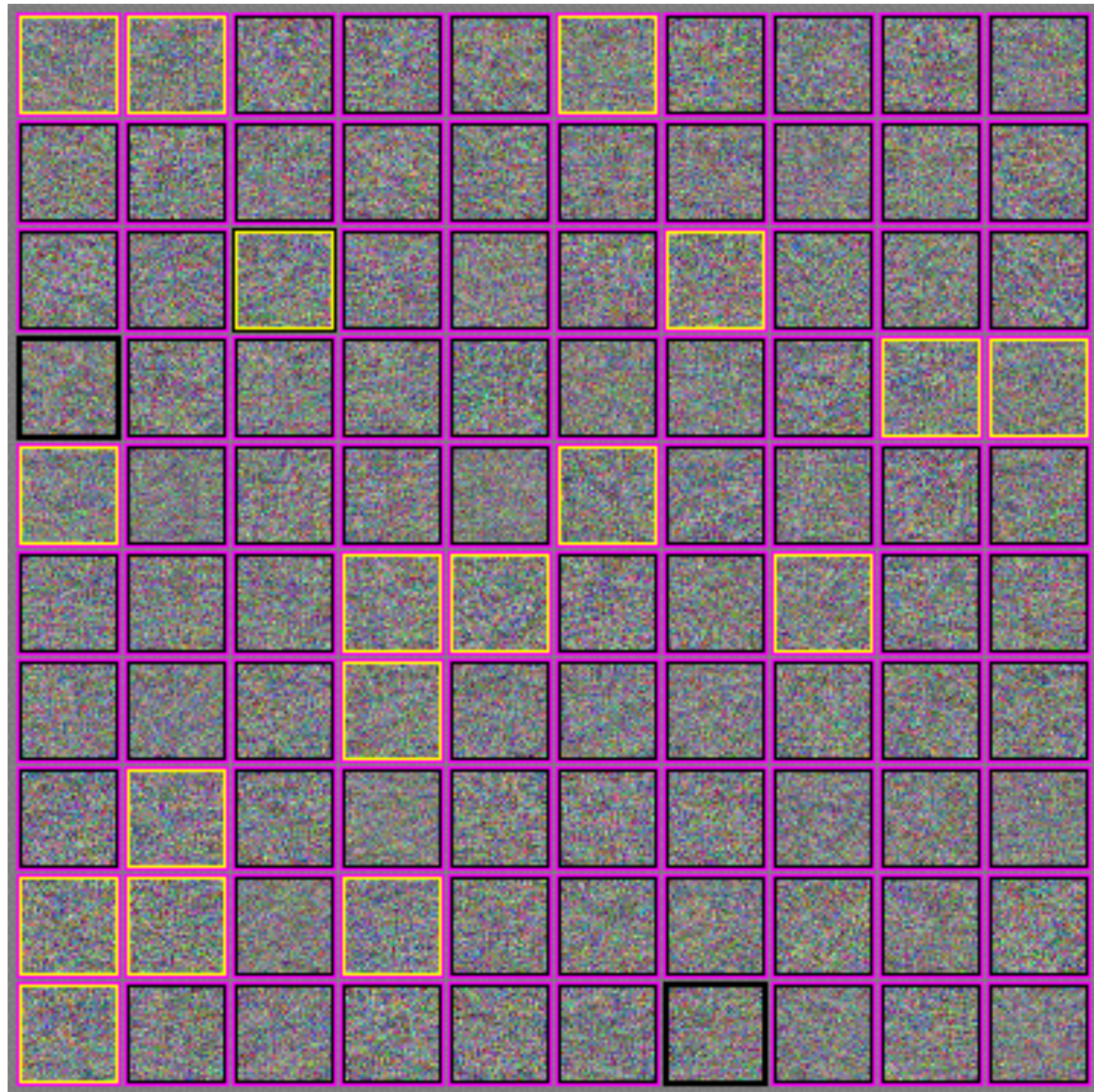
subject to

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_\infty \leq \epsilon$$

$$\Rightarrow \tilde{\mathbf{x}} = \mathbf{x} + \epsilon \text{sign}(\nabla_{\mathbf{x}} J(\mathbf{x})).$$

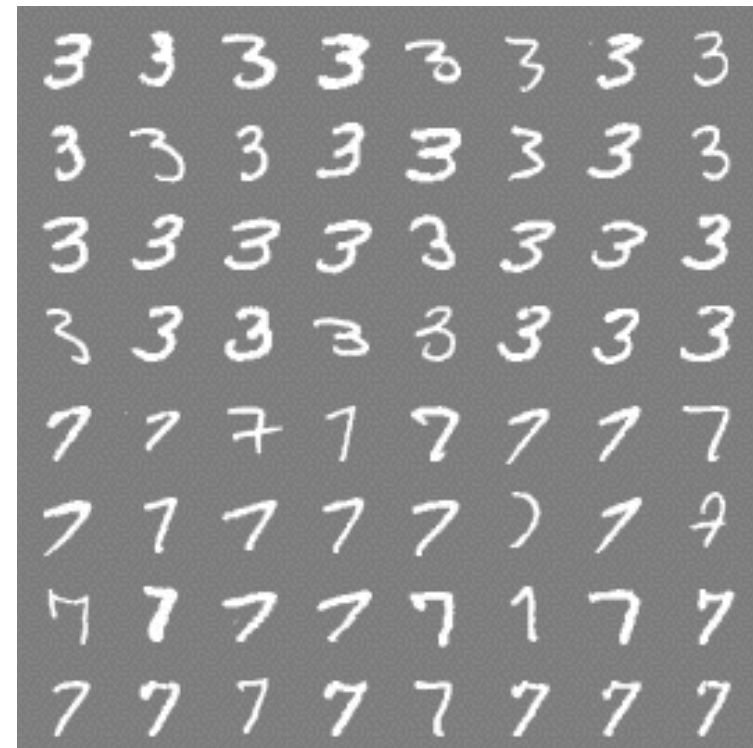
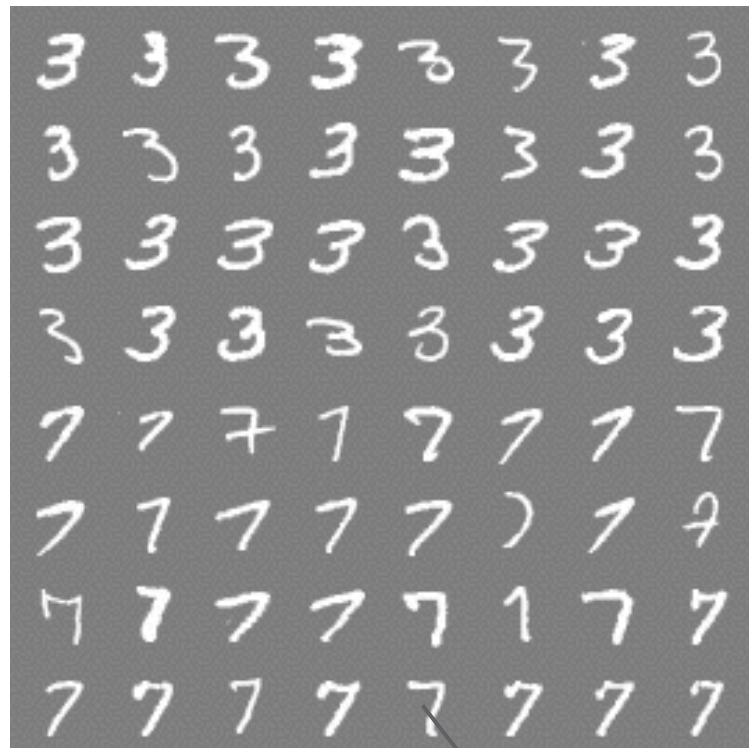


# Wrong almost everywhere

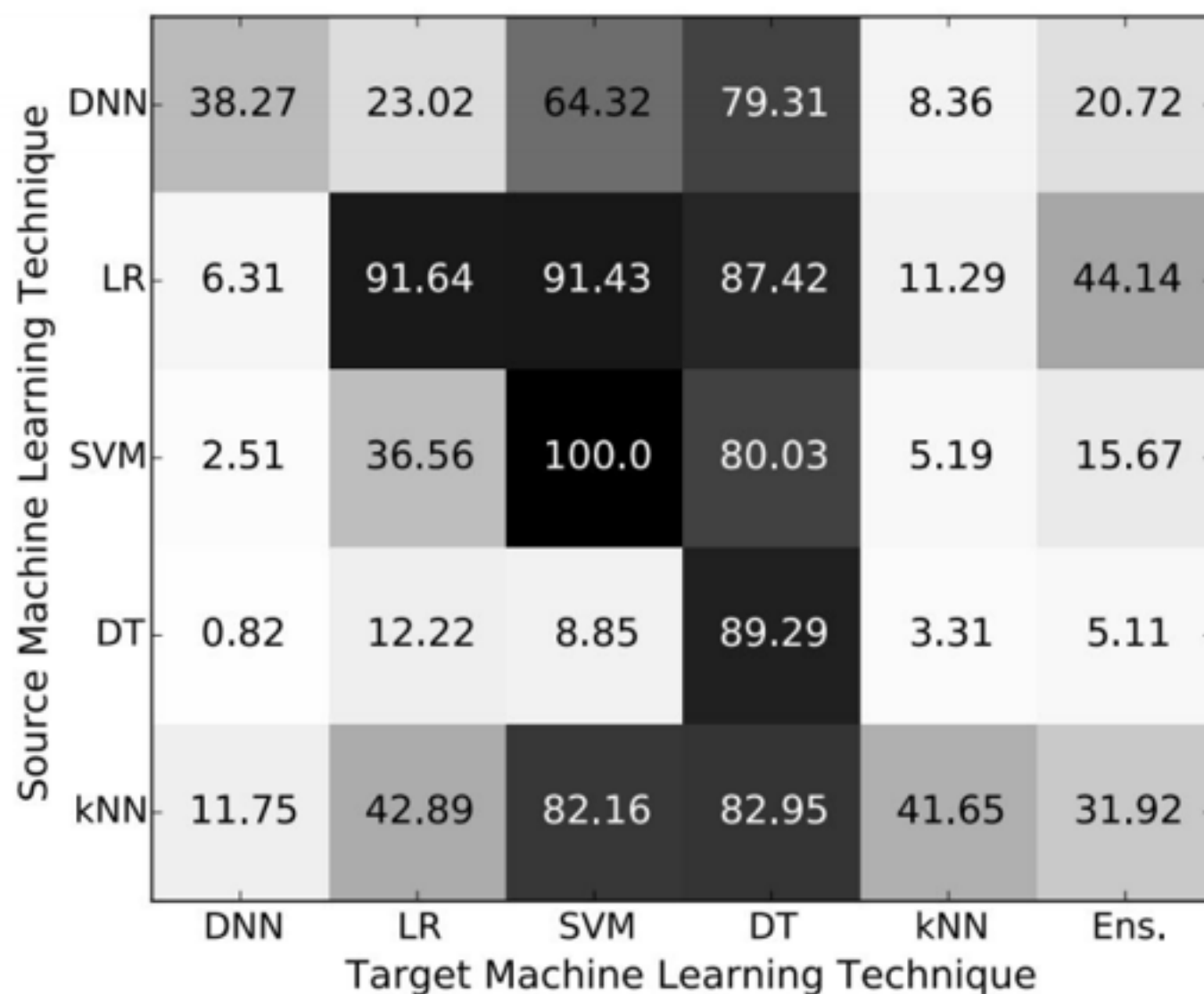




# Cross-model, cross-dataset generalization



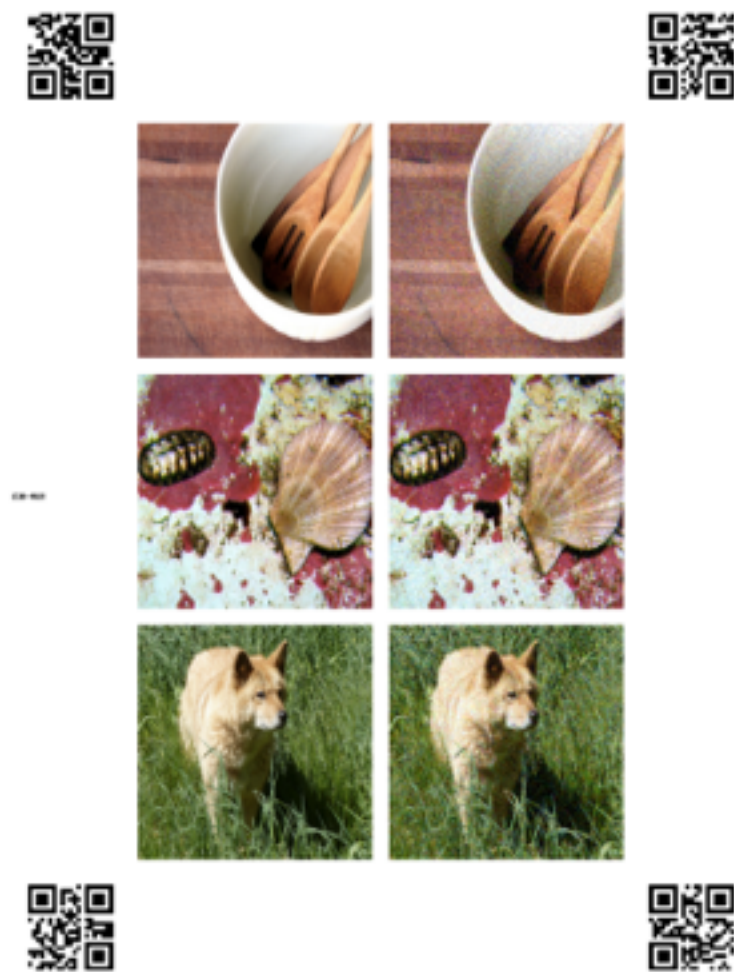
# Cross-technique transferability



- Fool cloud ML API
  - Amazon
  - Google
  - MetaMind
- Fool malware detector

(Papernot 2016)

# Adversarial Examples in the Physical World



(a) Printout

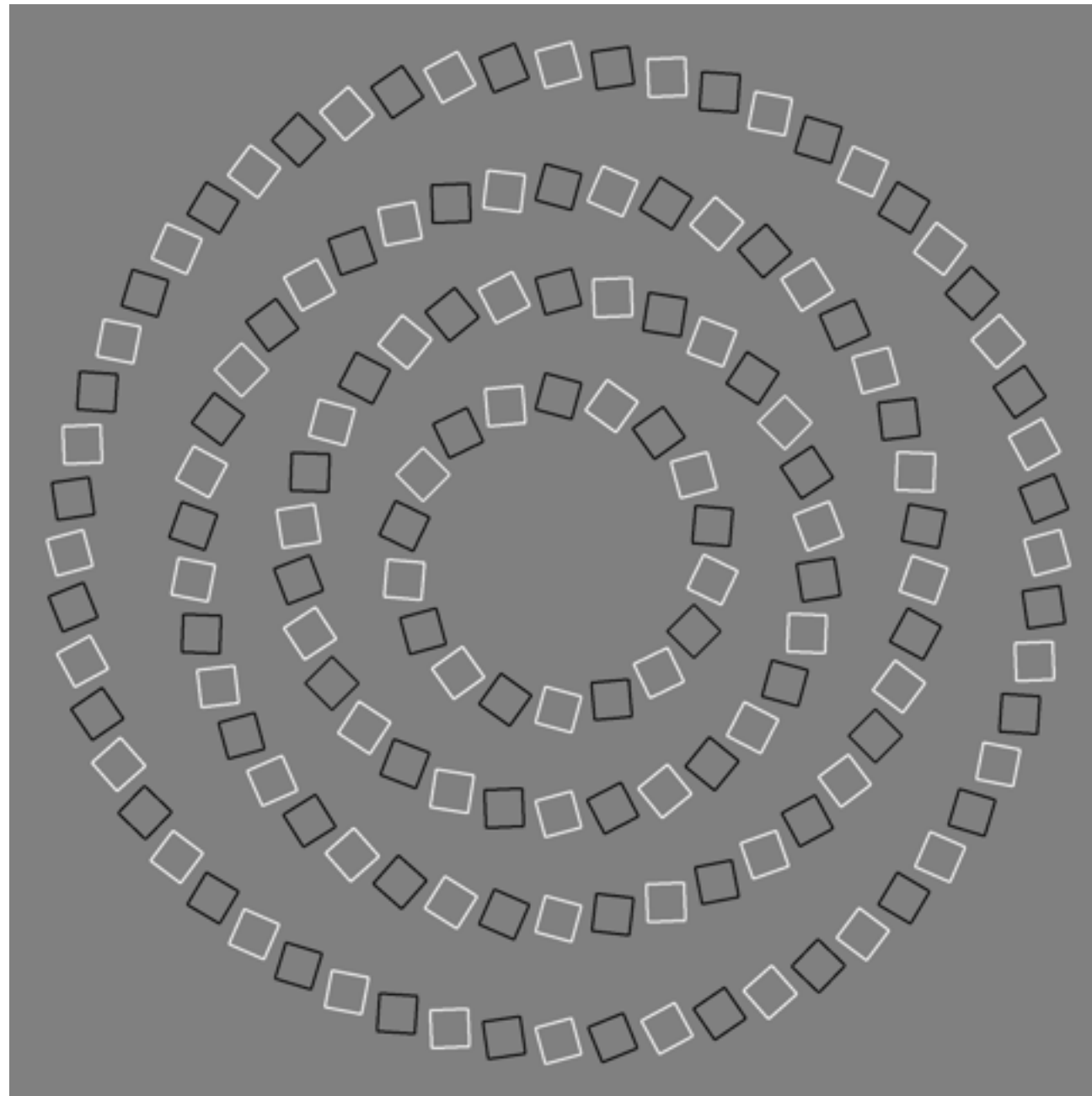


(b) Photo of printout



(c) Cropped image

# Adversarial Examples in the Human Brain



These are  
concentric  
circles,  
not  
intertwined  
spirals.

(Pinna and Gregory, 2002)

# Failed defenses

Generative  
pretraining

Removing perturbation  
with an autoencoder

Adding noise  
at test time

Ensembles

Confidence-reducing  
perturbation at test time

Error correcting  
codes

Multiple glimpses

Weight decay

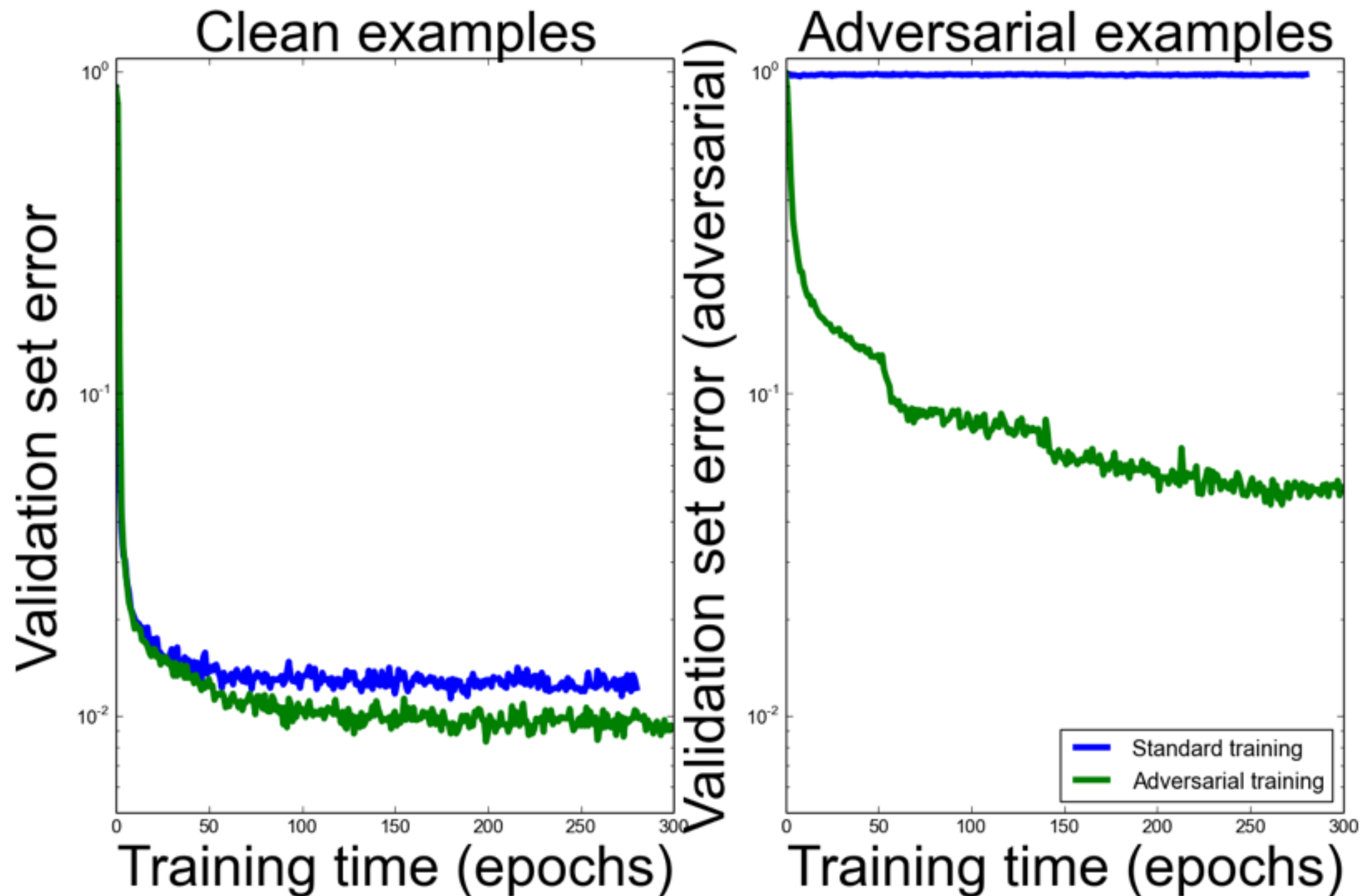
Double backprop

Adding noise  
at train time

Various  
non-linear units

Dropout

# Training on Adversarial Examples





# Virtual Adversarial Training

Unlabeled; model  
guesses it's probably  
a bird, maybe a plane



New guess should  
match old guess  
(probably bird, maybe plane)



→  
Adversarial  
perturbation  
intended to  
change the guess

# cleverhans

Open-source library available at:

<https://github.com/openai/cleverhans>

Built on top of TensorFlow (Theano support anticipated)

Benchmark your model against different adversarial examples attacks

Beta version 0.1 released, more attacks and features to be added

