Adversarial Machine Learning and Attack Against Deep Neural Network

Bowei Xi
Department of Statistics
Purdue University
xbw@purdue.edu

Malicious Attacks Against Deep Neural Network

Deep neural network (DNN) successful at complex tasks, such as image classification, object recognition etc.

DNN used in autonomous systems, but forward thinking AI is not secured against potential attacks.

Existing work focused on adding minor perturbation to an input based on an optimization approach.

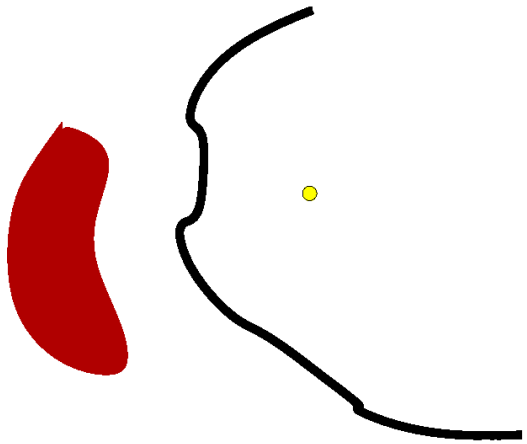
We study DNN's classification boundary through its response surface and uncertainty regions to see what cause the adversarial examples.

We study both convolutional NN and fully connected NN.

We develop a game theory inspired strategy to improve DNN robustness facing adversaries.

Many Unanswered Questions about DNN

What is the shape of DNN classification boundary?
How many adversarial examples are there given one clean image? – Currently one to a few dozen.
Are adversarial examples transferable? – Belief of transferability.
What caused these adversarial examples? – Linear vs Non-linear



Stronger Adversarial Attack

First work to study DNN response surface and propose the concept of DNN uncertainty regions.

Identify the regions for infinitely many adversarial examples in a small neighborhood surrounding a clean image – a stronger attack.

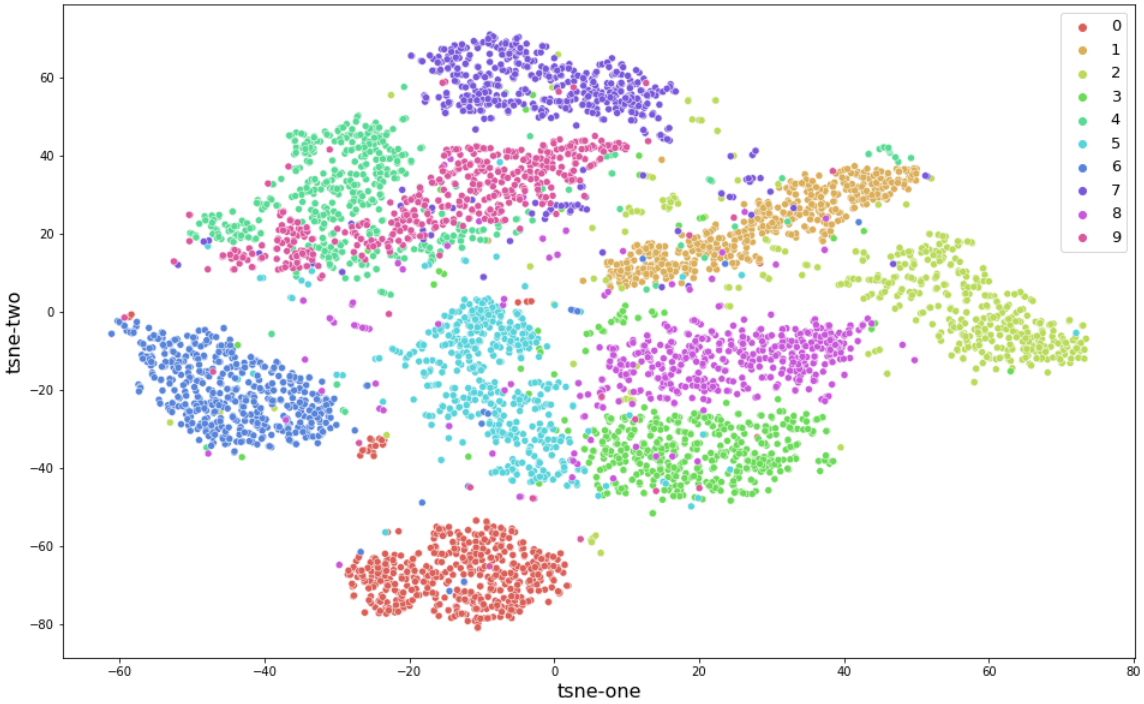
Transferability of adversarial examples is not universal.

Existing attacks, e.g., C&W L2, cause errors from only one targeted DNN model. We use less pixels, have smaller perturbations added to a clean image, and attack multiple DNNs simultaneously.

Shu, J., Xi, B., Kamhoua, C. A., Understanding Adversarial Examples Through Deep Neural Network's Response Surface and Uncertainty Regions, arXiv

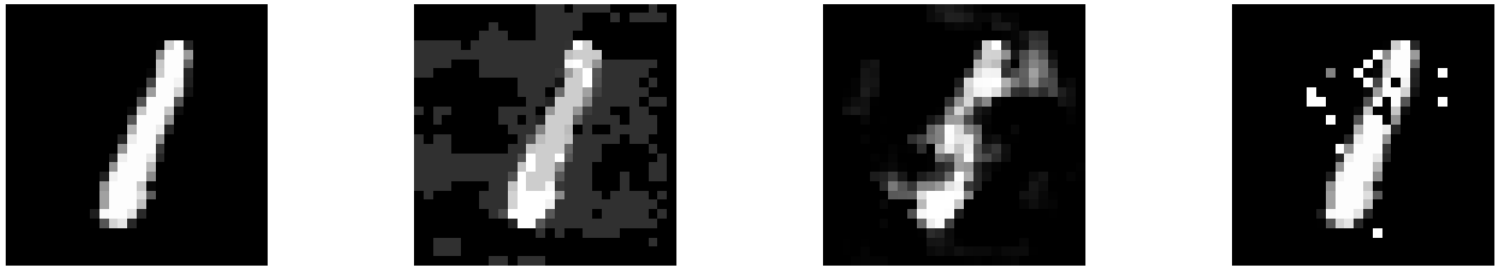
Lower Dimensional Projection of MNIST

Use t-SNE, a nonlinear dimension reduction technique.



Adversarial Examples

Left to right: (1) Clean image 1; (2)Fast Gradient Sign Method (FGSM) Attack 1 $\rightarrow$ 2; (3) Carlini & Wagner L_2 (CW2) Attack 1 $\rightarrow$ 5; (4) Pointwise (PW) Attack 1 $\rightarrow$ 8.



DNN Response Surface and Uncertainty Regions

Response surface methodology well studied in statistics, ignored by machine learning community.

Image W, a matrix for grayscale, a tensor for color. Elements integer valued, 0, 1, ..., 255. Rescaled to [0,1].

t is a object class (e.g., a digit for MNIST).

DNN response surface is described by $M_i(W) = t$.

M_1 to M_k are obtained by having one DNN model trained on the same training data, and only varying the initial values.

DNN uncertainty region

$U := \Big\{W : \exists i, j \in (1, ..., k),\ s.t.\ M_i(W) \neq M_j(W)\Big\}.$

MNIST Experiment

We use LeNet (LeCunn et. al. 1998) for MNIST. $W \in [0,1]^{784}$.

Train LeNet with different starting values, obtained 10 models.

LeCunn et. al. 1998 reported the highest test accuracy as 99.05%
Others between 96% to around 99%

Below is misclassification rate for clean test images.

M_1	M_2	M_3	M_4	M_5
0.033	0.035	0.025	0.019	0.017
M_6	M_7	M_8	M_9	M_{10}
0.015	0.013	0.012	0.012	0.012

Adversarial Attacks

Foolbox has 42 attack algorithms. Most either generate a handful of adversarial examples or add large perturbations. Six used in the experiments.

Pointwise (PW) Attack, $\min||W^a - W||_0$, the number of perturbed pixels.

Carlini & Wagner L_2 (CW2), $\min\{\text{Distance}(W, W^a) + c \times \text{loss}(W^a)\}$.

Fast Gradient Sign Method (FGSM), $W^a = W - \epsilon \times \text{sign}(\nabla \text{loss}(W))$.

NewtonFool (NF) Attack, Basic Iterative Method (BIM), Moment Iterative (MI) Attack also search for the gradient.

DNN Response Surface and Uncertainty Regions

We compute L_2 distance $d(W^c, W^a) = ||W^c - W^a||_2$.

We study DNN response surface and uncertainty regions in $B(\delta) := \{W : d(W^c, W) \leq \delta\}$, with a small δ.

Attack M_1. Need each attack algorithms to generate at least 80-100 adversarial images for a digit $t \neq 1$.

If $\exists W_i^a \neq W_i^c$, compute $s_i = max_{attack\ k}(W_{k,i}^a) - min_{attack\ k}(W_{k,i}^a)$. Rank perturbed pixels by $s_{(1)} \geq s_{(2)} \geq \cdots s_{(m)}$.

Construct a hyper-rectangle based on intervals not deemed as a constant.

$R_k(t) = [min\ (W_{k,(1)}^a),\ max\ (W_{k,(1)}^a)] \otimes \cdots \otimes [min\ (W_{k,(h)}^a),\ max\ (W_{k,(h)}^a)]$.

DNN Response Surface and Uncertainty Regions

Additional dimensions – add a constant amount to a pixel. Do not change the shape and size of $R_k(t)$. Only move $R_k(t)$ to a different location, increasing total perturbations. Table shows misclassification rate.

	$s_{(i)}$	M_1	M_3	M_6
FGSM 375d $1 \rightarrow 2$	0.012	**0.915**	0	0
CW2 175d $1 \rightarrow 5$	0.011	**0.047**	0.135	**1**
PW 35d $1 \rightarrow 8$	1	**0.809**	**0.48**	**0.974**

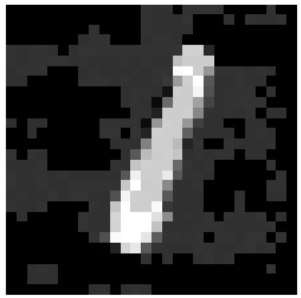
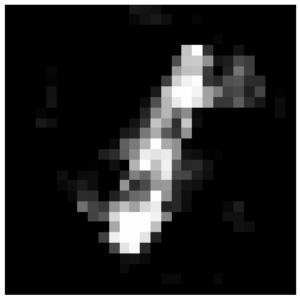
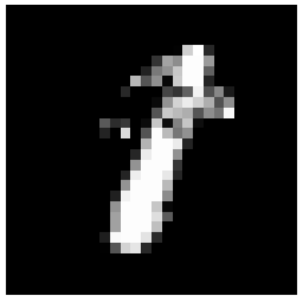
DNN Response Surface and Uncertainty Regions

Carlini & Wagner L_2 attack and FGSM eventually move $R_k(t)$ to where M_1 misclassification rate nearly 100%, other M_js 0. Table shows L_2 distance.

	L_2^{min}	L_2^{max}	$\bar{L}_2$	$L_{2,a}^{min}$	$L_{2,a}^{max}$	$\bar{L}_{2,a}$
FGSM $1 \rightarrow 2$	14.259	14.41	14.335	14.413	15.213	14.784
CW2 $1 \rightarrow 5$	10.743	12.989	11.567	9.932	16.703	12.298
PW $1 \rightarrow 8$	5.205	14.84	10.023	12.526	26.526	17.329

Sampled from Hyper-Rectangles

Left to right: (1)FGSM subspace $1 \rightarrow 2$; (2) CW2 subspace $1 \rightarrow 5$; (3) PW subspace $1 \rightarrow 8$.

CIFAR10

CIFAR10 has 60,000 32x32 color images in 10 classes, 50,000 as training and 10,000 as test.

A CIFAR10 image is in $[0, 1]^{3072}$.

We re-train the MobileNet, which has an initial convolution layers followed by 19 residual bottleneck layers, a complex structure to reduce memory usage.

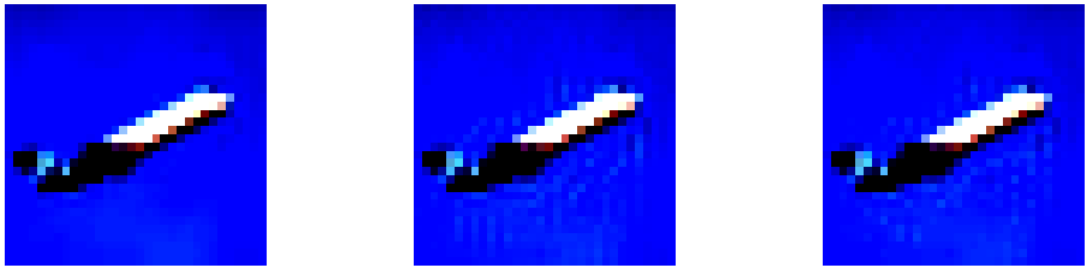
The misclassification rates of five re-trained MobileNet models on the clean test images.

M_1	M_2	M_3	M_4	M_5
0.0767	0.0728	0.0734	0.0727	0.0744

CIFAR10

MobileNet Attack Misclassification Rates

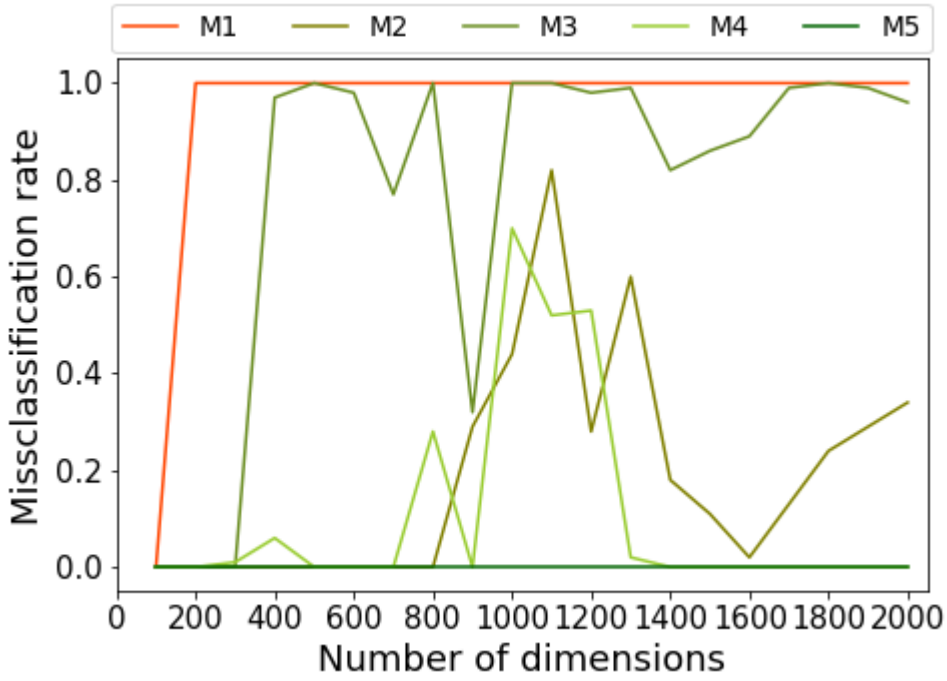
	M_1	M_2	M_3	M_4	M_5
BIM L_2 3017d airplane$\rightarrow$deer	1	0	0.88	0	0



Left: clean airplane; Mid: airplane labeled as deer by BIM L_2 attack; Right: airplane labeled as deer by sampling.

CIFAR10

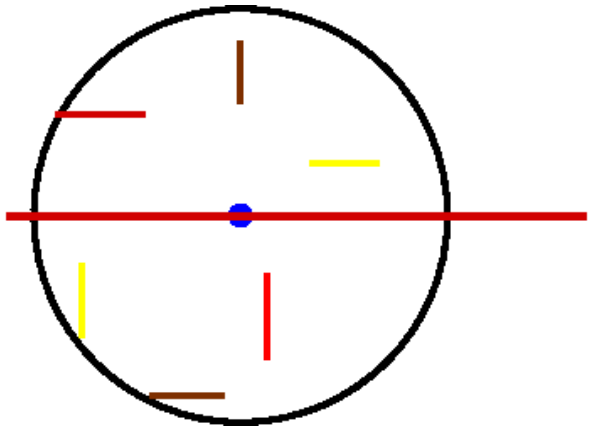
The misclassification rates as increasing the dimensions of the hyper-rectangle. The largest interval is 0.2 and the 2000th interval is 0.017.



$M1$ under attack. See also $M5$, $M2$ and $M4$. The direction of adversarial perturbation is important. Adversarial examples cannot be generated by randomly sampling in 3072d $B(\delta, W)$.

DNN Classification Boundary

Conceptual plot in a $\delta-$neighborhood of a clean image. For digit 1 (784-dim), let $\delta = 6$. Three types of "cracks".



Robust DNN

Randomization strategy inspired by mixed strategy in game theory.

1. Randomly select a DNN from a pool of DNNs
2. Ensemble of a random set of DNNs
3. Add small random noise to the learned weights of a DNN

Utilizing the non-convex optimization process in DNN training.

Zhou, Y., Kantarcioglu, M., Xi, B., Exploring the Effect of Randomness on Transferability of Adversarial Samples against Deep Neural Networks, revision submitted

Robust DNN

Assume $d(W^c, W^a) \leq \epsilon$. Attacking M_j in a pool of M_1, ..., M_n. Attacker has perfect knowledge about M_j, but doesn't know the rest of DNNs.

Baseline: accuracy on clean test images W^c.

Static: accuracy on W^a using the model under attack M_j.

Random-Model-n: randomly pick 1 from (M_1, ..., M_n) to label W^a.

Ensemble-n: majority vote of (M_1, ..., M_n).

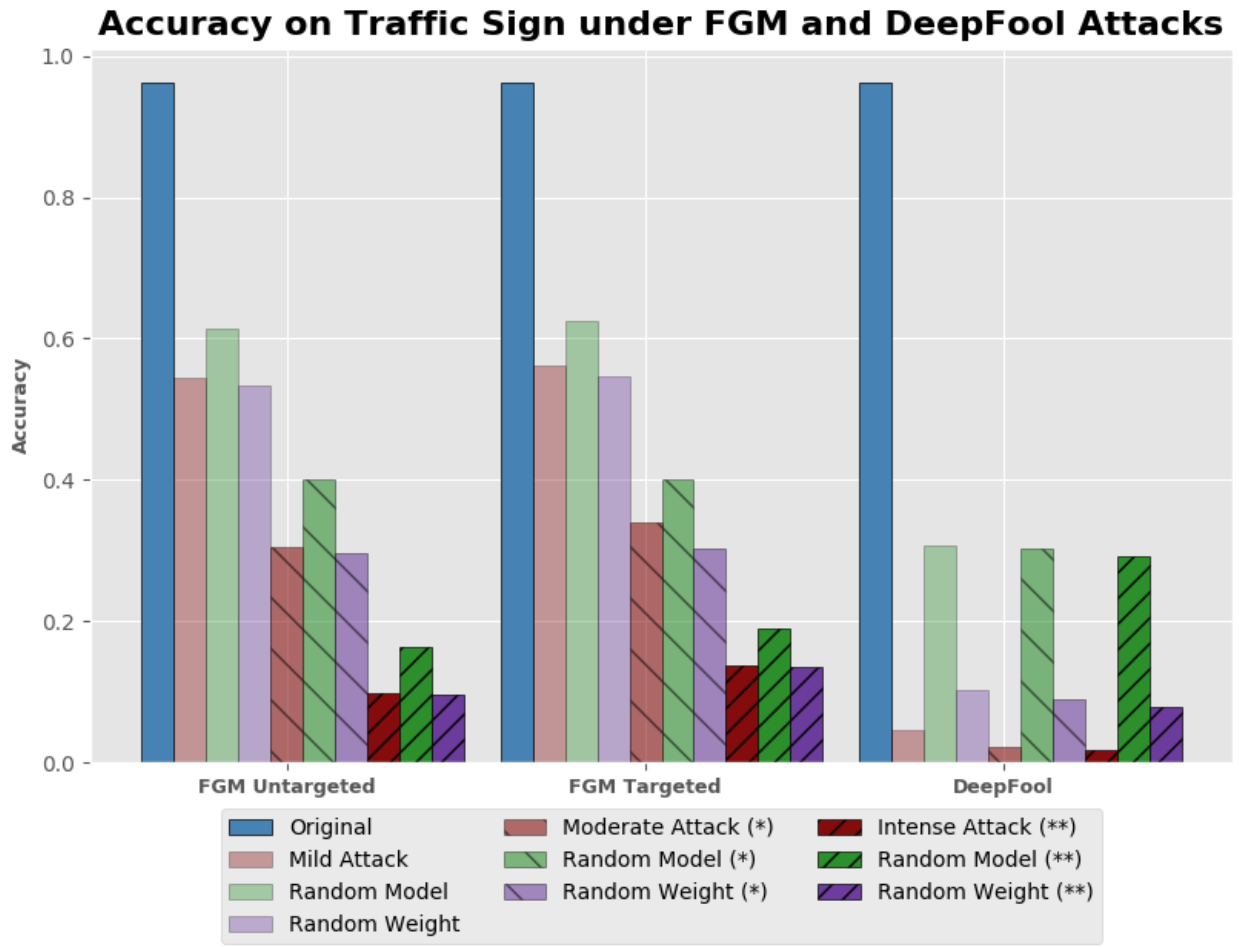
Ensemble-AdTrain: for comparison, Ensemble Adversarial Training.

Ensemble-AdTrain-Random: randomly pick 1 model from adversarially trained models.

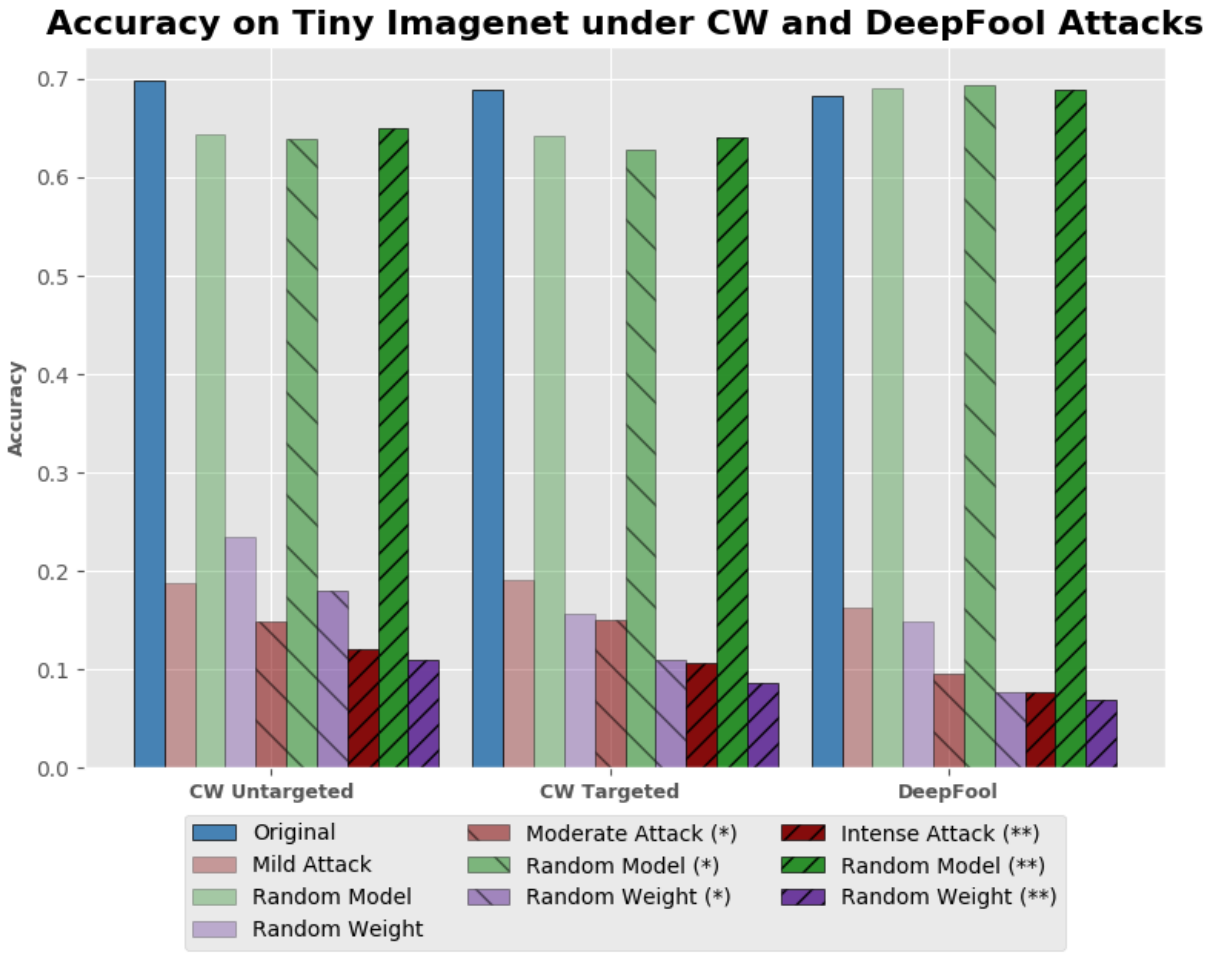
Random-Weight: randomly pick 1 M_k, add random noises to some weights.

Random-Weight-n: add random noises to every model in the pool, use majority vote.

German Traffic Sign



Tiny ImageNet



DNN classification boundary is fractured, unlike other classifiers.

Adversarial examples stem from DNN's structural defect. A more serious problem than previously imagined.

DNN has uncertainty regions and transferable adversarial regions.

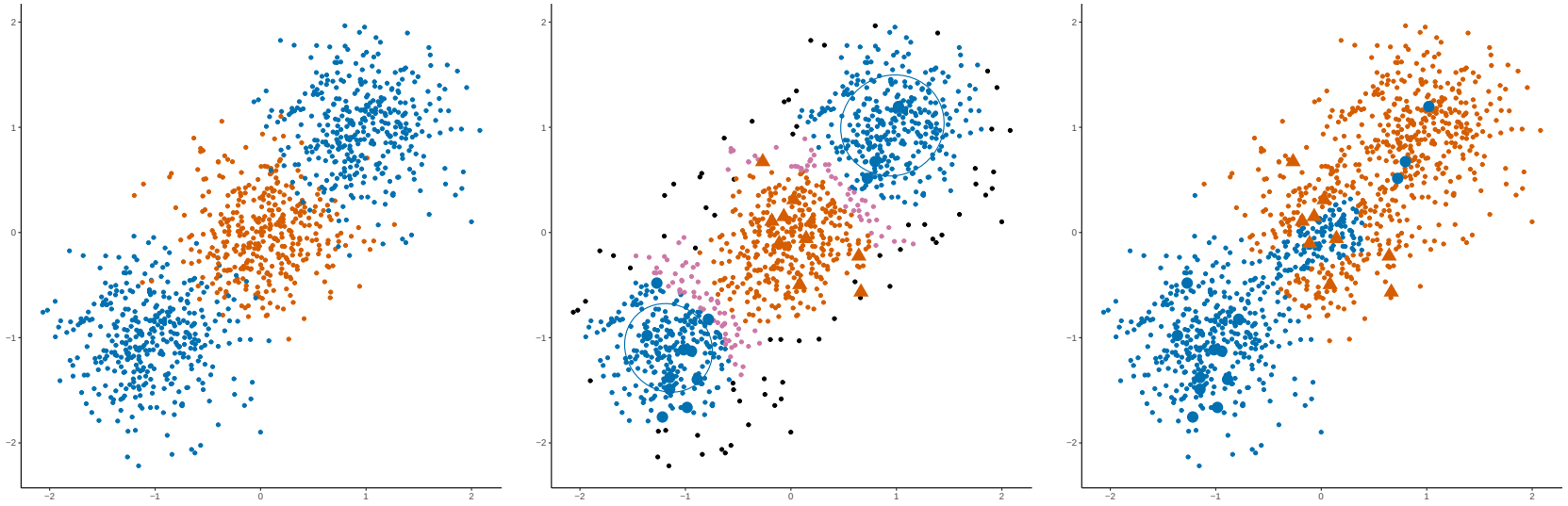
Established theoretical guarantee for DNN, generalization error as $O(\frac{c(depth,\ width)}{\sqrt{n}})$, cannot adequately describe the phenomenon of DNN adversarial examples.

Randomization strategy can improve the robustness against adversarial examples.

Improving a single DNN's robustness needs more effort.

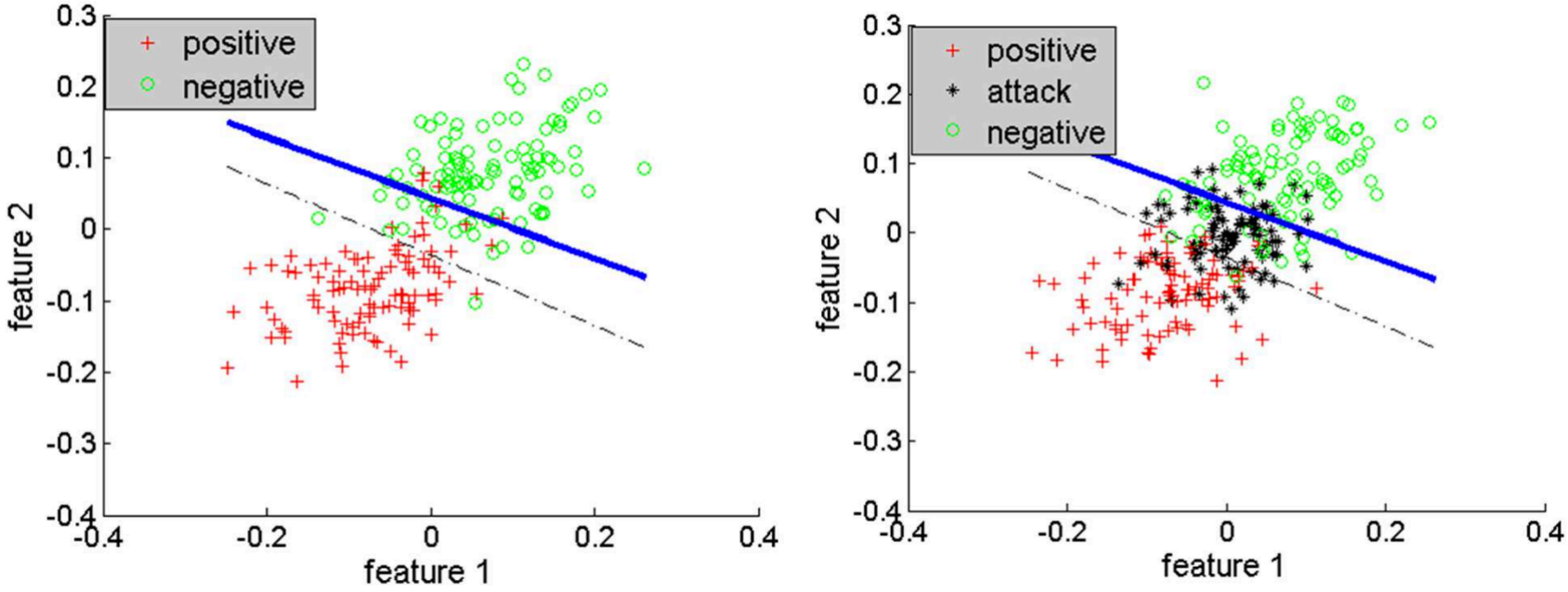
A Grid Adversarial Clustering Algorithm

Compare with a semi-supervised learning algorithm, S4VM. $\alpha = 0.6$.



Left: actual clusters with blue for normal and orange for abnormal; Middle: our ADClust with purple for unlabeled; Right: S4VM. Solid circles (normal) and solid triangles (abnormal) are known correctly labeled objects.

Adversarial SVM



AD-SVM solves a convex optimization problem where the constraints are tied to adversarial attack models.

Need to improve ML techniques for adversarial environment.

Game theory equilibrium solution provides a conservative strategy facing adversaries for many classifiers.