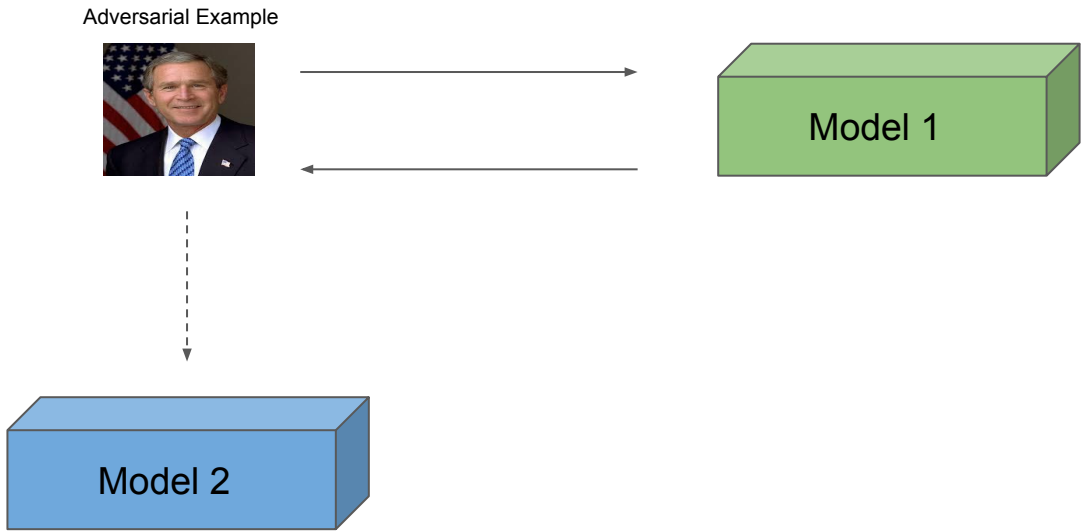
Learning Universal Adversarial Perturbations with Generative Models

Jamie Hayes & George Danezis
UCL

Adversarial examples transfer between different models.

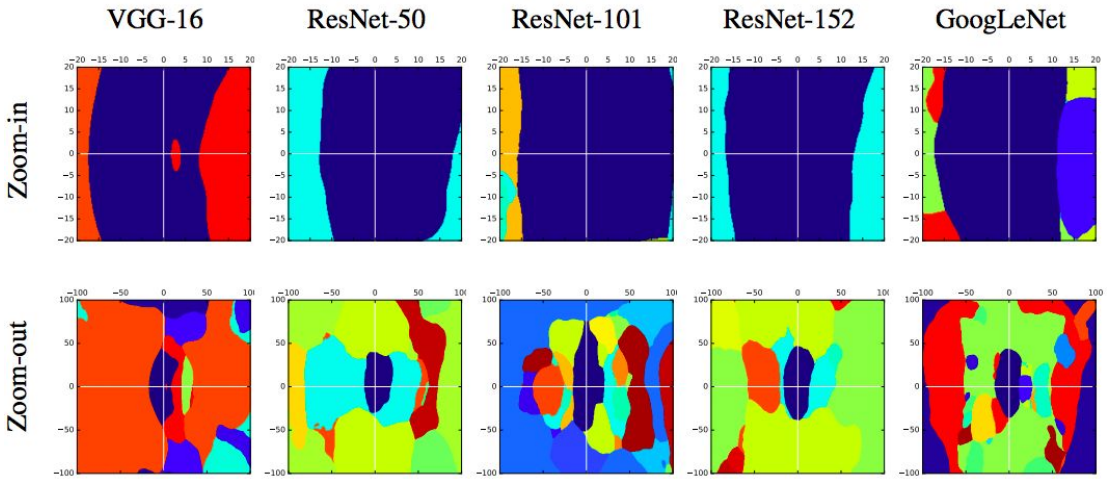
An adversarial example crafted against one model will generally fool other models.



[SZS13] Szegedy et al. Intriguing properties of neural networks.

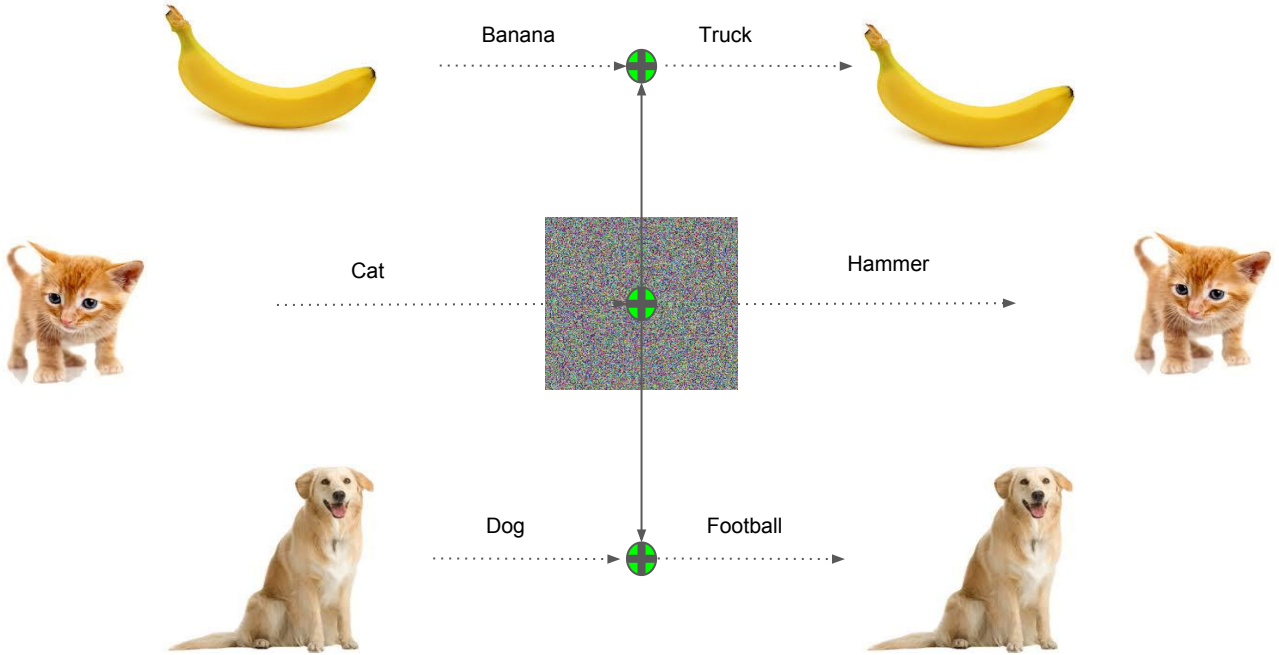
Why do adversarial examples transfer?

Why do adversarial examples transfer?



[GSS15] Goodfellow et al. Explaining and Harnessing Adversarial Examples
[LCL17] Liu et al. Delving into Transferable Adversarial Examples and Black-Box Attacks

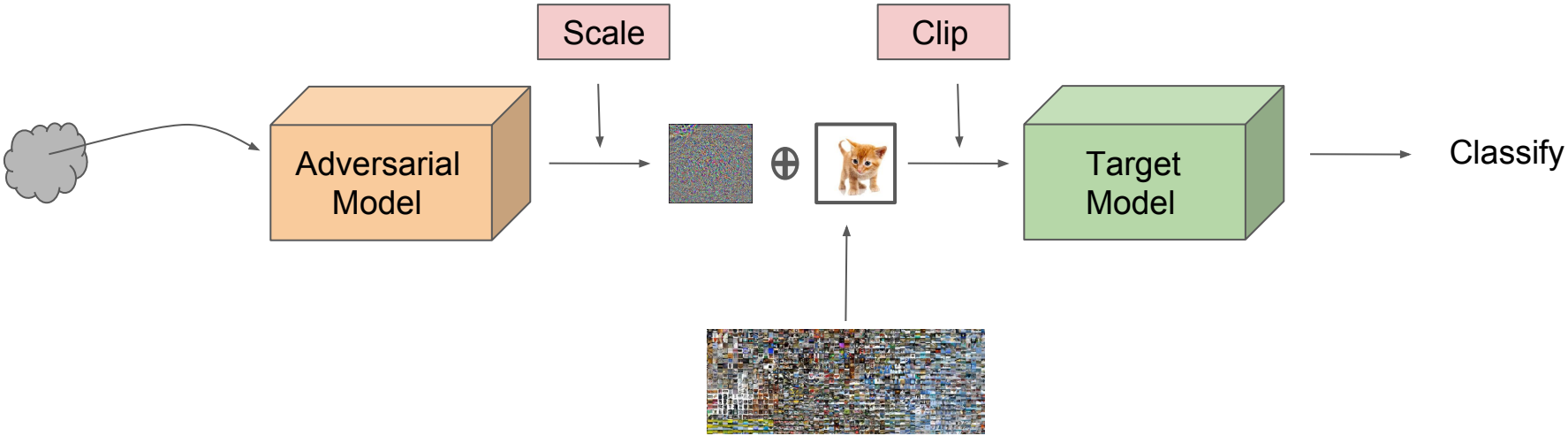
In the most extreme case, it is possible to construct a single perturbation that will fool a model when added to any image!



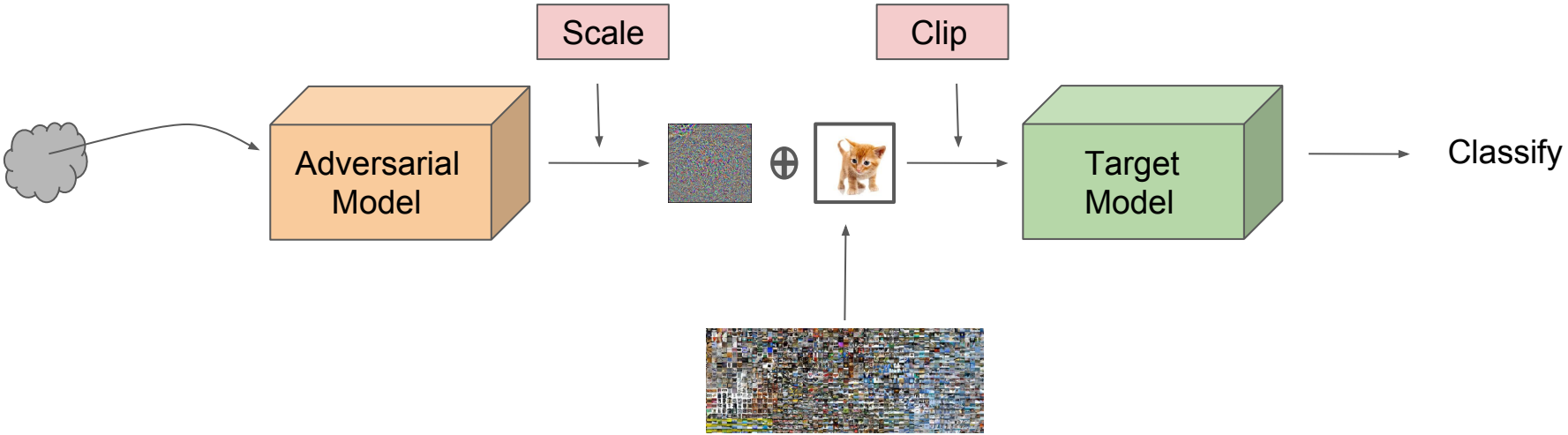
[GSS15] Goodfellow et al. Explaining and Harnessing Adversarial Examples
[MFF16] Moosavi-Dezfooli. Universal adversarial perturbations.

Can a neural network learn universal adversarial perturbations?

Can a neural network learn universal adversarial perturbations?



Can a neural network learn universal adversarial perturbations?

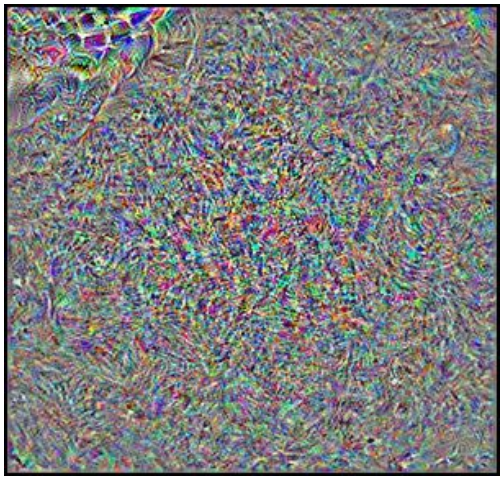


Given a model, *f*, and a image, *x*, classified correctly as c_0 , the attacker model is training to minimize:

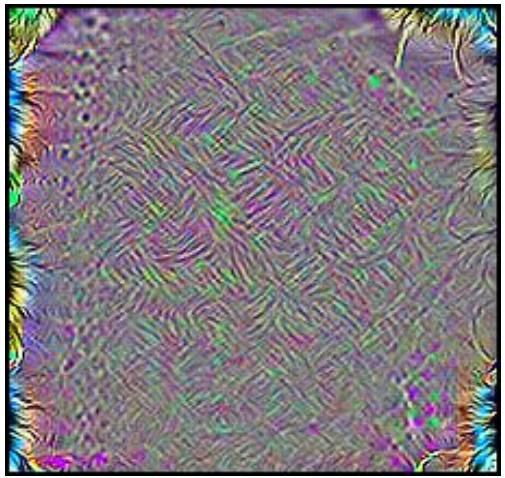
$$L_{nt} = \max\{\log[f(\delta' + x)]_{c_0} - \max_{i \neq c_0} \log[f(\delta' + x)]_i, -\kappa\} + \alpha \cdot \left\|\delta'\right\|_p$$

We scale the perturbation such that $\frac{\left\|\delta'\right\|_p}{\|x\|_p}$ never exceeds 0.04.

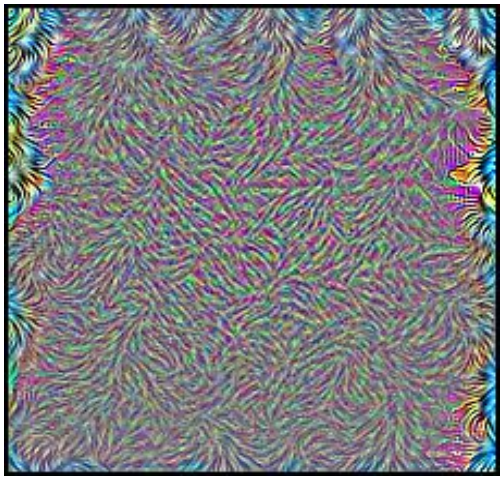
Learned Universal Adversarial Perturbations



Inception-V3



ResNet-152



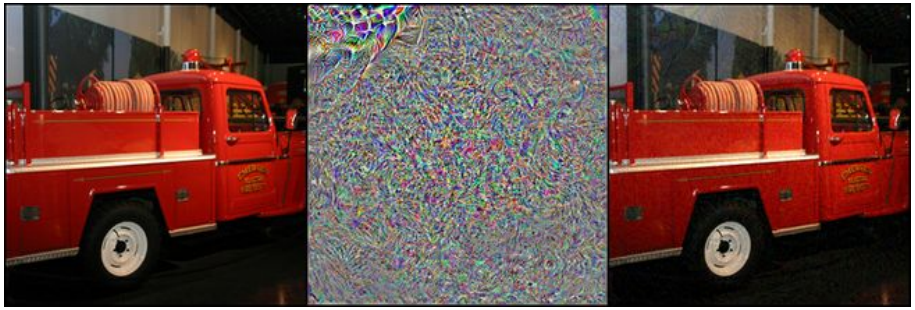
VGG-19

ImageNet test accuracy

Original:	77.2%	78.4%	71.0%
Adversarial:	22.7%	11.1%	15.1%

Inception-V3:

Fire engine (54.6%)

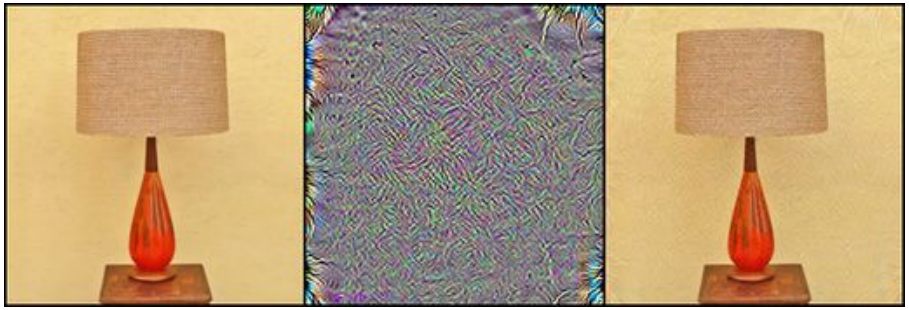


Inception-V3:

Wrecker (79.4%)

ResNet-152:

Table lamp (87.2%)

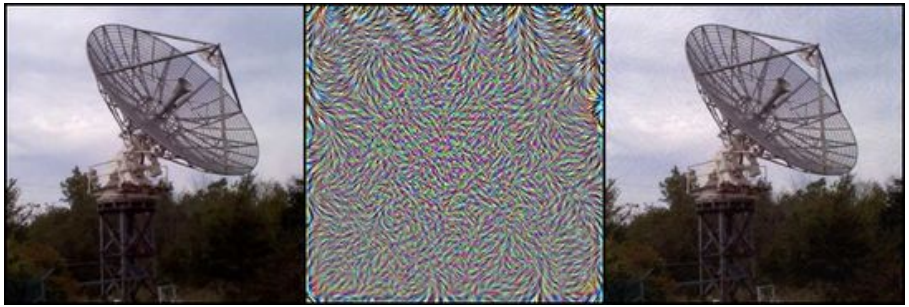


ResNet-152:

Tabby cat (41.9%)

VGG-19:

Radio telescope (97.5%)



VGG-19:

Great Pyrenees (36.7%)

We can perform targeted attacks to force the model to always classify as label, *c*, by changing the loss term from:

$$L_{nt} = \max\{\log[f(\delta' + x)]_{c_0} - \max_{i \neq c_0} \log[f(\delta' + x)]_i, -\kappa\} + \alpha \cdot \left\|\delta'\right\|_p$$

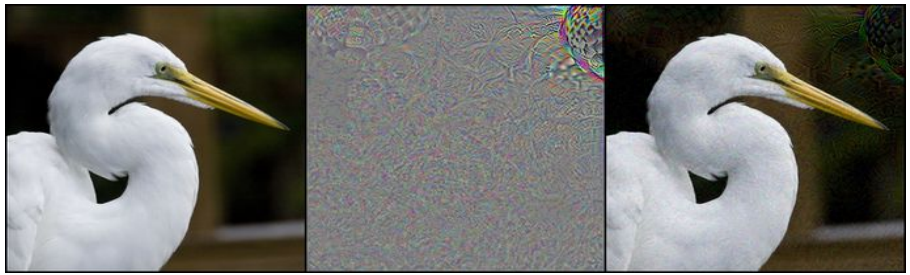
To:

$$L_t = \max\{\max_{i \neq c} \log[f(\delta' + x)]_i - \log[f(\delta' + x)]_c, -\kappa\} + \alpha \cdot \left\|\delta'\right\|_p$$

Target class: Golf Ball

Inception-V3:

American egret (95.0%)

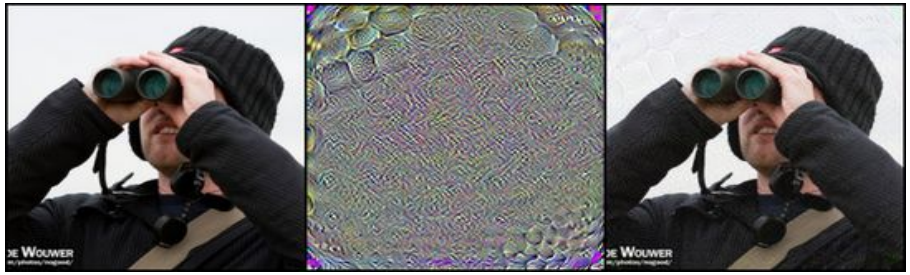


Inception-V3:

Golf ball (98.8%)

ResNet-152:

Binoculars (99.9%)

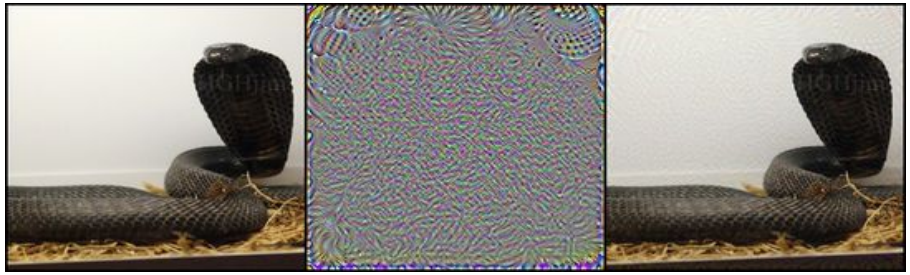


ResNet-152:

Golf ball (62.9%)

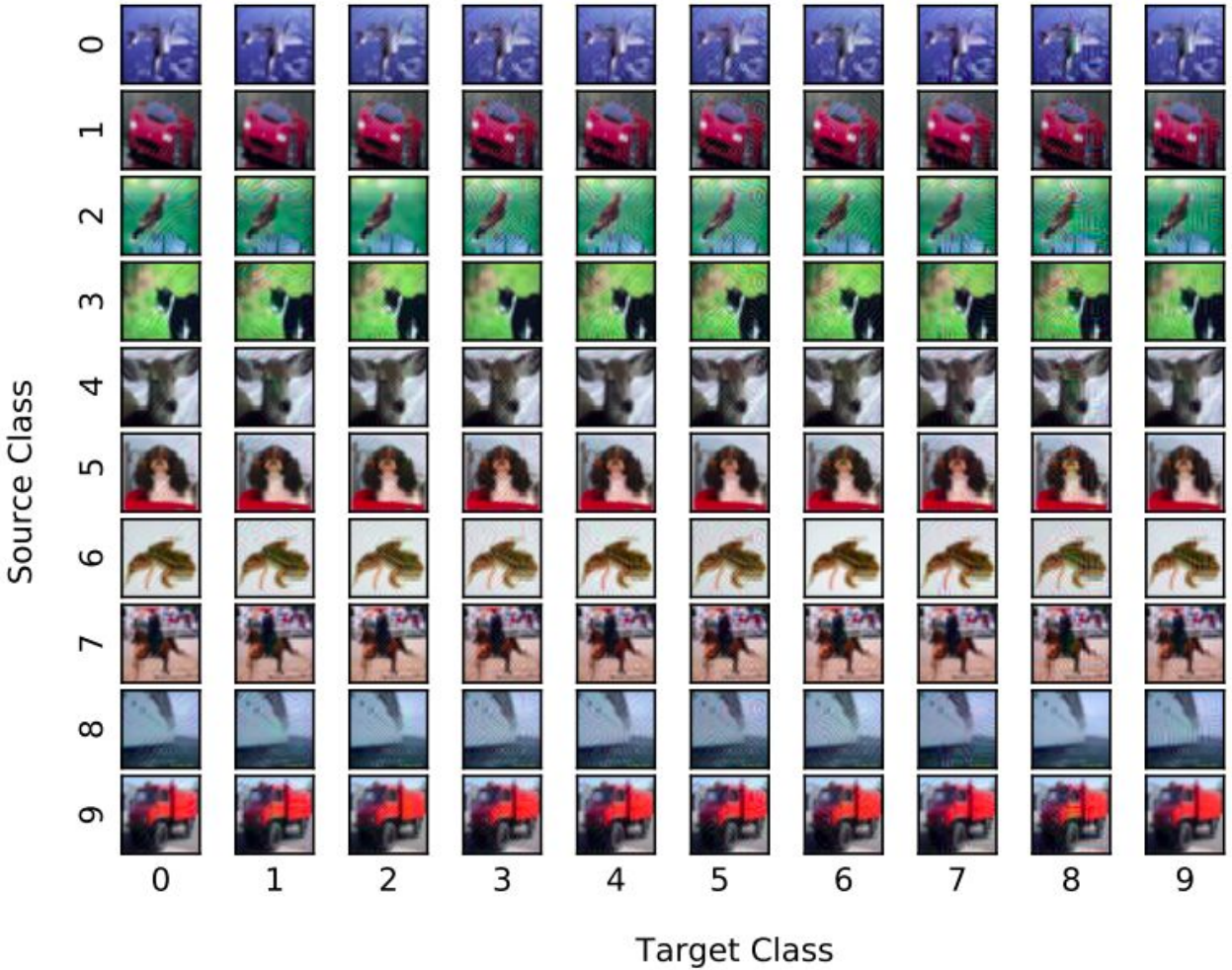
VGG-19:

Indian cobra (99.9%)



VGG-19:

Golf ball (99.7%)


Source Class
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9
Target Class

Adversarial Training Defense

Include adversarial examples during training to improve robustness.

Instead of optimizing $L(\theta, x, y)$, optimize $\alpha \cdot L(\theta, x, y) + (1 - \alpha) \cdot L(\theta, x + \delta', y)$

Adversarial Training Defense

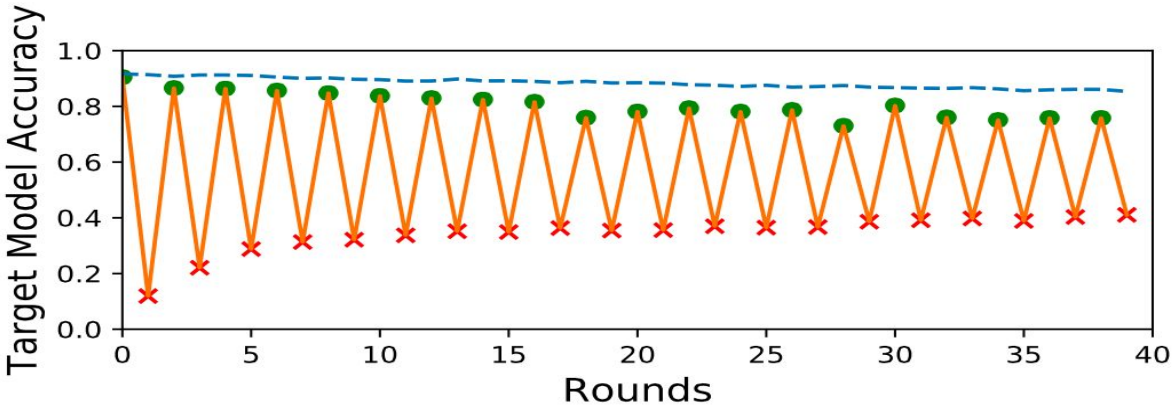
Play Cat and Mouse game:

1) Train generative model to create perturbations, report target model accuracy on adversarial examples
2) Use adversarial training to defend target model, report target model accuracy on adversarial examples.
3) Go to (1)

Adversarial Training Defense

Play Cat and Mouse game:

1) Train generative model to create perturbations, report target model accuracy on adversarial examples
2) Use adversarial training to defend target model, report target model accuracy on adversarial examples.
3) Go to (1)



Related Work

Three pre-prints using the same technique appeared online within a few days of one another.

This work, Poursaeed et al. [1], Mopuri et al. [2].

	VGG-19	INCEPTION-V1
This work	0.846	0.809
Poursaeed et al. [1]	0.801	0.792
Mopuri et al. [2]	0.838	0.904

[1] Poursaeed et al. Generative Adversarial Perturbations.
[2] Moosavi-Dezfooli. NAG: Network for Adversary Generation.

Thanks!

j.hayes@cs.ucl.ac.uk

@_jamiedh