

**Carnegie  
Mellon  
University**

**Software Engineering  
Institute**

# Journal Club Discussion:

Adversarial Examples Are Not Bugs, They Are Features

---

**JUNE 27, 2019**

Paper Authors: Andrew Ilyas, Shibani Santurkar, Dimitri Tsipras,  
Logan Engstrom, Brandon Tan, and Aleksander Madry

Presenter: Nathan VanHoudnos

# Document Markings

Copyright 2019 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

## GOVERNMENT PURPOSE RIGHTS – Technical Data

Contract No.: FA8702-15-D-0002

Contractor Name: Carnegie Mellon University

Contractor Address: 4500 Fifth Avenue, Pittsburgh, PA 15213

The Government's rights to use, modify, reproduce, release, perform, display, or disclose these technical data are restricted by paragraph (b)(2) of the Rights in Technical Data—Noncommercial Items clause contained in the above identified contract. Any reproduction of technical data or portions thereof marked with this legend must also reproduce the markings.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

DM19-0709

# Adversarial Examples Are Not Bugs, They Are Features

Andrew Ilyas\*  
MIT  
ailyas@mit.edu

Shibani Santurkar\*  
MIT  
shibani@mit.edu

Dimitris Tsipras\*  
MIT  
tsipras@mit.edu

Logan Engstrom\*  
MIT  
engstrom@mit.edu

Brandon Tran  
MIT  
btran115@mit.edu

Aleksander Mađry  
MIT  
madry@mit.edu

## Abstract

Adversarial examples have attracted significant attention in machine learning, but the reasons for their existence and pervasiveness remain unclear. We demonstrate that adversarial examples can be directly attributed to the presence of *non-robust features*: features (derived from patterns in the data distribution) that are highly predictive, yet brittle and (thus) incomprehensible to humans. After capturing these features within a theoretical framework, we establish their widespread existence in standard datasets. Finally, we present a simple setting where we can rigorously tie the phenomena we observe in practice to a *misalignment* between the (human-specified) notion of robustness and the inherent geometry of the data.

## Claims to discuss:

1. Adversarial examples are a result of *non-robust features* (... derived from patterns in the data distribution).
2. After capturing these features within a theoretical framework, **we establish their widespread existence in standard datasets.**

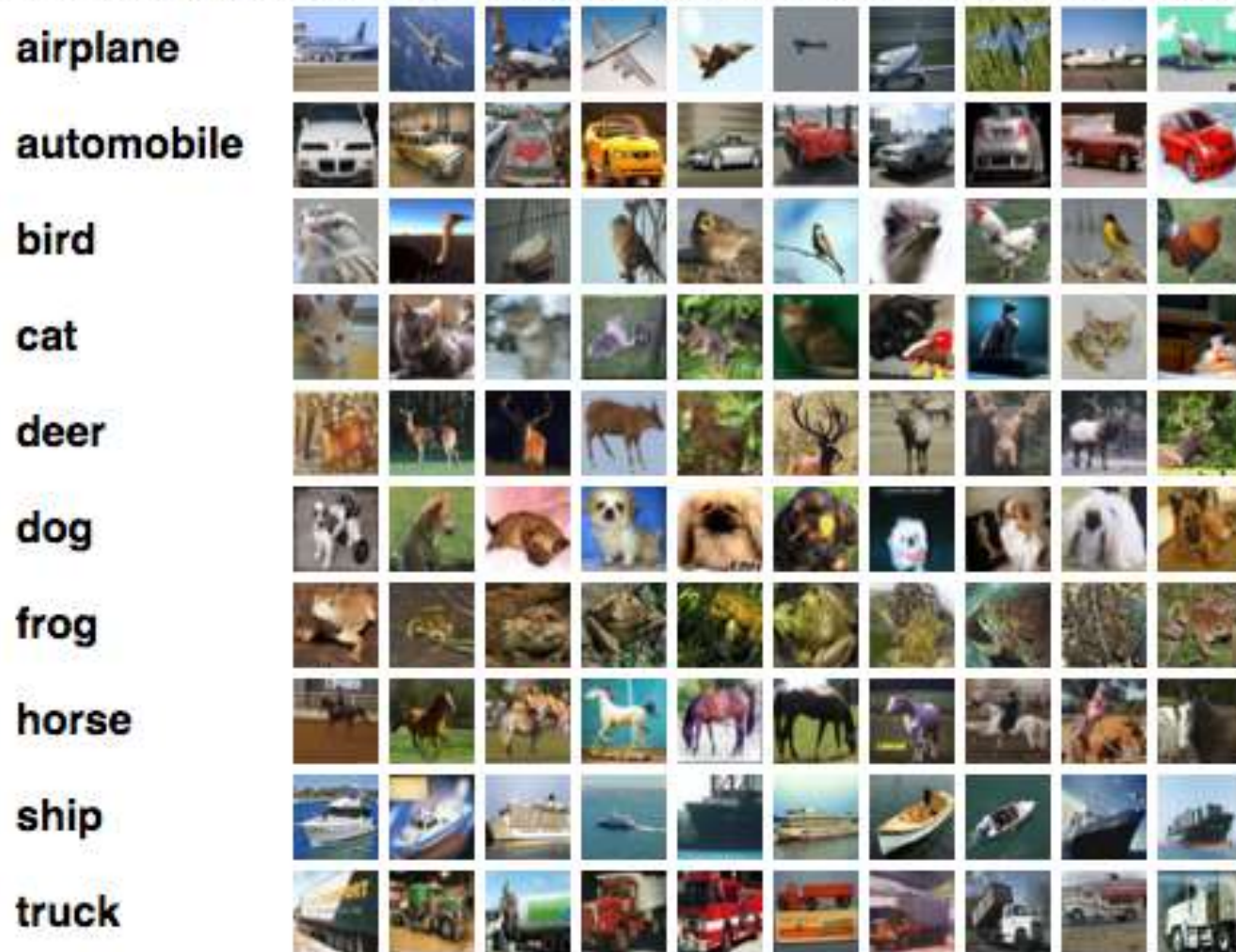
# General Ilyas et al. (2019) process

1. Collect data
2. Choose a classification model
3. Estimate the parameters in the model (fit the model to the data)
4. Estimate a performance metric

Ilyas et al. (2019) holds the model fixed and varies each other component.

# First dataset: CIFAR-10

Here are the classes in the dataset, as well as 10 random images from each:

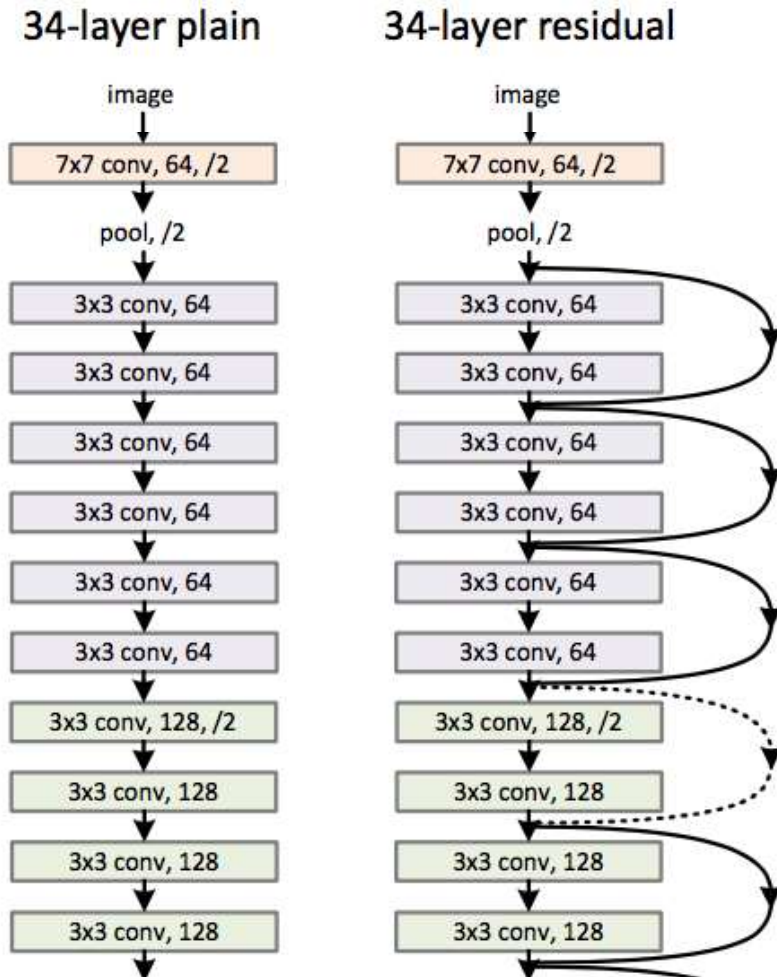


60000 32x32 color images in 10 classes  
with 6000 images per class

<https://www.cs.toronto.edu/~kriz/cifar.html>

# Ilyas et al. (2019) model: ResNet-50

ResNet-50: A 50 layer CNN with residual connections.



## ResNet 50

- Approx. 800K parameters.

## CIFAR-10

- 50K training images, 10K test images

method			error (%)
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	<b>6.43</b> (6.61±0.16)
ResNet	1202	19.4M	7.93

Table 6. Classification error on the **CIFAR-10** test set. All methods are with data augmentation. For ResNet-110, we run it 5 times and show “best (mean±std)” as in [43].

# The two ways Ilyas et al. estimate parameters

“Standard”

Choose  $\theta$  to minimize

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [\mathcal{L}_{\theta}(x, y)]$$

- Recall for ResNet-50,  $\theta$  is of dimension 800K
- $\theta$  is estimated with stochastic gradient descent

This is maximum likelihood.

“Adversarial”

Choose  $\theta$  to minimize

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ \max_{\delta \in \Delta(x)} \mathcal{L}_{\theta}(x + \delta, y) \right].$$

- $\Delta(\cdot)$  defines the adversary who is trying to maximize the loss
- $\Delta(x)$  is 7 steps of a PGD attack with an L2 norm, where each step is  $\epsilon/5$  epsilon is the adversary budget

This is empirical minimax estimation.

# Two metrics Ilyas et al. use to estimate performance

Accuracy

$$\hat{y} = \text{model}(x, \hat{\theta})$$

Accuracy =  $\text{count}(\hat{y} = y) / \text{count}(\text{all cases})$

Adversarial Accuracy

$$\hat{y}_{\text{attacked}} = \text{model}(x_{\text{attack}}, \hat{\theta})$$

Where  $x_{\text{attack}}$  is calculated with ResNet-50( $\hat{\theta}$ ) on

- 2,500 steps of PGD attack, or
- 1,000 steps of Carlini-Wagner L2 attack with grid search

Adversarial Accuracy

: Replace  $\hat{y}$  with  $\hat{y}_{\text{attacked}}$



# Experiment #1

Data: CIFAR-10

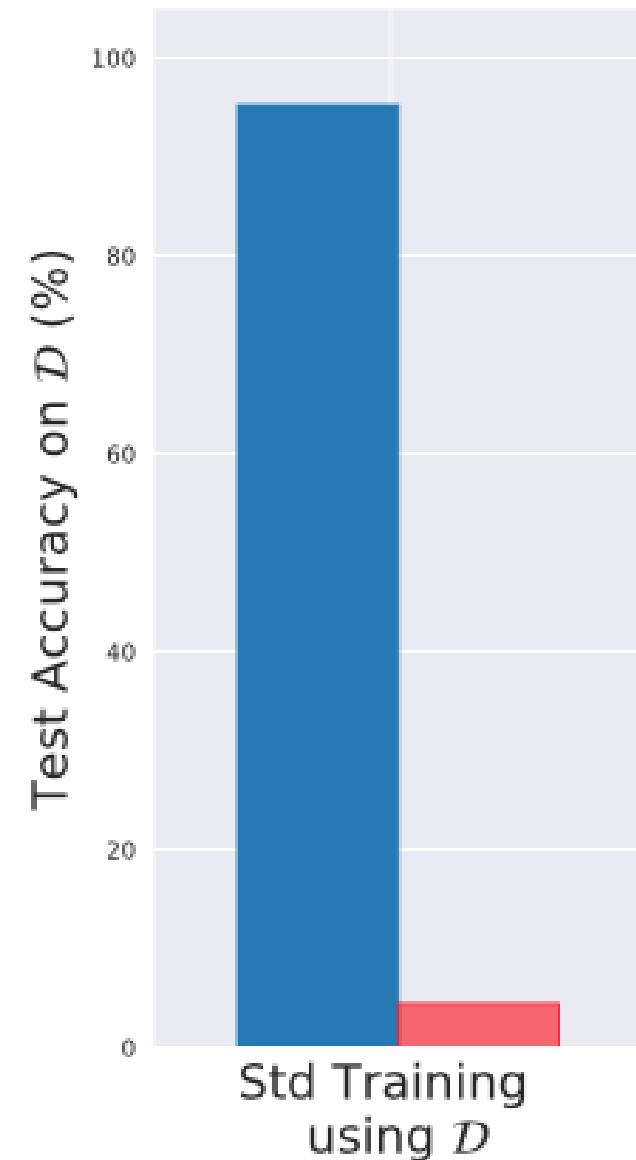
Model: ResNet-50

Estimation: SGD / maximum likelihood

Metrics: Accuracy and Adversarial Accuracy

Answer: Well known. Maximum likelihood estimators can fail spectacularly when evaluated on “worst case” data distributions.

■ Std accuracy    ■ Adv accuracy ( $\epsilon = 0.25$ )



# Experiment #2

Data: CIFAR-10

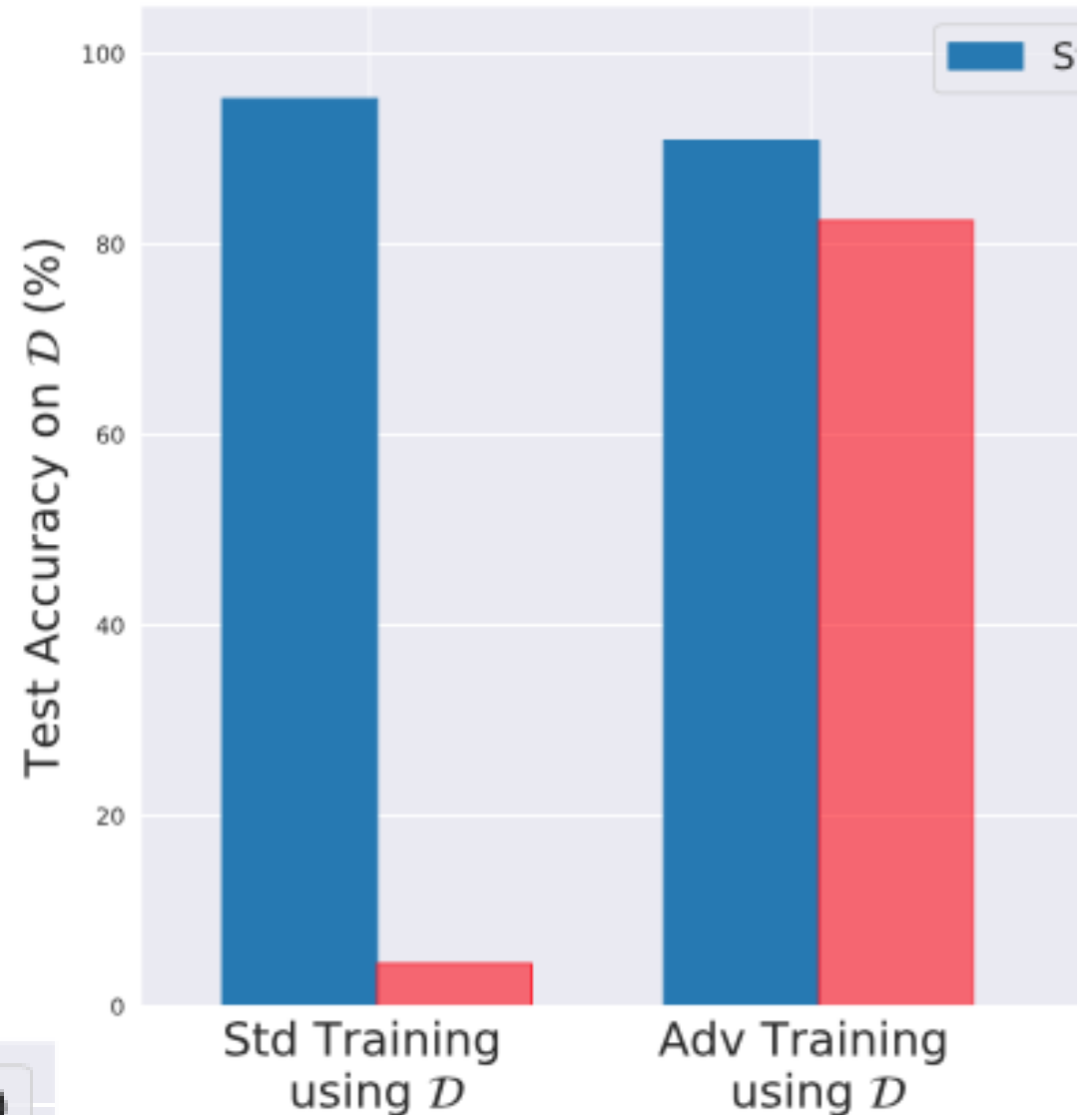
Model: ResNet-50

Estimation: Adversarial / Empirical Minimax

Metrics: Accuracy and Adversarial Accuracy

Answer: Well known. Minimax works well, especially when the attack models are aligned.

■ Std accuracy    ■ Adv accuracy ( $\epsilon = 0.25$ )



# An aside: which estimation procedure is better?

Posterior Predictive Checking (PPC):

- Check a generative model by generating new data and checking if the generated data matched the real data

One way to do this for ResNet-50 (which isn't generative) is to choose an  $x_r$  such that

$$\min_{x_r} \|g(x_r) - g(x)\|_2,$$

Where  $g(\cdot)$  is the penultimate layer of the neural network

# For example...

$$\min_{x_r} \|g(x_r) - g(x)\|_2,$$



$x$

$g(\cdot)$  maximum likelihood (Exp 1)

$g(\cdot)$  minimax (Exp 2)

$x_r$



**NOTE:** The only difference between these is the method of training, i.e. maximum likelihood or minimax

# More examples of $\min_{x_r} \|g(x_r) - g(x)\|_2,$

“deer”

“truck”

“cat”

“bird”

“ship”

$x$



$x_r$  with  $g(\cdot)$  minimax  
(Exp 2)



$x_r$  with  $g(\cdot)$  maximum  
likelihood (Exp 1)



# Additional Ilyas et al. experiments

## Experiment 3

1. Train ResNet-50 on CIFAR to minimize  $\theta$  s.t.

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [\mathcal{L}_{\theta}(x,y)]$$

2. Synthesize an approximation to CIFAR with  $g(\cdot)$  from Step 1.

$$\min_{x_r} \|g(x_r) - g(x)\|_2,$$

3. Train ResNet-50 on the  $x_r$  from steps 1 and 2.
4. Check accuracy and adversarial accuracy

## Experiment 4

1. Train ResNet-50 on CIFAR to minimize  $\theta$  s.t.

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ \max_{\delta \in \Delta(x)} \mathcal{L}_{\theta}(x + \delta, y) \right].$$

2. Synthesize an approximation to CIFAR with  $g(\cdot)$  from Step 1.

$$\min_{x_r} \|g(x_r) - g(x)\|_2,$$

3. Train ResNet-50 on the  $x_r$  from steps 1 and 2.
4. Check accuracy and adversarial accuracy

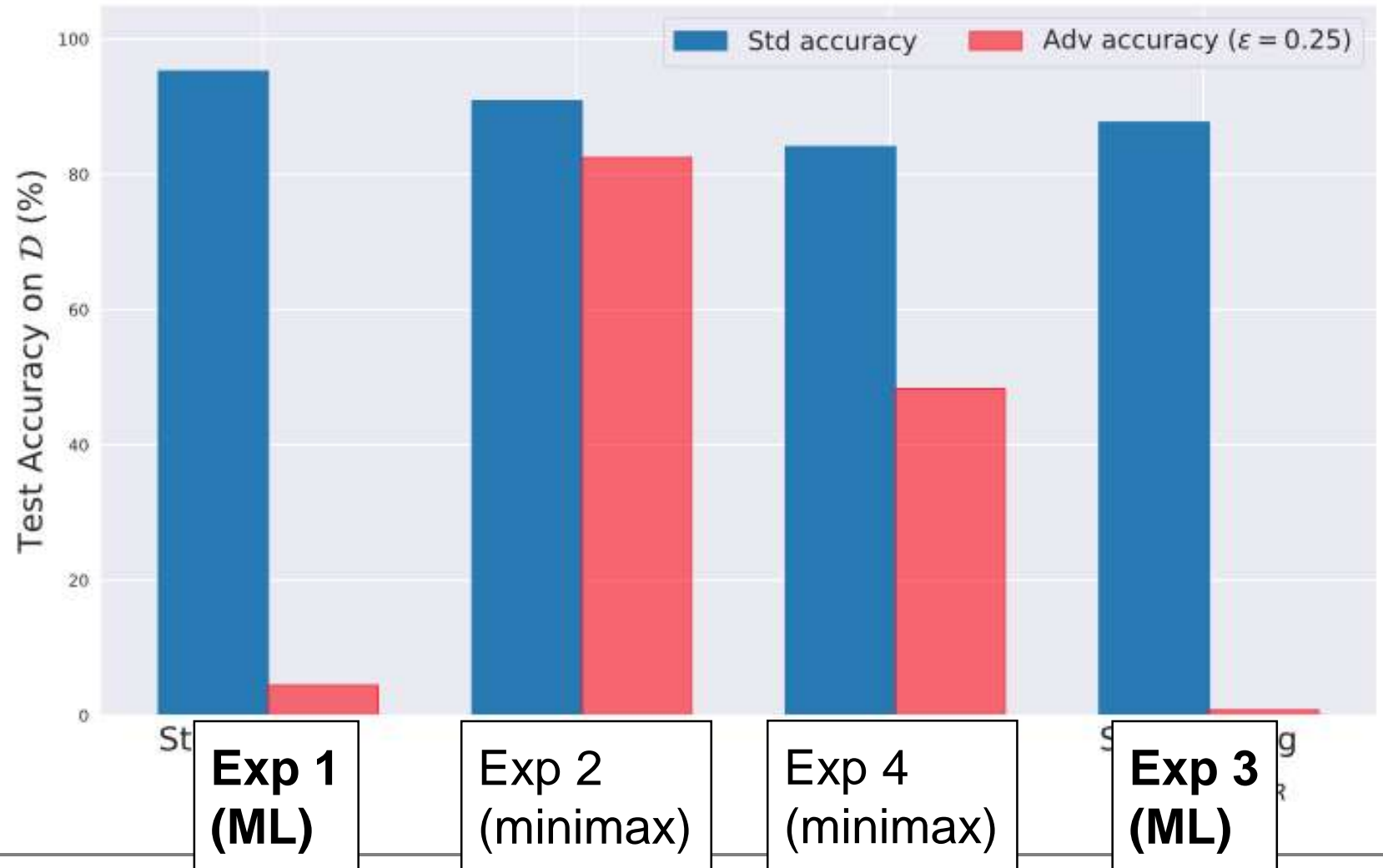
# Ilyas et al. results and discussion

Ilyas claims:

- “Adversarial examples are not bugs, they are features ... (derived from patterns in the data distribution).”

**This does not follow.**

- The only variation between Exp 3 and Exp 4 is the method of training to generate the  $g(\cdot)$  for  $x_r$



# Some comments to spark discussion

- The adversarial accuracy metric and PPC results suggest that minimax is a better estimation procedure.
- “Adversarial Examples are not Bugs, they are Features” are NOT features of the data, they are properties of the estimation procedure, specifically, the loss.

Claims to discuss:

1. Adversarial examples are a result of *non-robust features* (... *derived from patterns in the data distribution*).
2. After capturing these features within a theoretical framework, **we establish their widespread existence in standard datasets.**



Dr. Nathan VanHoudnos (van-HOD-ness)  
Senior Machine Learning Research Scientist  
Software Engineering Institute  
Carnegie Mellon University  
Low: nmvanhoudnos@cert.org  
High: nathan.vanhoudnos\_CTR@af.ic.gov

U.S. Mail  
Software Engineering Institute  
4500 Fifth Avenue  
Pittsburgh, PA 15213-2612 USA

Website  
[www.sei.cmu.edu/contact.cfm](http://www.sei.cmu.edu/contact.cfm)

