Lessons Learned from Evaluating the Robustness of Defenses to Adversarial Examples

Nicholas Carlini
Google Research

Lessons Learned from Evaluating the Robustness of Defenses to Adversarial Examples

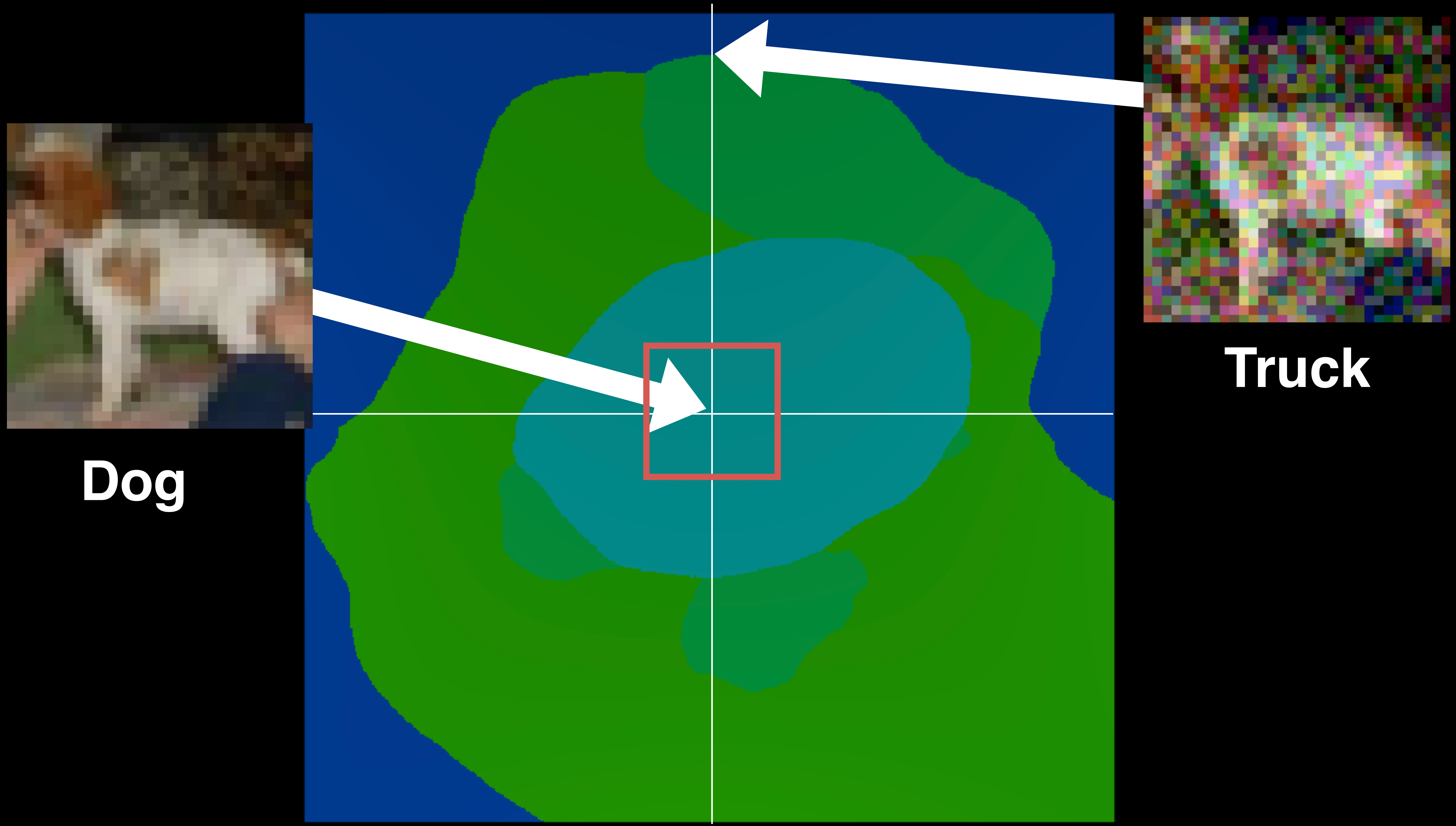
Lessons Learned from Evaluating the Robustness of Defenses to **Adversarial Examples**

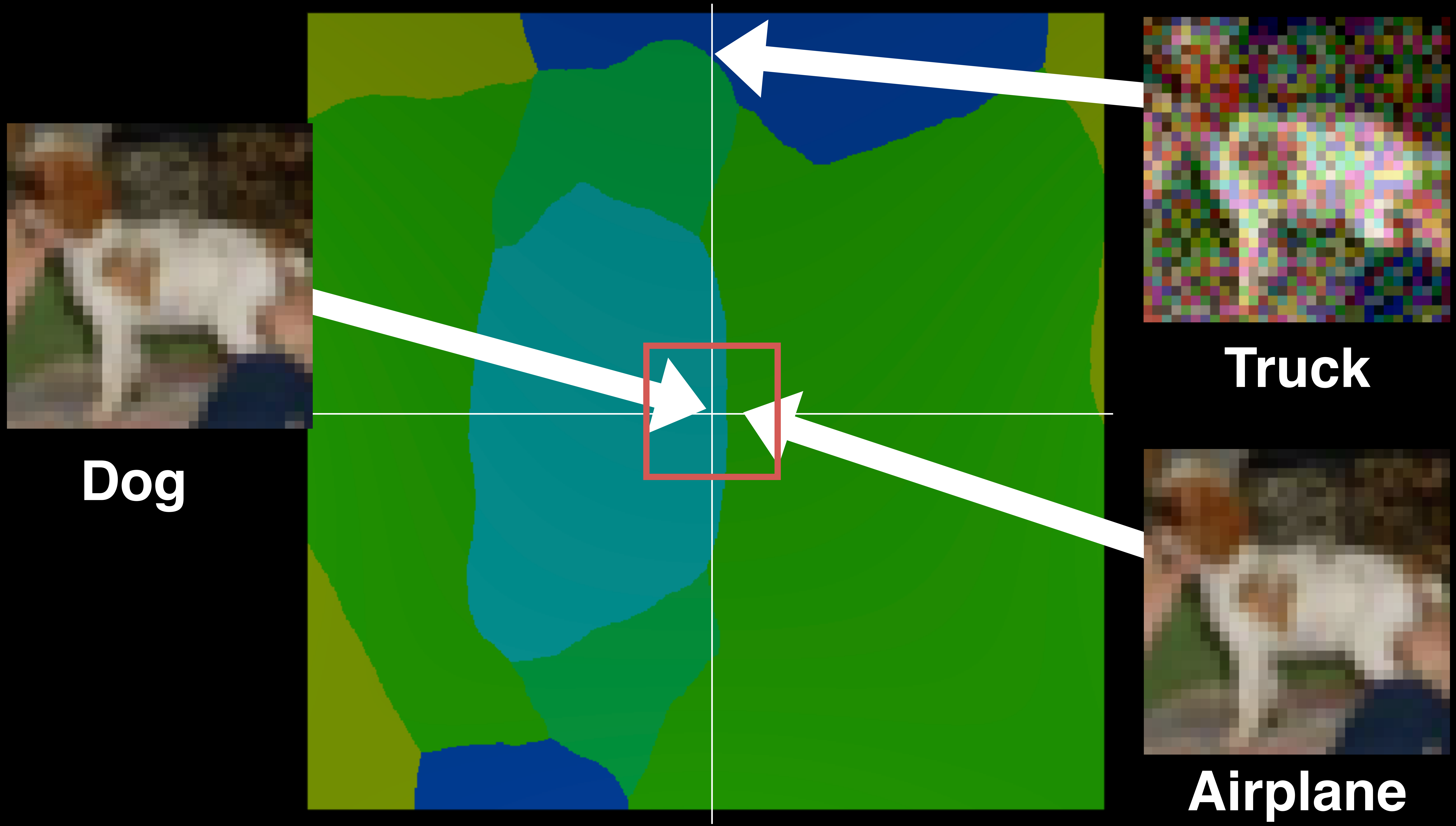
adversarial
perturbation

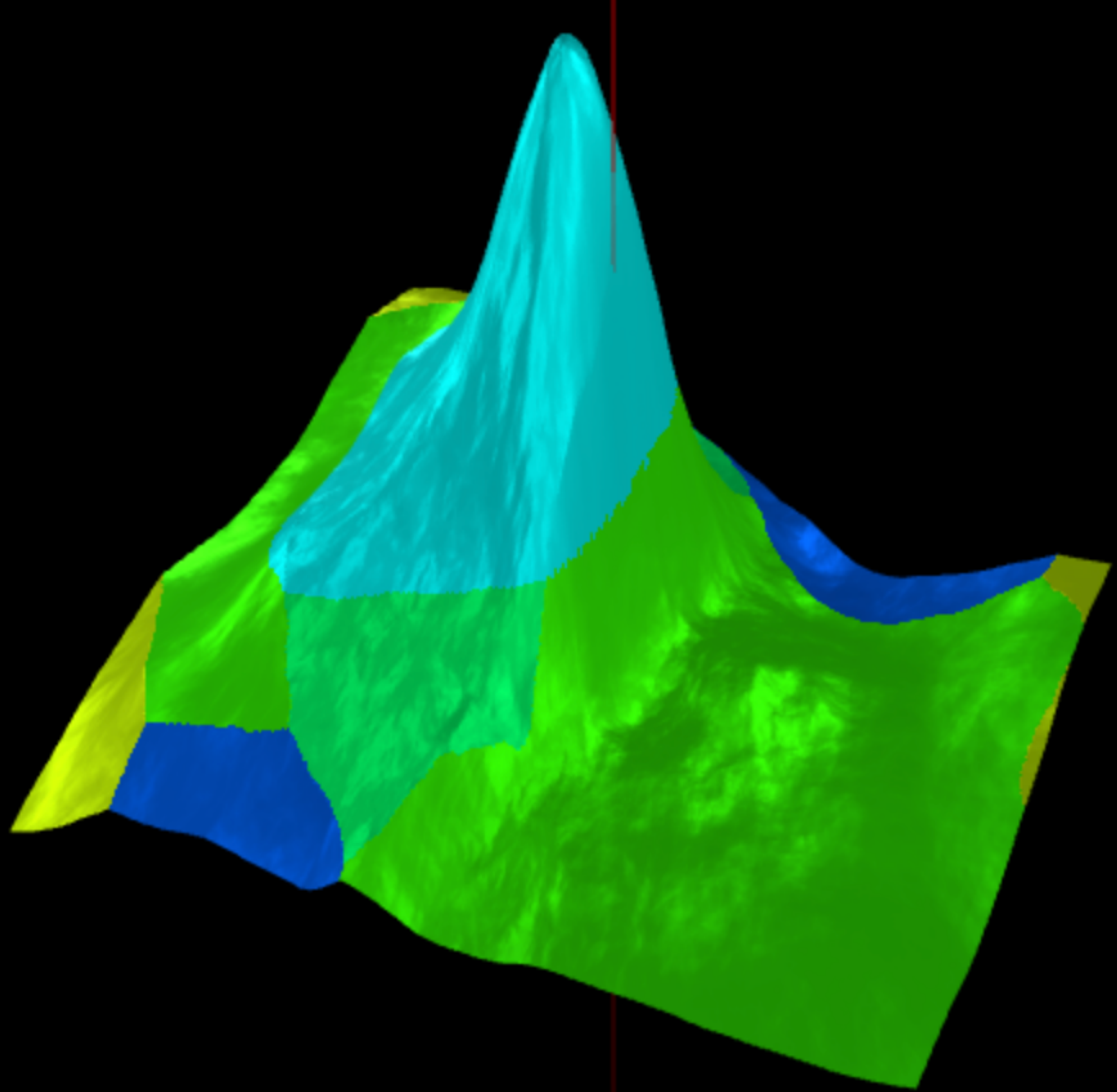
88% tabby cat

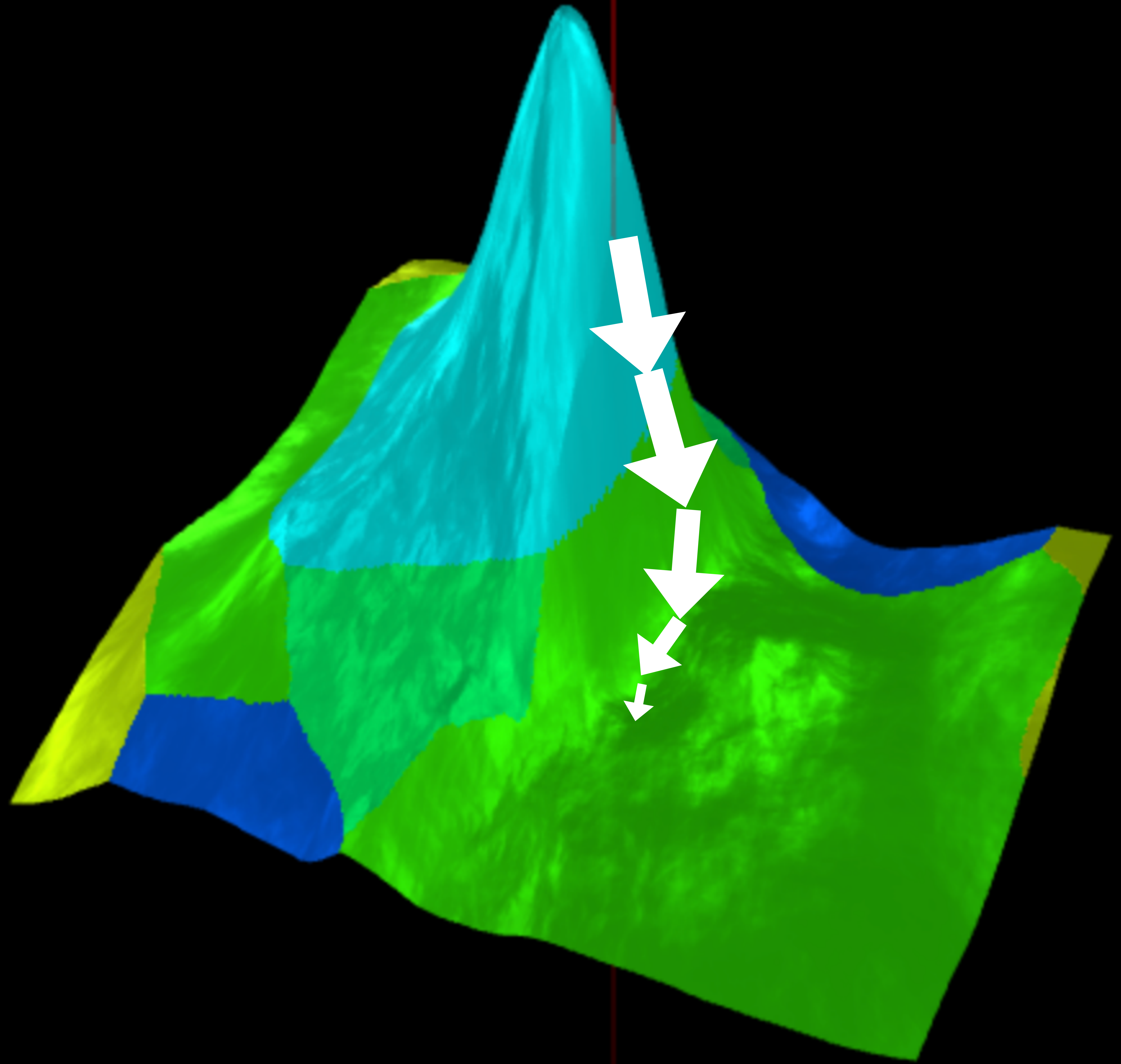
99% guacamole

How do we generate adversarial examples?


Dog
Truck


Dog
Truck
Airplane





Threat Models

A threat model is a **formal** statement defining when a system is intended to be secure.

What dataset is considered?

Adversarial example definition?

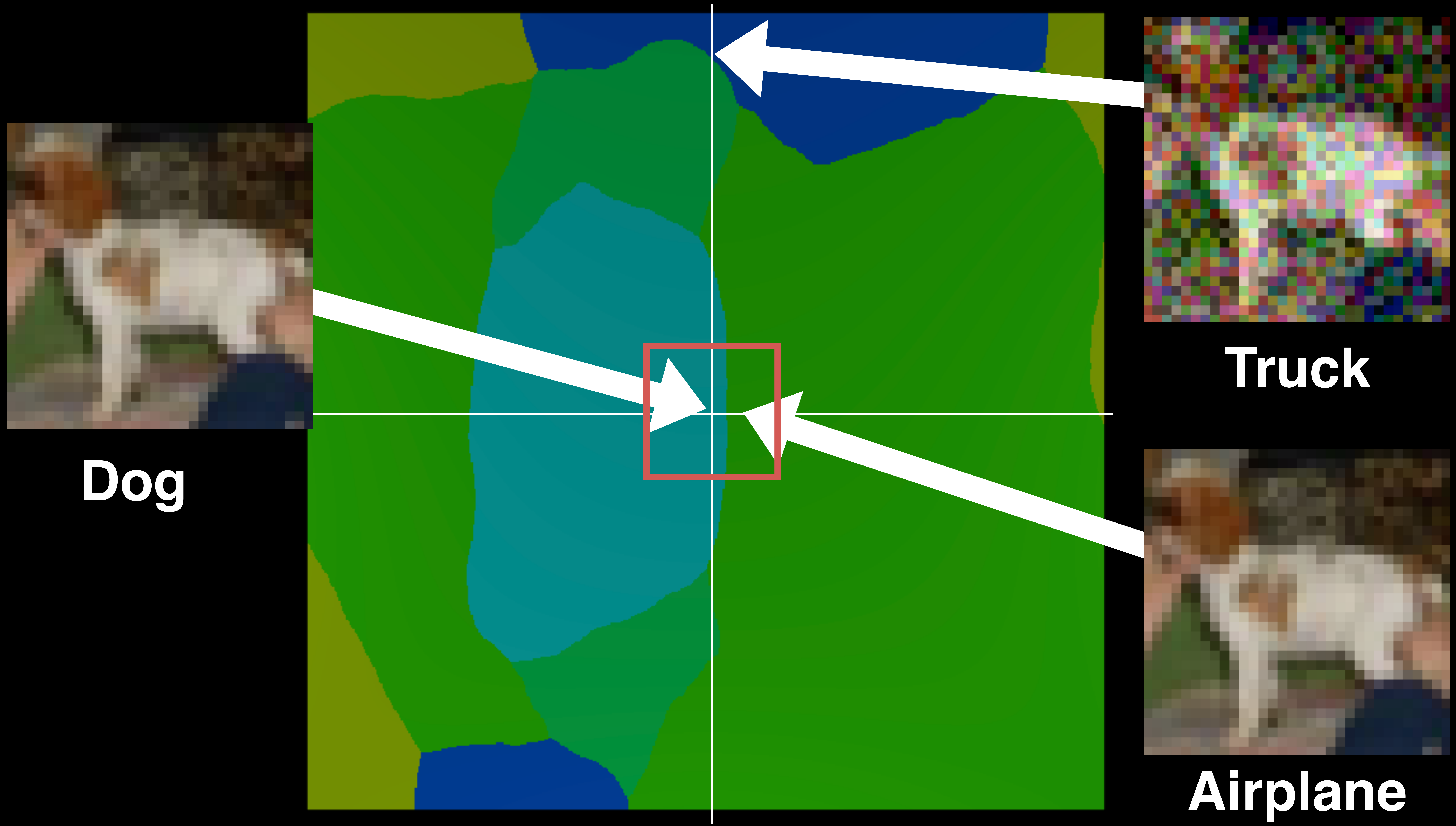
What does the attacker know?
(model architecture? parameters? training data? randomness?)

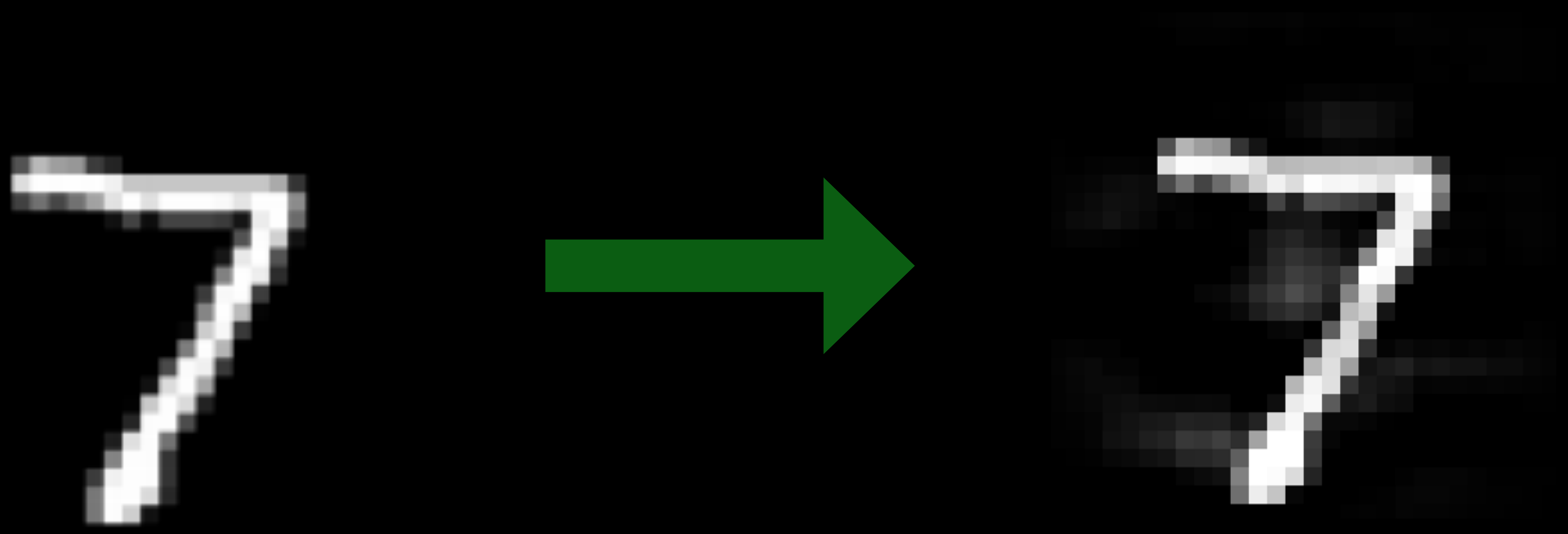
If black-box: are queries allowed?

Good Threat Model:

"Robust when L_2 distortion is less than 5, given the attacker has white-box knowledge"

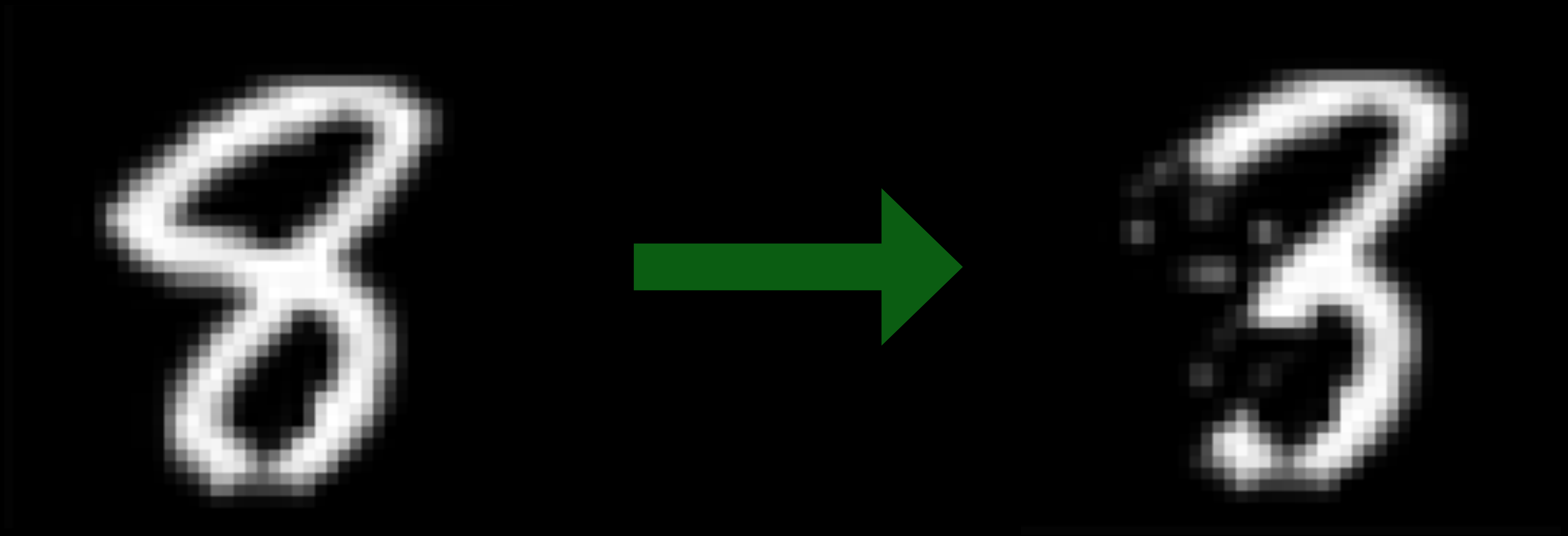
Claim: *90% accuracy on ImageNet*


Dog
Truck
Airplane



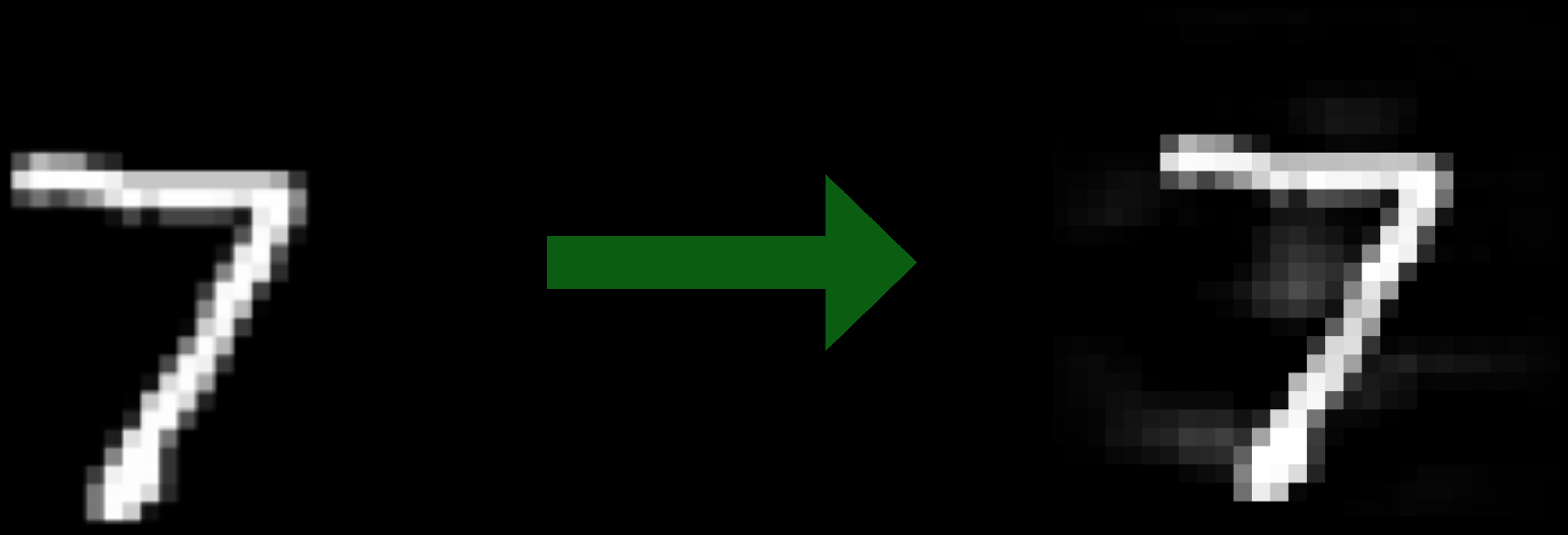
Classified as 7

Classified as 1



Classified
as 8

Classified
as 8



Classified as 7

Classified as 1

Lessons Learned from
Evaluating the Robustness of
Defenses to Adversarial Examples

A **defense** is a neural network that

1. Is accurate on the test data
2. Resists adversarial examples

This talk: non-certified defenses

For example:
adversarial training

For example:

Adversarial Training

Claim:

Neural networks don't generalize

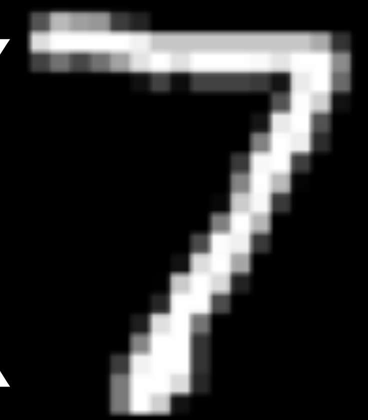
Normal Training

(, 7)

(, 3)

Training

Adversarial Training (1)

(, 7)

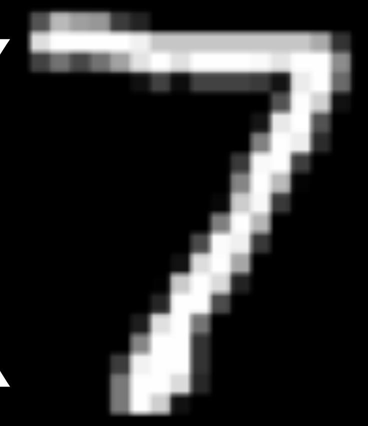
(, 3)

(, 7)

(, 3)

Attack

Adversarial Training (2)

(, 7)

(, 3)

(, 7)

(, 3)

Training

Or:

Thermometer Encoding

Claim:

Neural networks are "overly linear"

Solution

T(0.13) = 1 1 0 0 0 0 0 0 0 0

T(0.66) = 1 1 1 1 1 1 0 0 0 0

T(0.97) = 1 1 1 1 1 1 1 1 1 1

Or:

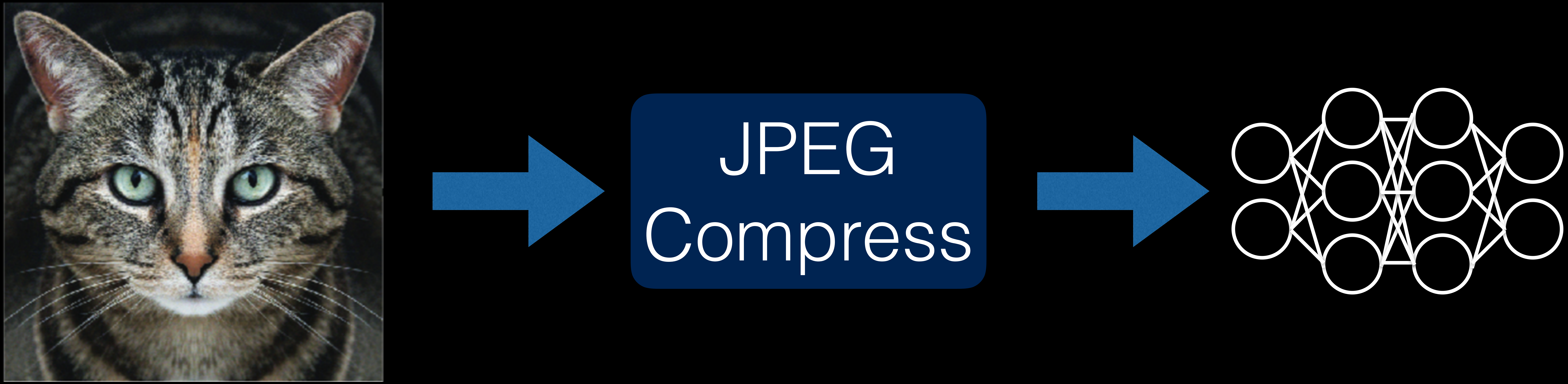
Input Transformations

Claim:

Perturbations are brittle

Solution

Solution



Lessons Learned from
Evaluating the Robustness of
Defenses to Adversarial Examples

What does it meant to evaluate the robustness of a defense?

Standard ML Pipeline

```
model = train_model(x_train, y_train)
acc, loss = model.evaluate(
              x_test, y_test)
if acc > 0.96:
    print("State-of-the-art")
else:
    print("Keep Tuning
             Hyperparameters")
```

Standard ML Pipeline

```
model = train_model(x_train, y_train)
acc, loss = model.evaluate(
                x_test, y_test)
if acc > 0.96:
    print("State-of-the-art")
else:
    print("Keep Tuning
           Hyperparameters")
```

Standard ML Pipeline

```
model = train_model(x_train, y_train)
acc, loss = model.evaluate(
                x_test, y_test)
if acc > 0.96:
    print("State-of-the-art")
else:
    print("Keep Tuning
             Hyperparameters")
```

Standard ML Evaluations

```
model = train_model(x_train, y_train)
acc, loss = model.evaluate(
                x_test, y_test)
if acc > 0.96:
    print("State-of-the-art")
else:
    print("Keep Tuning
              Hyperparameters")
```

Standard ML Evaluations

```
model = train_model(x_train, y_train)
acc, loss = model.evaluate(
                x_test, y_test)
if acc > 0.96:
    print("State-of-the-art")
else:
    print("Keep Tuning
              Hyperparameters")
```

What are robustness evaluations?

Standard ML Evaluations

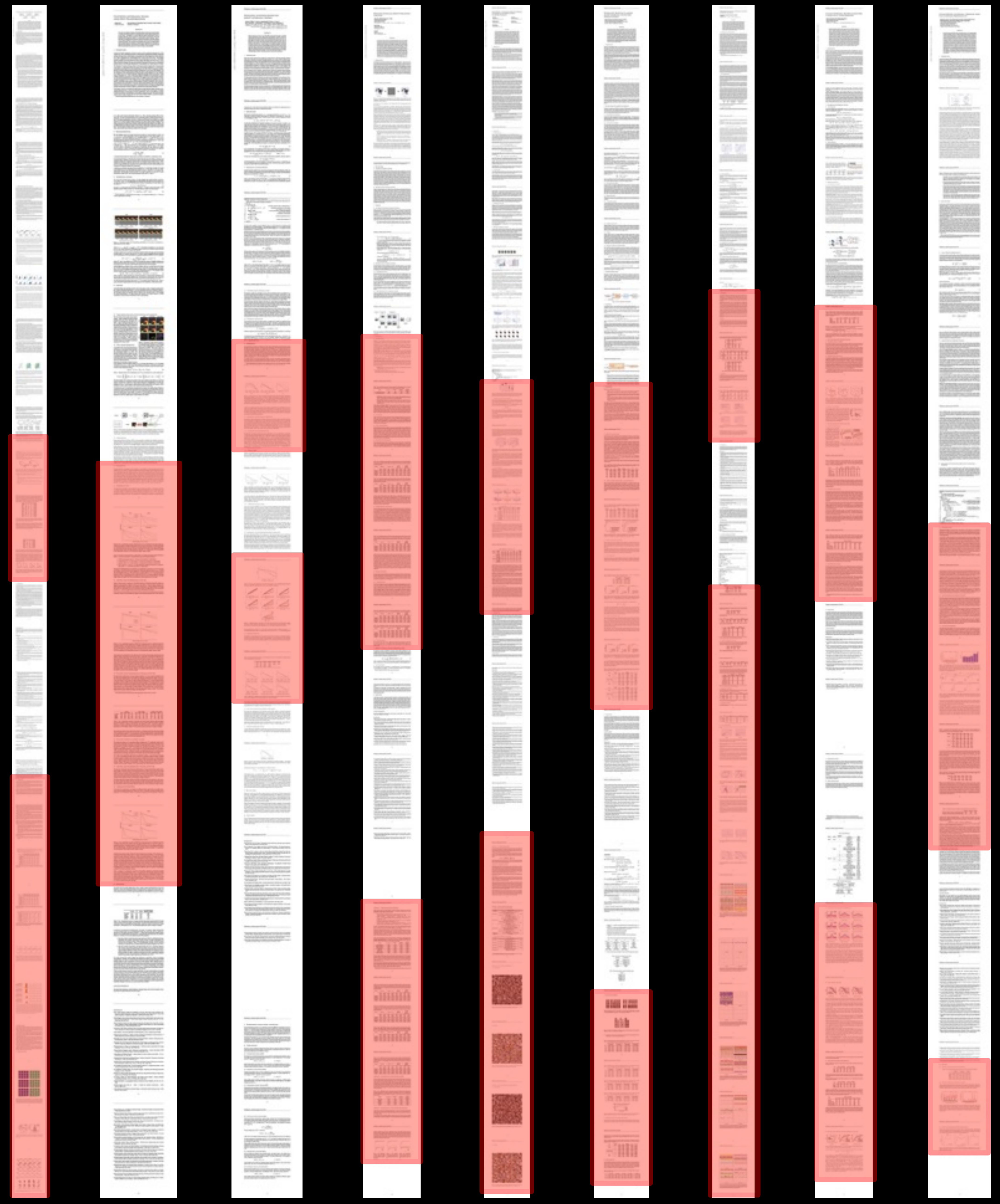
```
model = train_model(x_train, y_train)
acc, loss = model.evaluate(
              x_test, y_test)
if acc > 0.96:
    print("State-of-the-art")
else:
    print("Keep Tuning
           Hyperparameters")
```

Adversarial ML Evaluations

```
model = train_model(x_train, y_train)
acc, loss = model.evaluate(
                A(x_test), y_test)
if acc > 0.96:
    print("State-of-the-art")
else:
    print("Keep Tuning
           Hyperparameters")
```

How complete are evaluations?

Case Study:
ICLR 2018

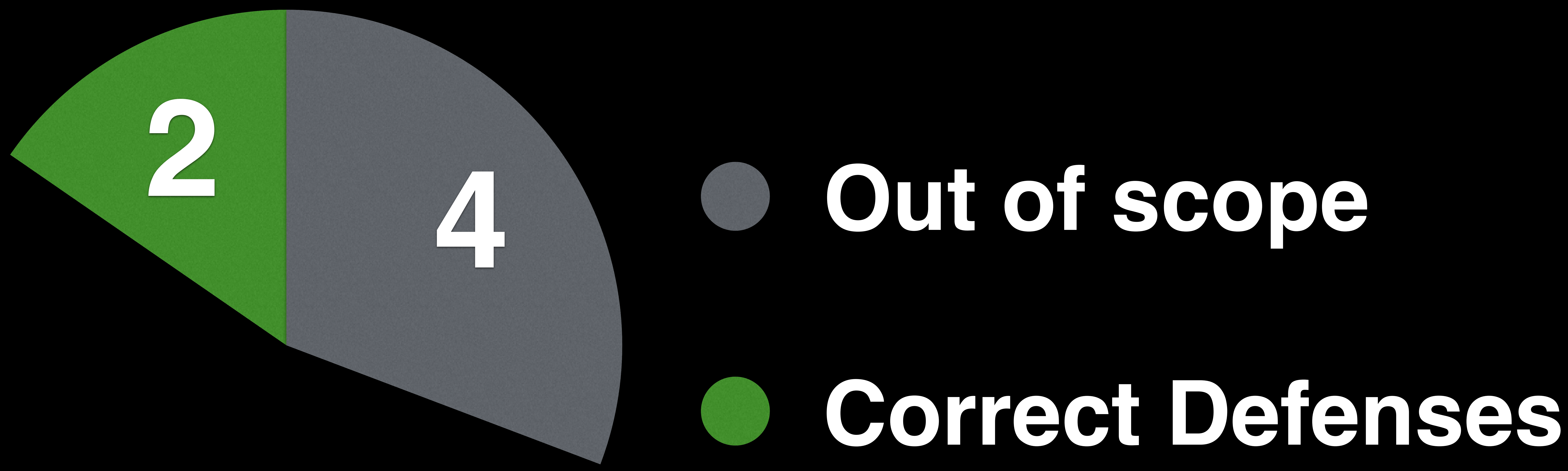


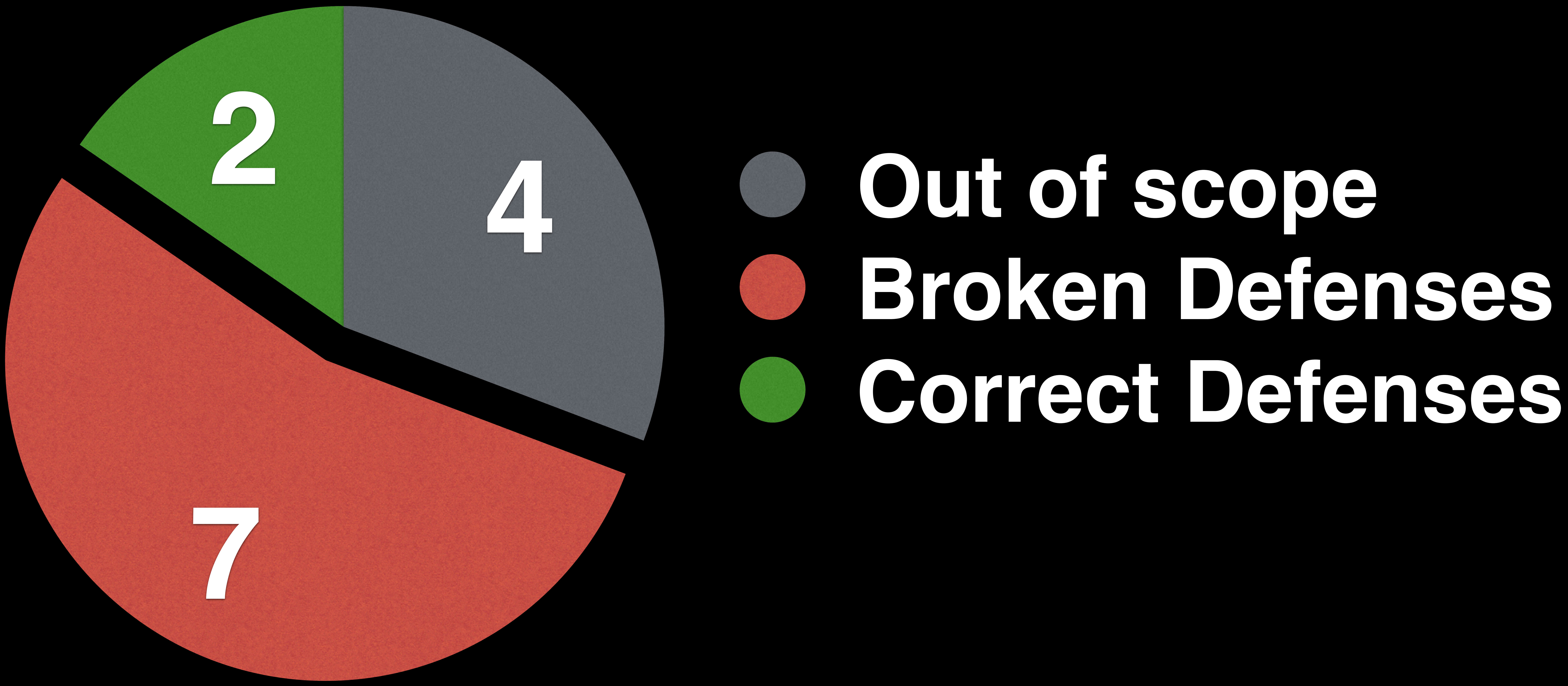
Serious effort to evaluate

By space, most papers are ½ evaluation

We re-evalauted
these defenses ...

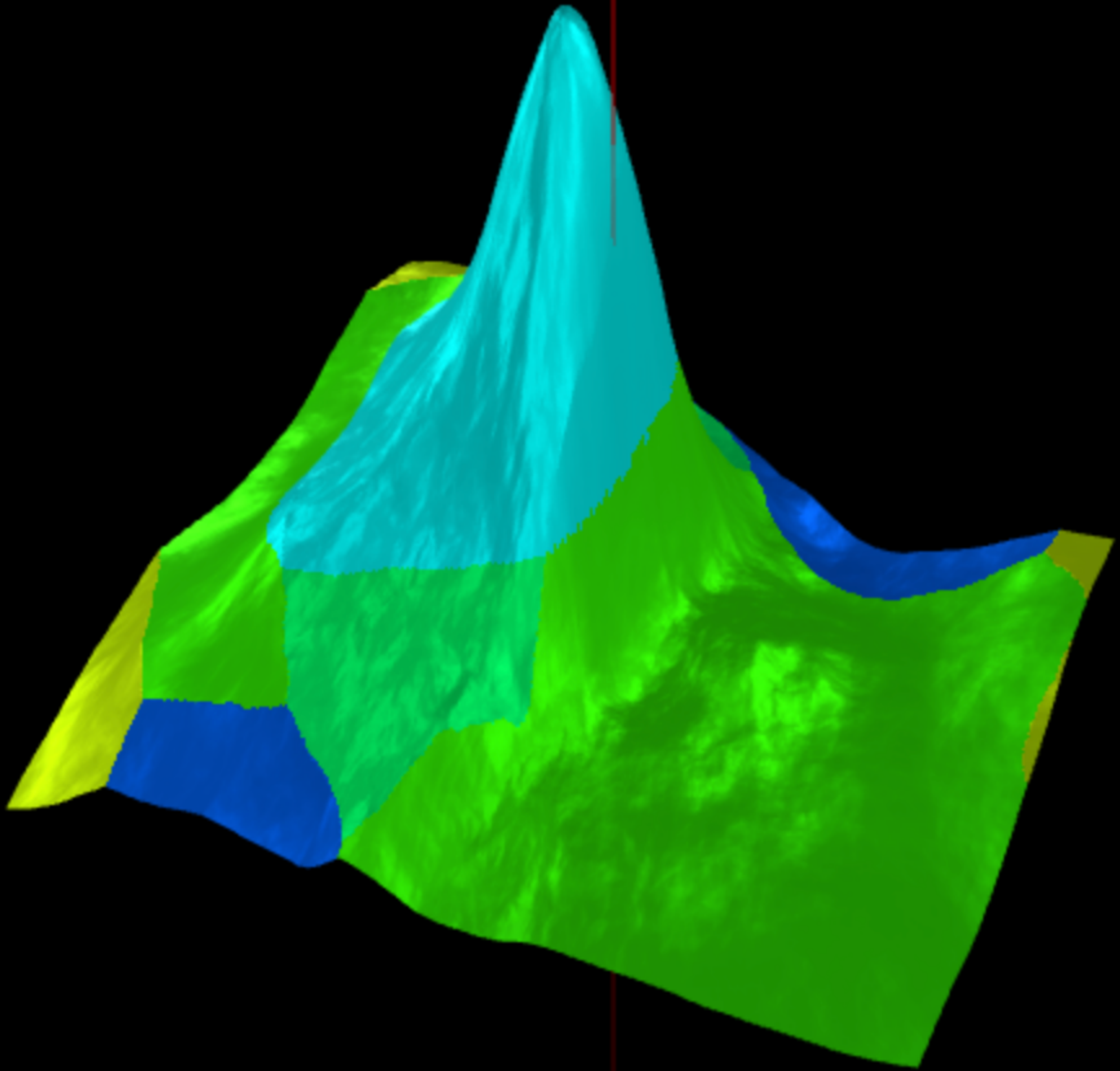

4
Out of scope

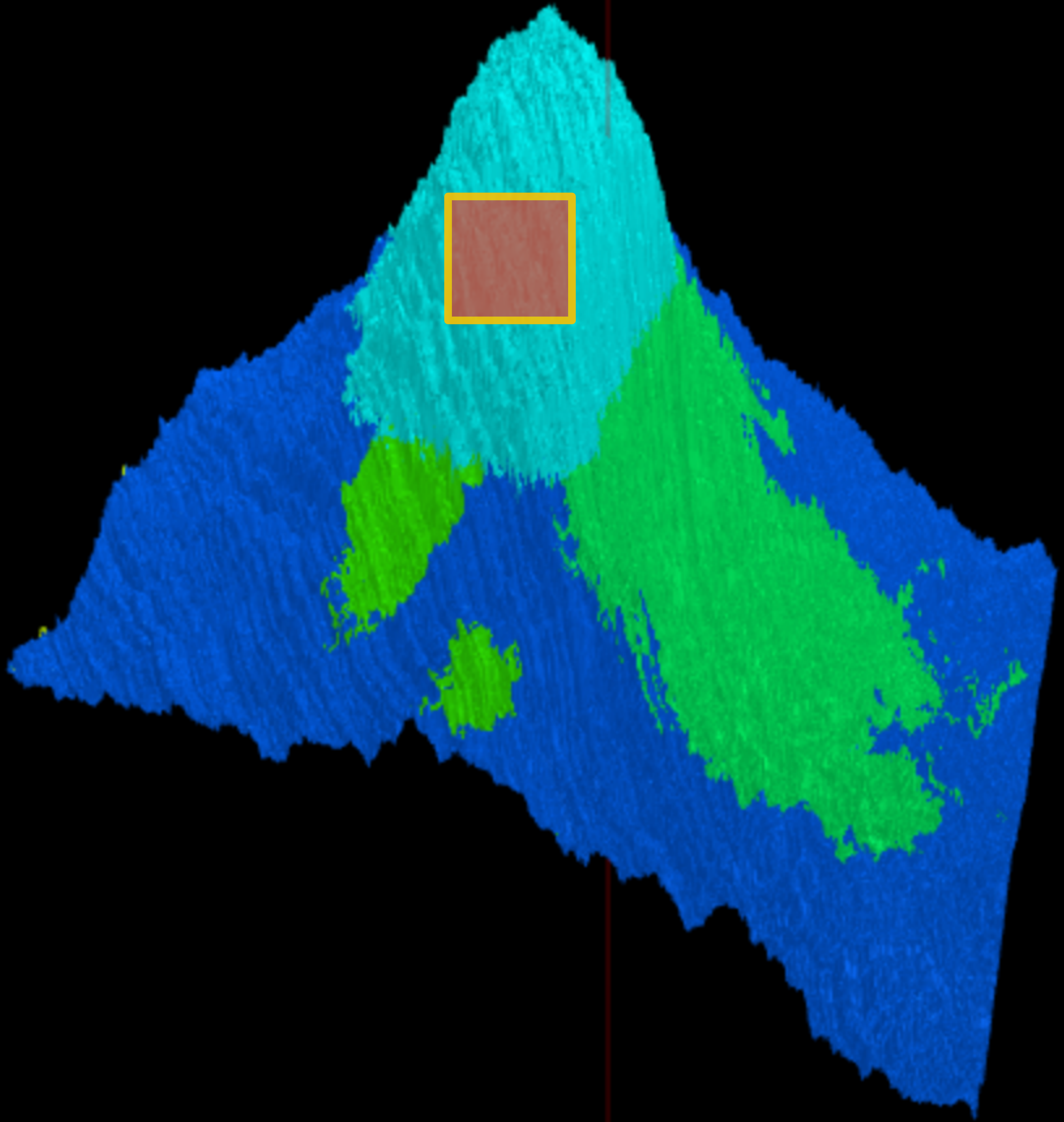

2
4
Out of scope
Correct Defenses

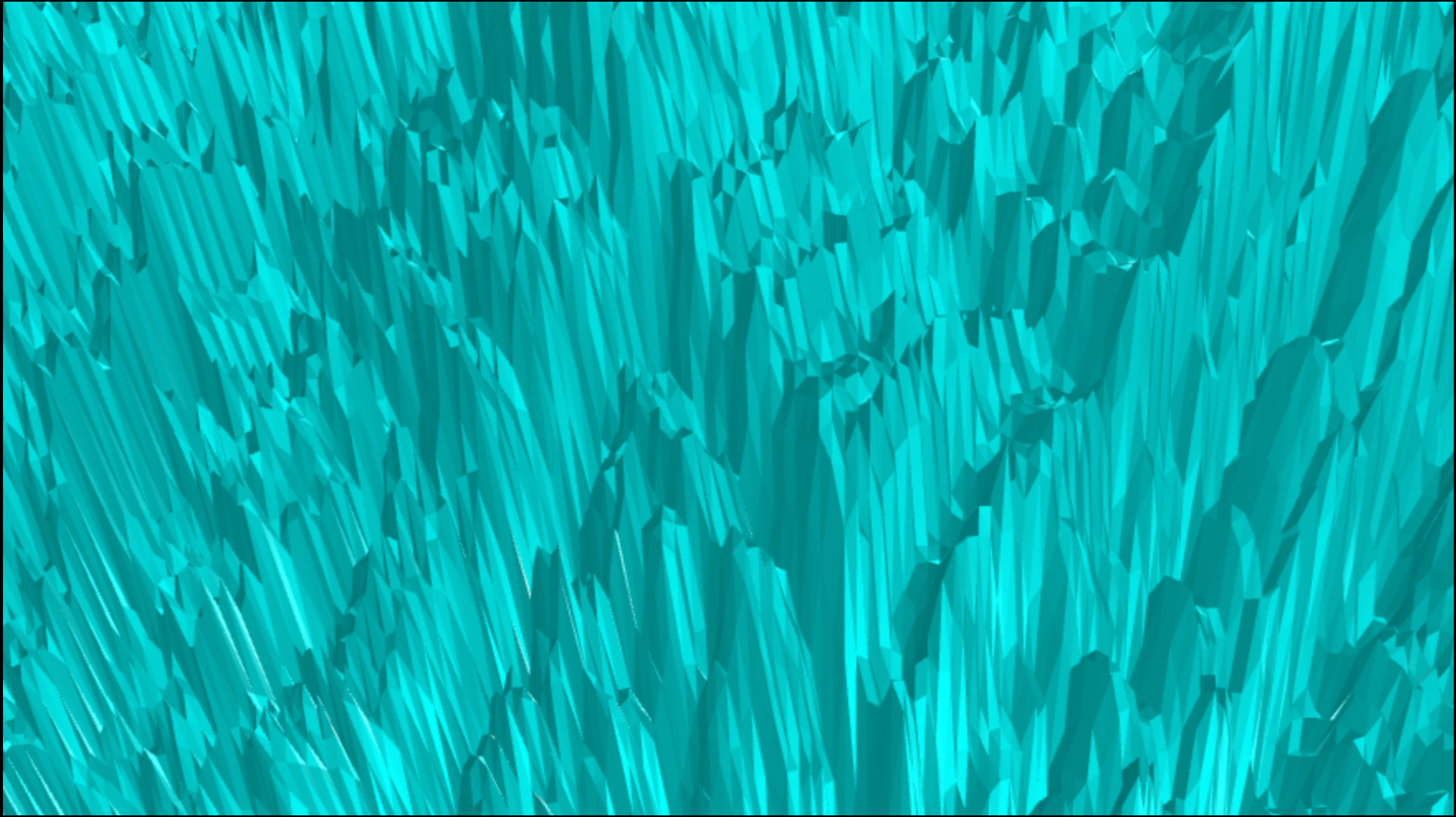


2
4
7
Out of scope
Broken Defenses
Correct Defenses

So what did defenses do?







Defensive Distillation is Not Robust to Adversarial Examples

Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods

MagNet and "Efficient Defenses Against Adversarial Attacks" are Not Robust to Adversarial Examples

Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples

On the Robustness of the CVPR 2018 White-Box Adversarial Example Defenses

Is AmI (Attacks Meet Interpretability) Robust to Adversarial Examples?

Nicholas Carlini (*Google Brain*)

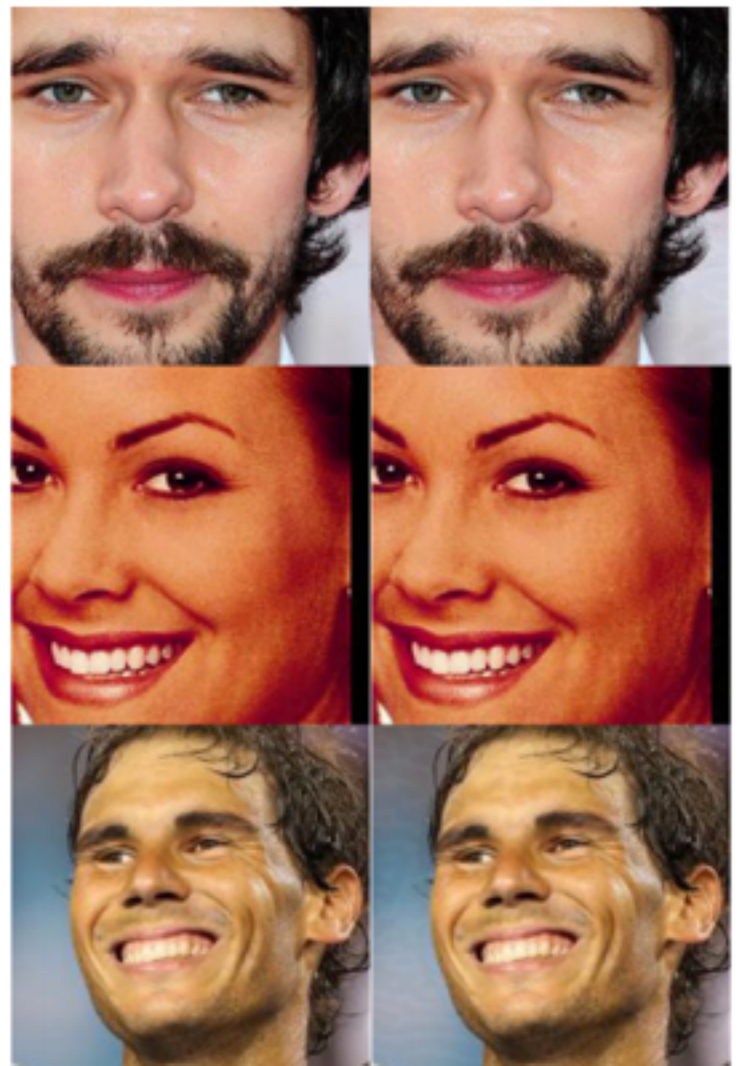
***Abstract*—No.**

I. ATTACKING "ATTACKS MEET INTERPRETABILITY"

AmI (Attacks meet Interpretability) is an "attribute-steered" defense [3] to detect [1] adversarial examples [2] on face-recognition models. By applying interpretability techniques to a pre-trained neural network, AmI identifies "important" neurons. It then creates a second augmented neural network with the same parameters but increases the weight activations of important neurons. AmI rejects inputs where the original and augmented neural network disagree.

We find that this defense (presented at at NeurIPS 2018 as a spotlight paper—the top 3% of submissions) is completely ineffective, and even *defense-oblivious*[1] attacks reduce the detection rate to 0% on untargeted attacks. That is, AmI is no more robust to untargeted attacks than the undefended original network. Figure 1 contains examples of adversarial examples that fool the AmI defense. We are incredibly grateful to the authors for releasing their source code[2] which we build on[3]. We hope that future work will continue to release source code by publication time to accelerate progress in this field.

A. Evaluation



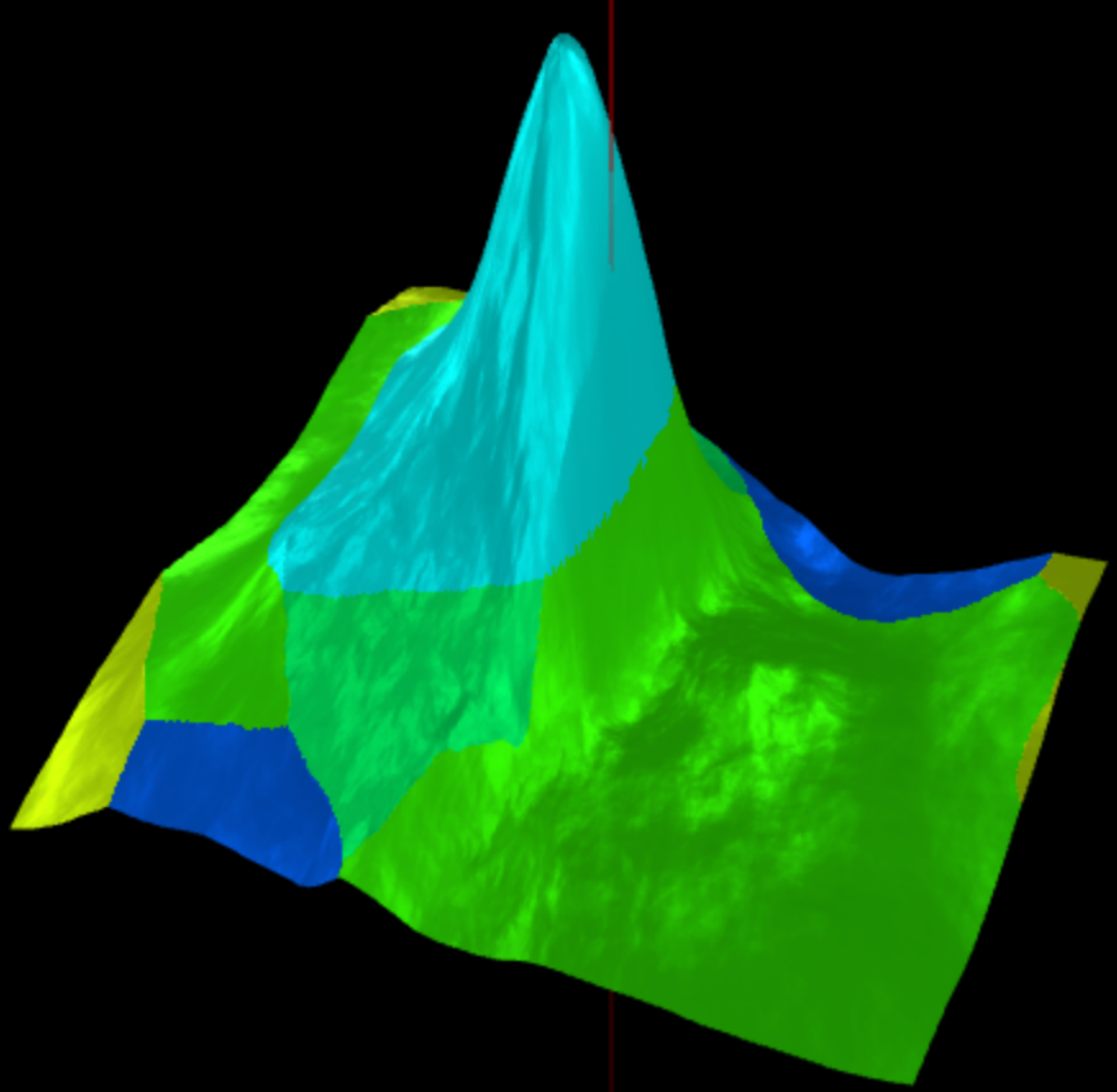
Lessons Learned from
Evaluating the Robustness of
Defenses to Adversarial Examples

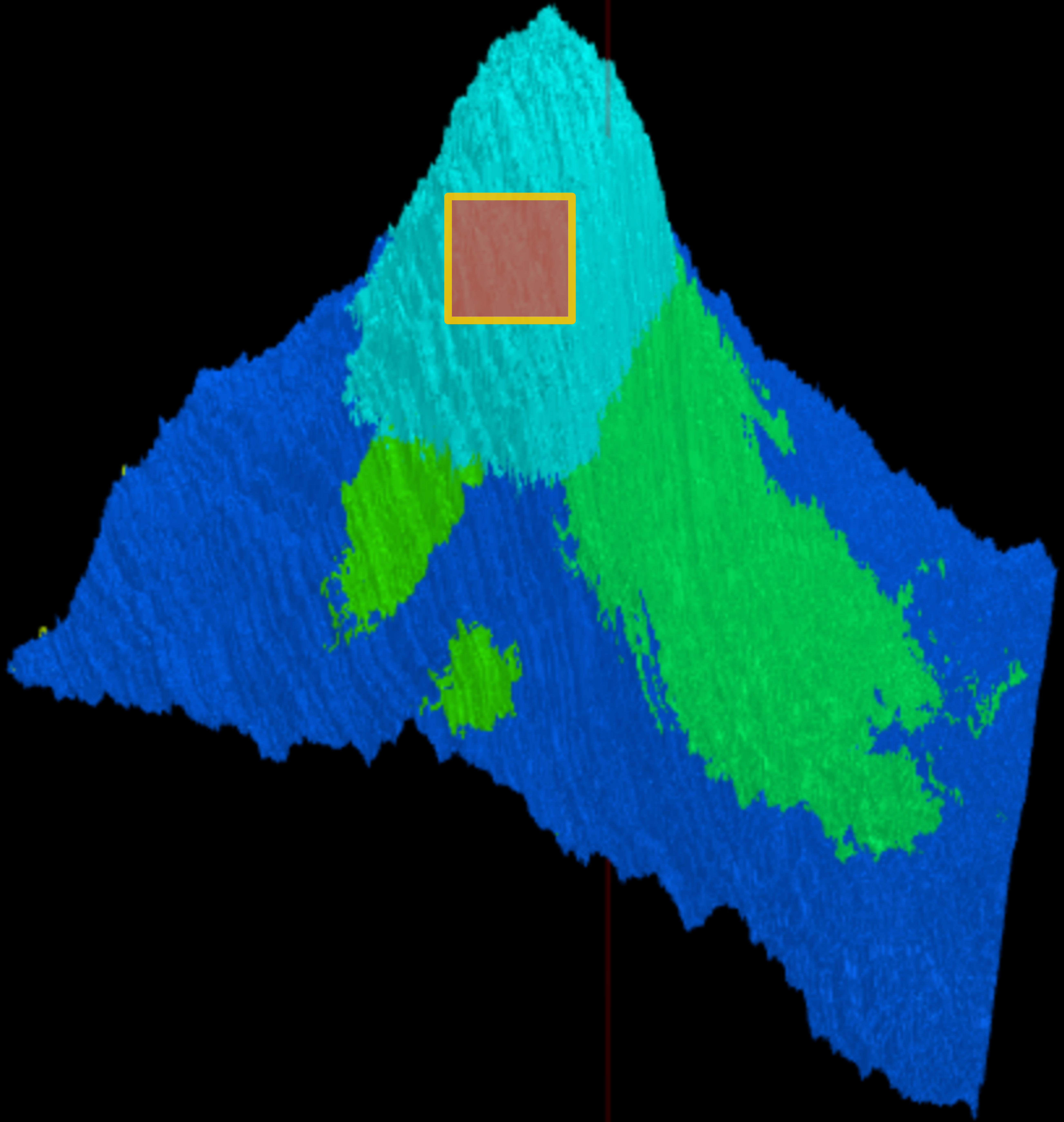
Lessons (1 of 2)

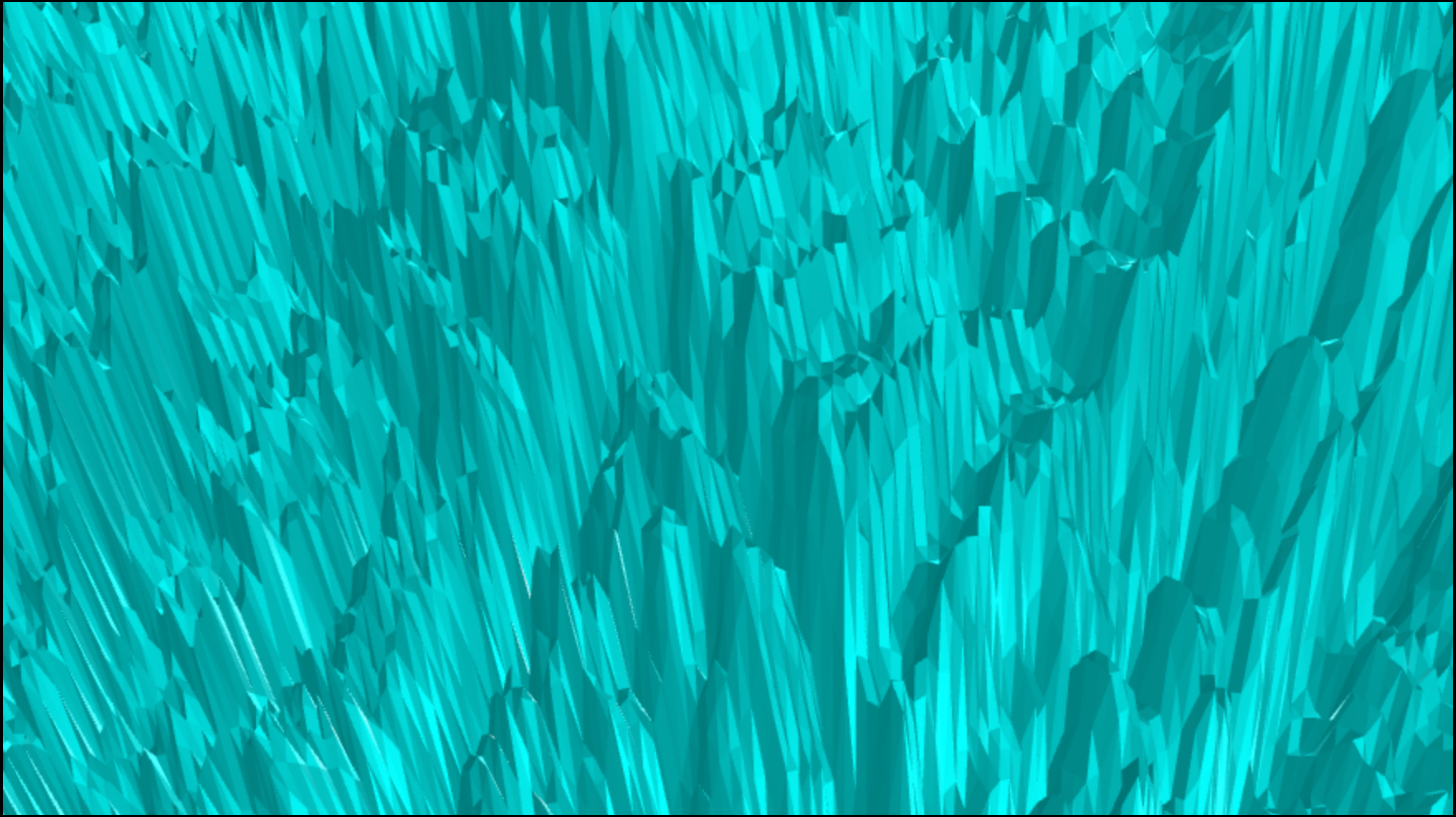
what we learn from evaluations
(and why to evaluate thoroughly)

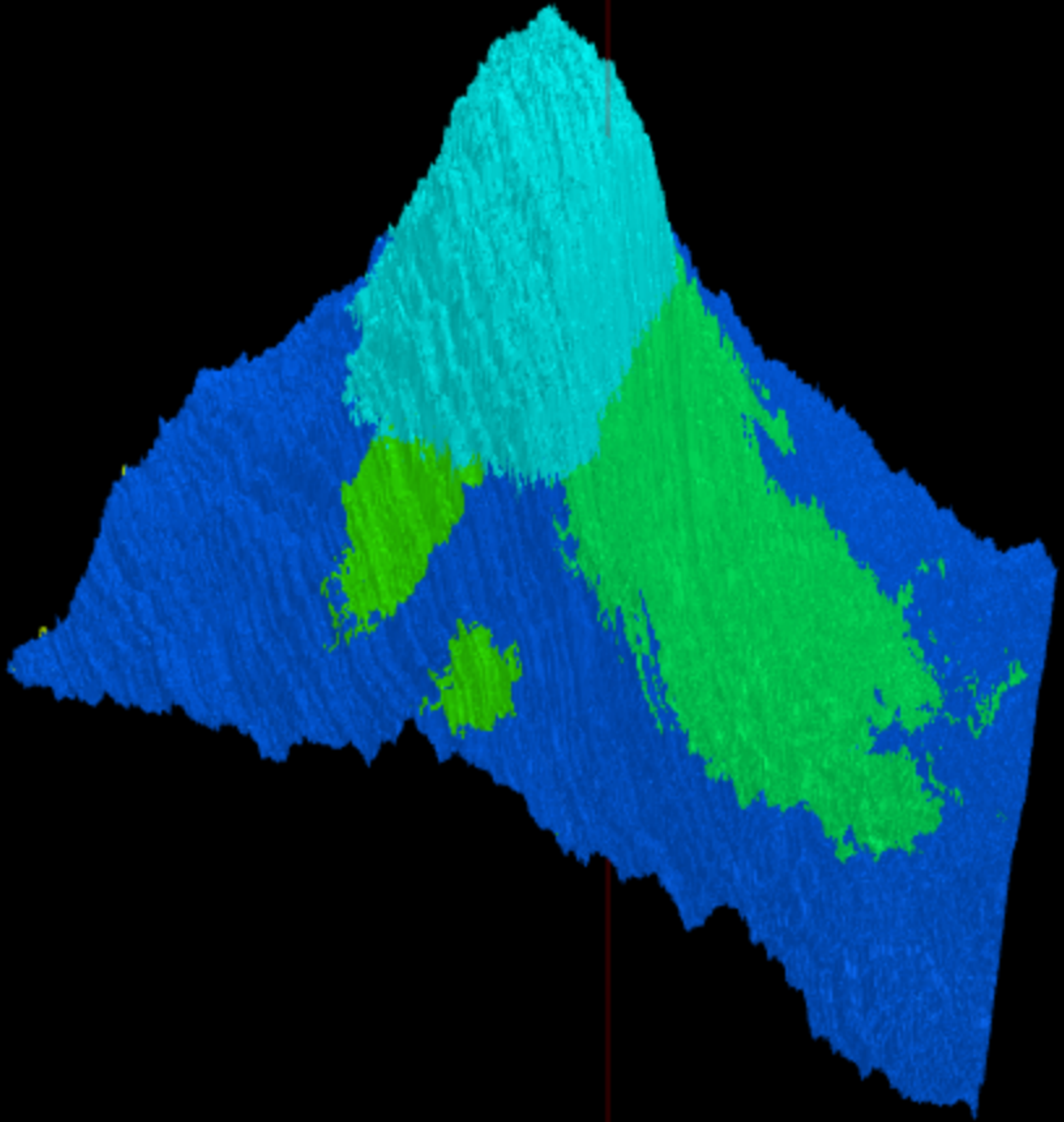
A Brief History of ~~Time~~ Defenses

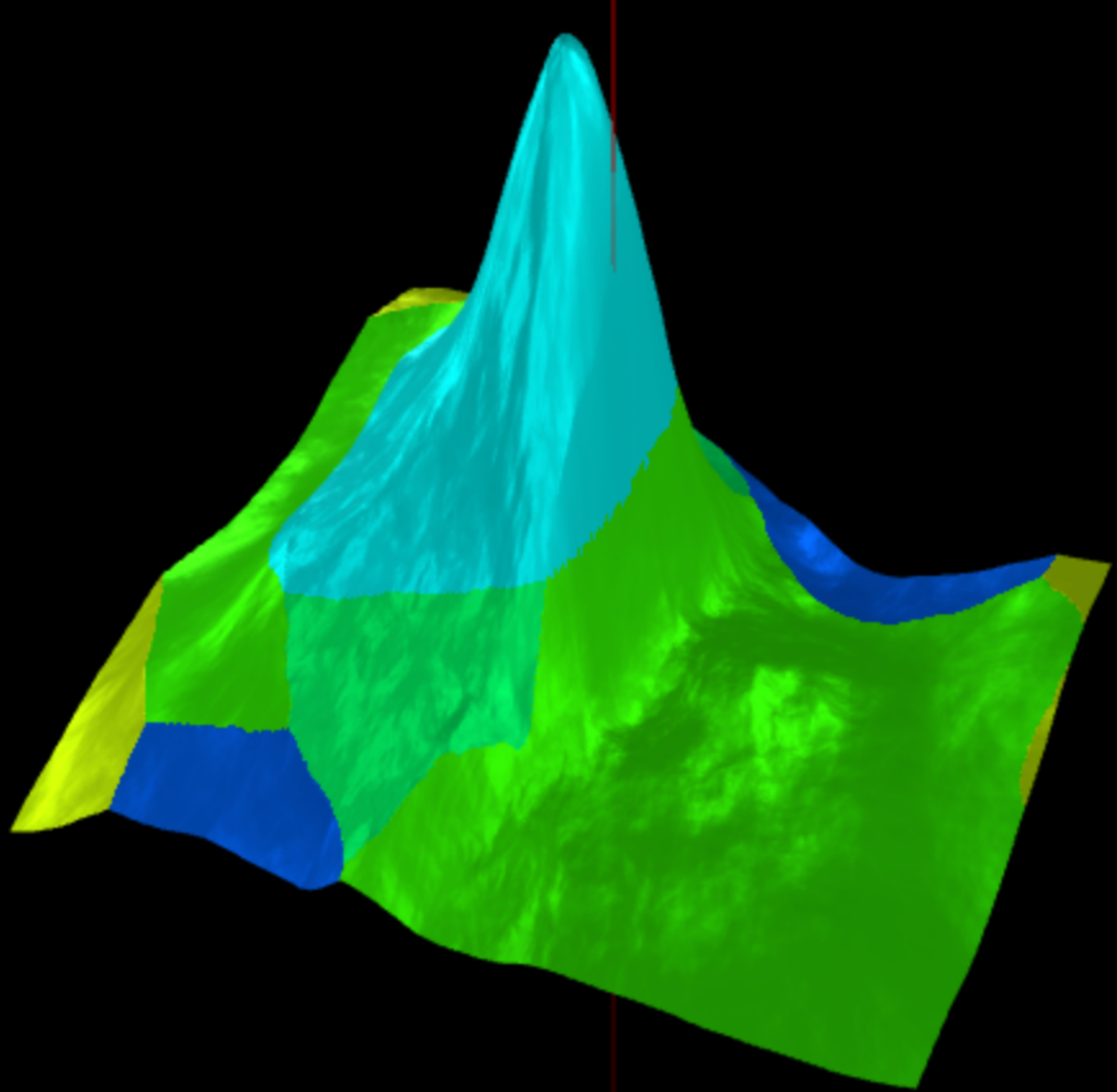
- S&P'16 - *gradient masking*
- ICLR'17 - *attack objective functions*
- CCS'17 - *transferability of examples*
- ICLR'18 - *obfuscated gradients*



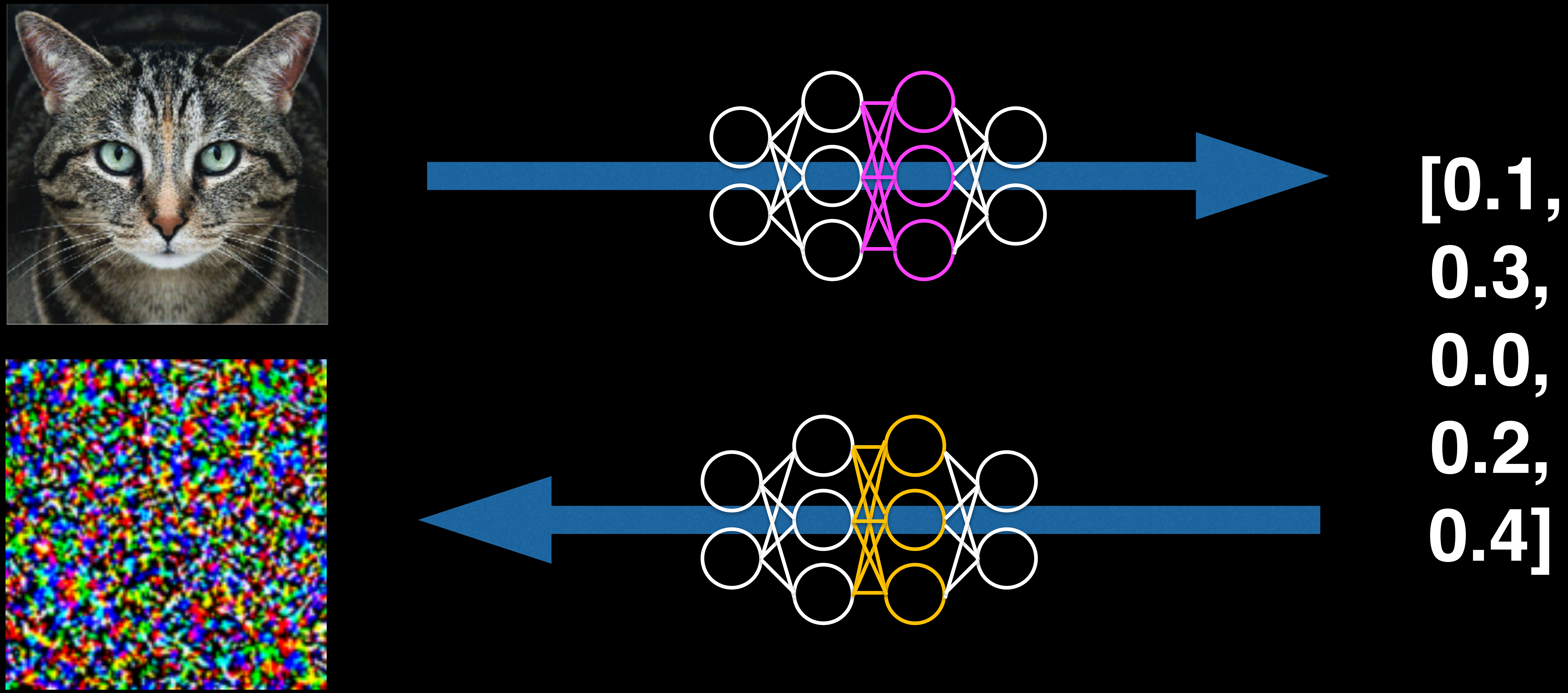








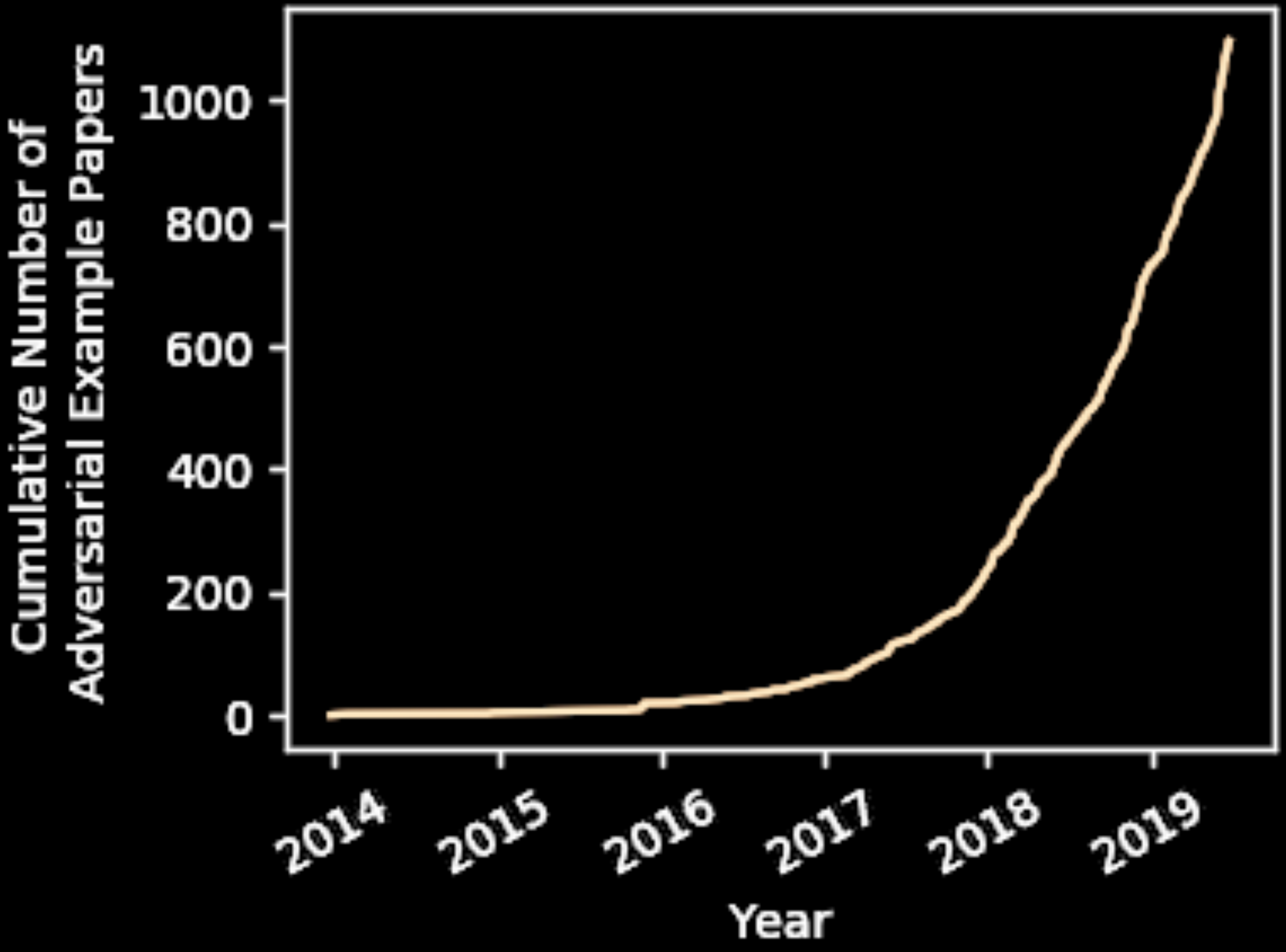
"Fixing" Gradient Descent



Disentangling
true robustness
from
apparent robustness
is nontrivial

Lessons (2 of 2)

performing better evaluations


Cumulative Number of
Adversarial Example Papers
1000
800
600
400
200
0
2014
2015
2016
2017
2018
2019
Year

On Evaluating Adversarial Robustness

Nicholas Carlini[1], Anish Athalye[2], Nicolas Papernot[1], Wieland Brendel[3], Jonas Rauber[3], Dimitris Tsipras[2], Ian Goodfellow[1], Aleksander Mądry[2], Alexey Kurakin[1] *

[1] Google Brain [2] MIT [3] University of Tübingen

Actionable advice requires specific, concrete examples

Everything the following papers do is standard practice the adversary has access to those networks (but does not have access to the input transformations applied at test time).

[2] The white-box attacks defined in this paper should be called oblivious attacks according to Carlini and Wagner's definition [3]

an adversary gains access to all parameters and weights of a model that is trained on benign images, but is unaware of the defense strategy.

Perform an *adaptive attack*

3.1. Effectiveness

Adversarial Attacks. We test on the following attacks:

3.1. Effectiveness

we trained on and L_{CW} is an objective encouraging misclassification. Under this threat model, *NeuralFP* achieves an AUC-ROC of **98.79%** against Adaptive-CW-L_2, with $N = 30$ and $\epsilon = 0.006$ for a set of unseen test-samples (1024 *pre-test*) and the corresponding adversarial examples. In contrast to other defenses that are vulnerable to Adaptive-CW-L_2 (Carlini & Wagner, 2017a), we find that *NeuralFP* is robust even under this whitebox-attack threat model.

3.4. Robustness to Adaptive Whitebox-Attackers

We further considered an adaptive attacker that has knowledge of the predetermined fingerprints and model weights, similar to (Carlini & Wagner, 2017a). Here, the adaptive attacker (Adaptive-CW-L_2) tries to find an adversarial example x' that also minimizes the fingerprint-loss, attacking a CIFAR-10 model trained with *NeuralFP*. To this end, the CW-L_2 objective is modified as:

$$\min_{x'} \|x - x'\|_2 + \gamma \left(L_{CW}(x') + L_{fp}(x', y^*, \xi; \theta)\right) \quad (29)$$

Here, y^* is the label-vector, $\gamma \in [10^{-3}, 10^6]$ is a scalar found through a bisection search, L_{fp} is the fingerprint-loss

4. Related Work

5. Discussion and Future Work


3.1. Effective
Adversarial Attacks. We test on the following attacks:
3.4. Robustness to Adaptive Whitebox-Attackers
4. Related Work
5. Discussion and Future Work
3.4. Robustness to Adaptive Whitebox-Attackers

We now evaluate on two held out L_0 attacks

A "hold out" set is
not an adaptive attack

To create adversarial examples in our evaluation, we use FGSM,

For the next series of experiments, we test
against the *Fast Gradient Sign Method*

In our experiment, we use the Fast Gradient Sign Method (FGSM)

TABLE 4: Performance of detecting FGSM adversarial
examples with different scalar quantization schemes.

Stop using FGSM
(exclusively)

- Number of attack steps: 10

experiments on CIFAR used $\varepsilon = 0.031$ and 7 steps for iterative attacks;

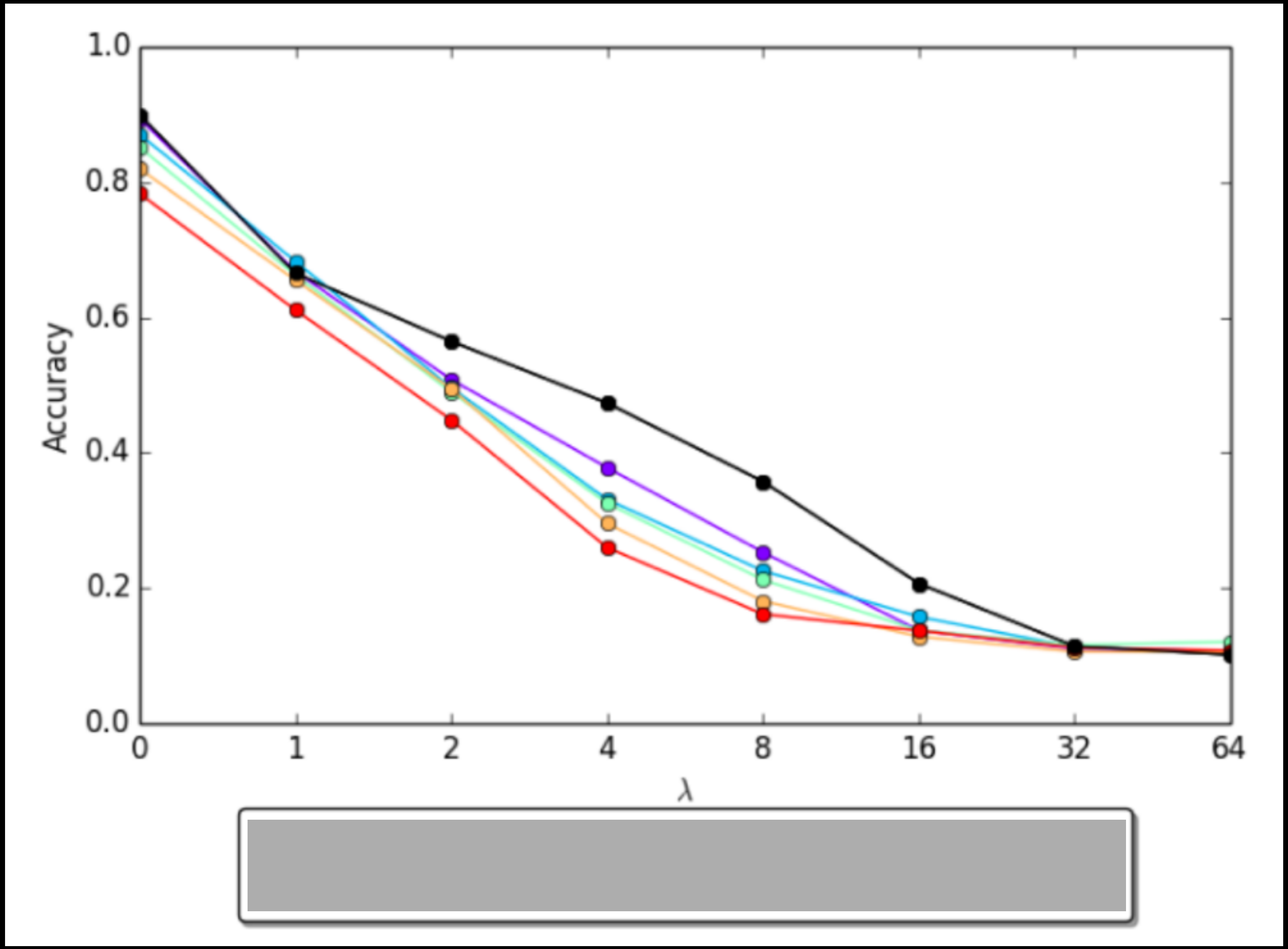
Use more than 100 (or 1000?) iteration of gradient descent

	Model	FGSM	PGD
Clean		25.10	4.10
		46.15	1.66
		43.89	3.57
		52.07	53.11
		48.50	50.50

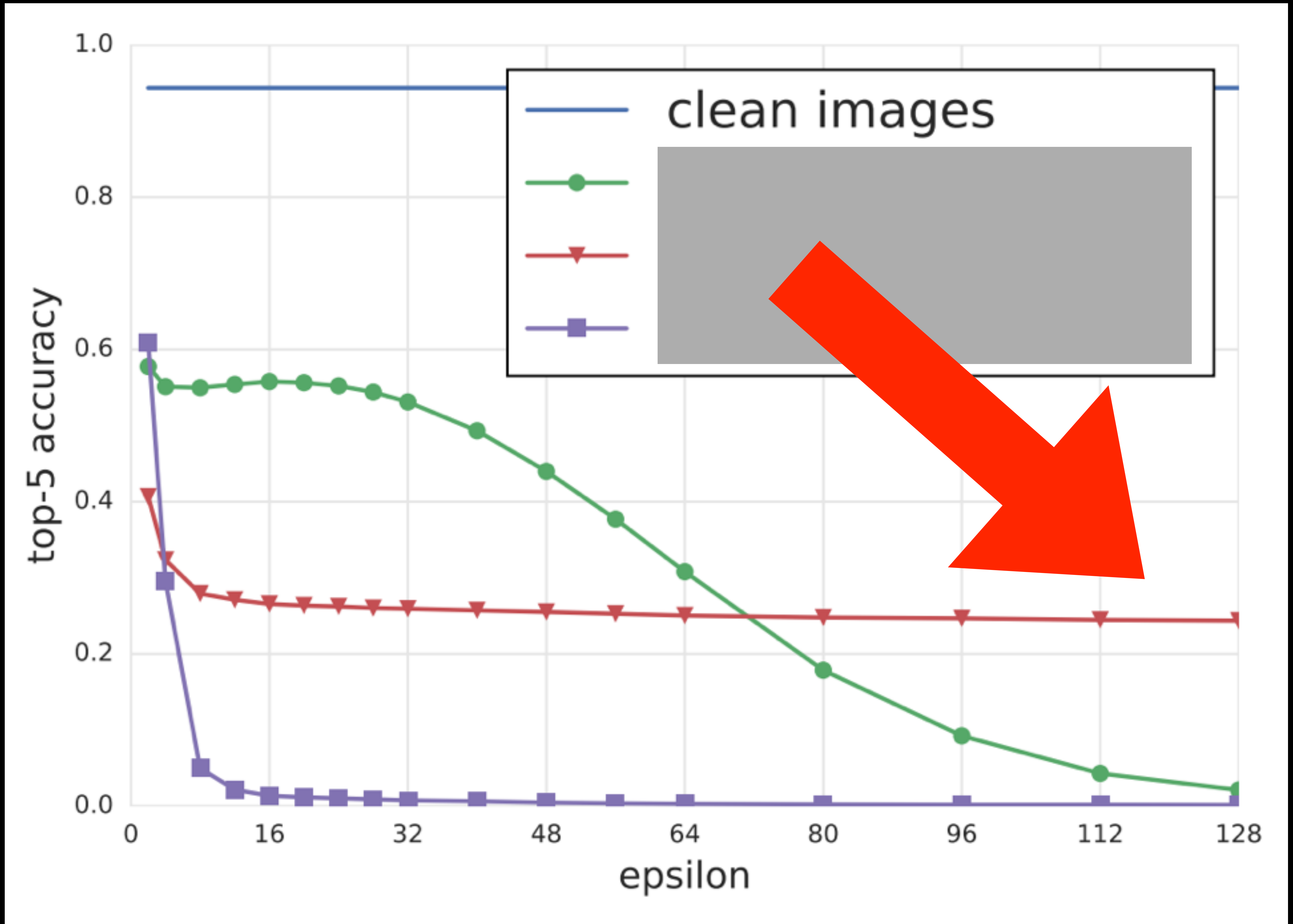
Iterative attacks should always do better than single step attacks.

Attack	Parameter	Fooling Rate	Detection Rate
DeepFool		99.35%	97.83%
Carlini	κ=0.0	100.0%	95.66%

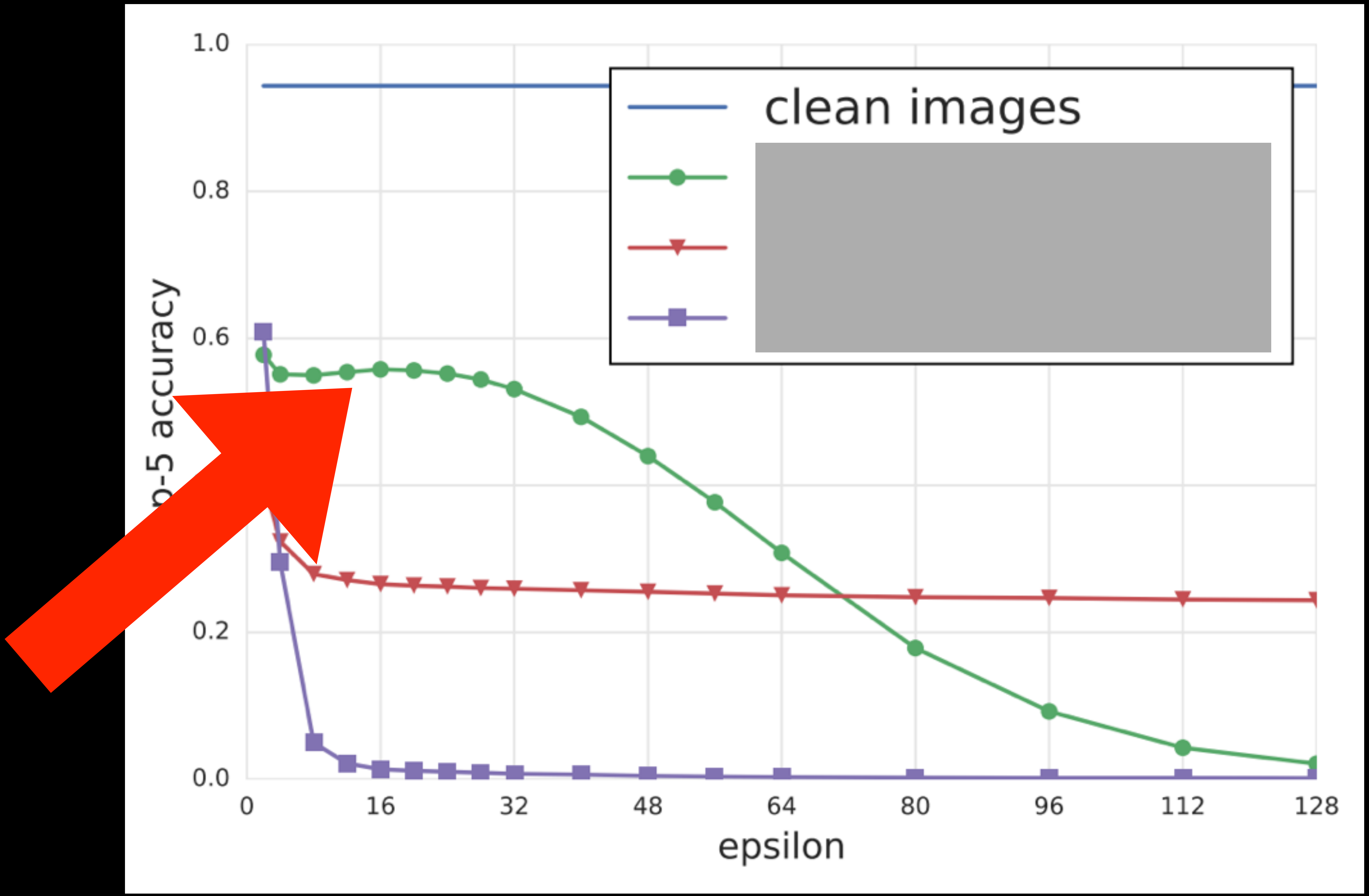
Unbounded optimization attacks should eventually reach in 0% accuracy



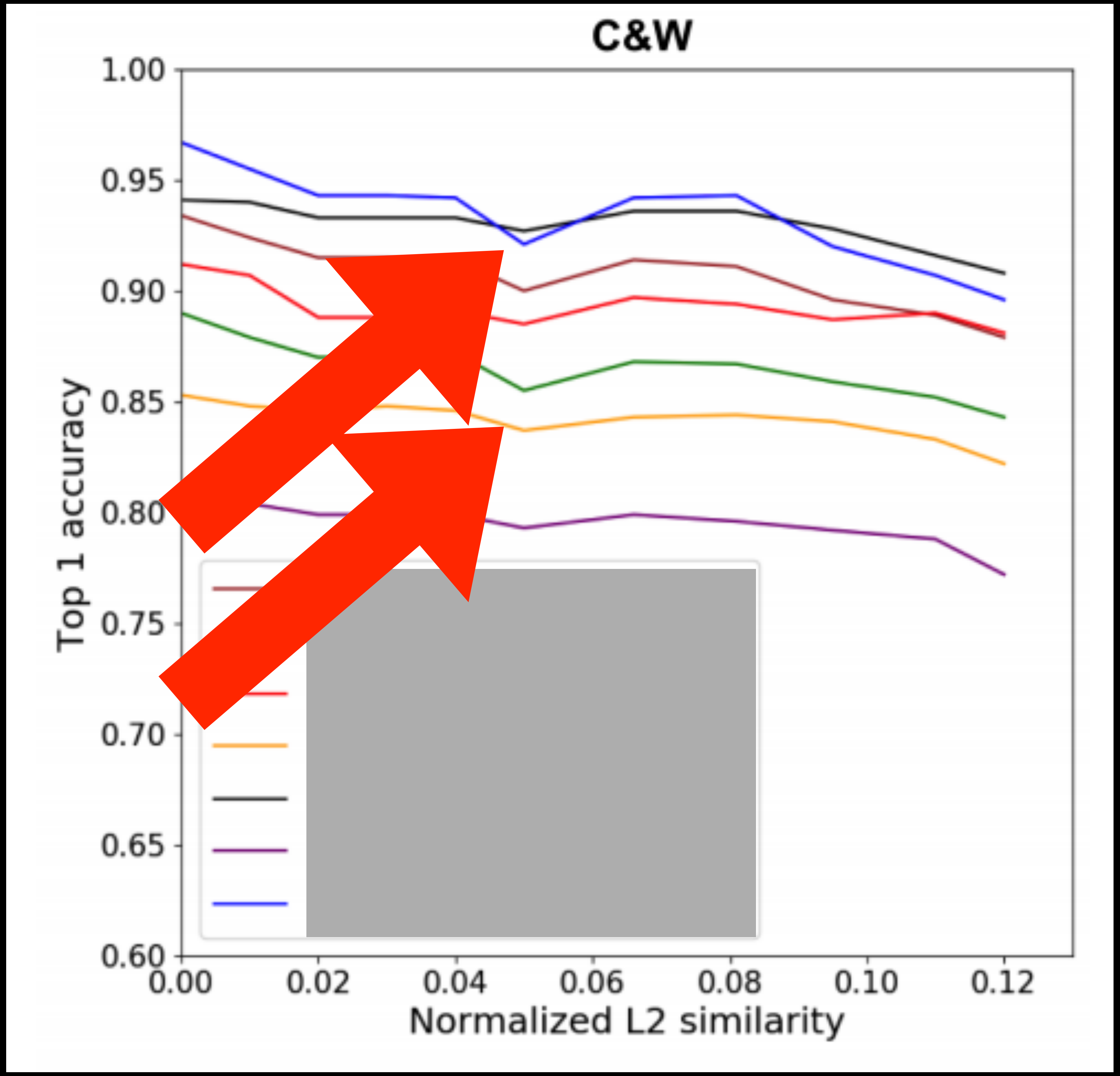
Unbounded optimization attacks should eventually reach in 0% accuracy



Unbounded optimization attacks should eventually reach in 0% accuracy



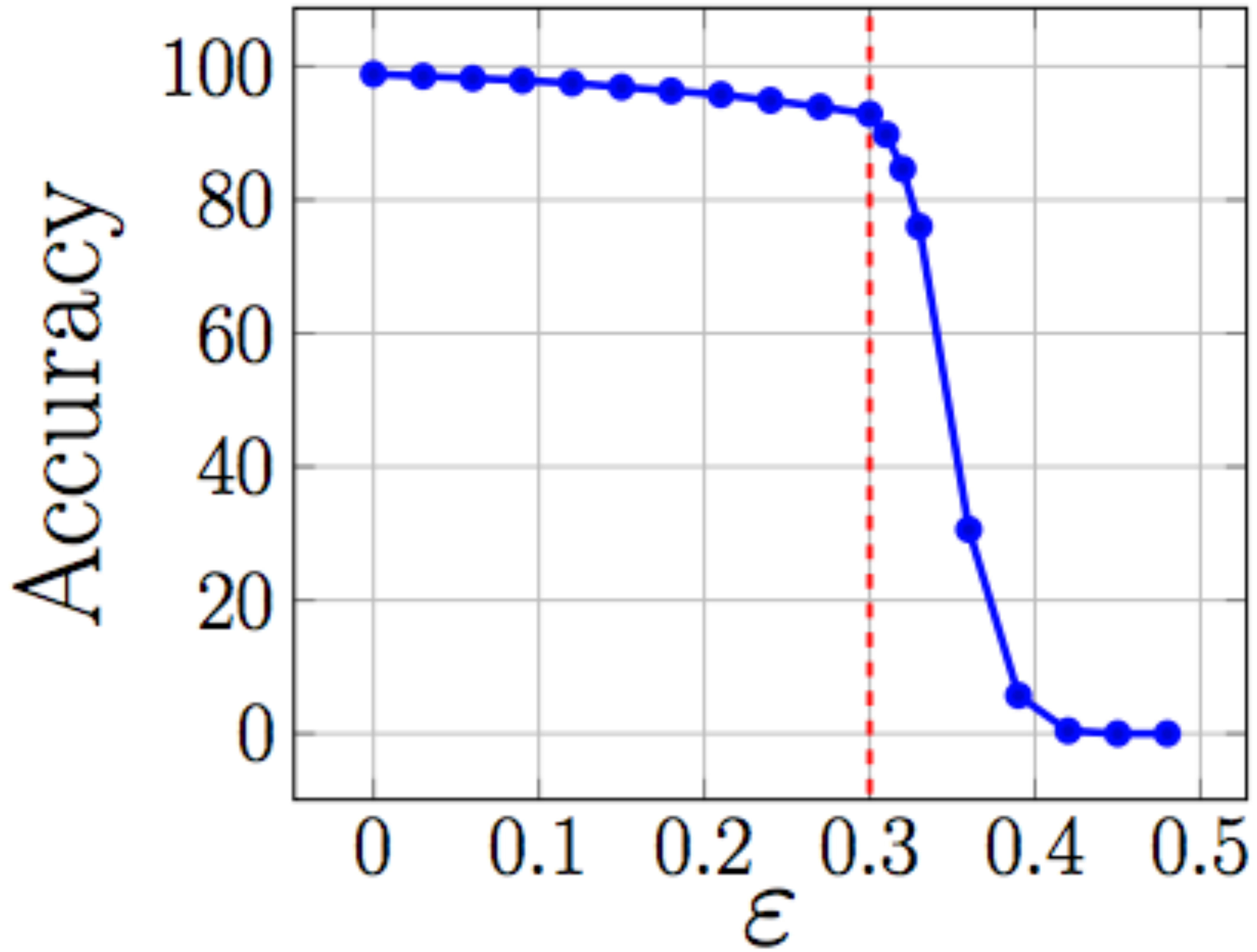
Model accuracy should be monotonically decreasing



Model accuracy should be monotonically decreasing

Model	clean	step_ll		step_FGSM		iter_FGSM		CW	
		ϵ=2	ϵ=16	ϵ=2	ϵ=16	ϵ=2	ϵ=4	ϵ=2	ϵ=4
R110$_K$	92.3	**88.3**	**90.7**	**86.0**	**95.2**	59.4	9.2	25	4
R110$_P$ (Ours)	92.3	86.0	89.4	81.6	91.6	64.1	20.9	32	7
R110$_E$	92.3	86.3	74.3	84.1	72.9	63.5	21.1	24	6
R110$_{K,C}$ (Ours)	92.3	86.2	72.8	82.6	66.7	69.3	33.4	20	5
R110$_{P,E}$ (Ours)	91.3	84.0	65.7	77.6	54.5	66.8	38.3	**38**	**16**
R110$_{P,C}$ (Ours)	91.5	85.7	76.4	82.4	69.1	**73.5**	**42.5**	27	15

Evaluate against the worst attack



(a) MNIST, ℓ_∞ norm

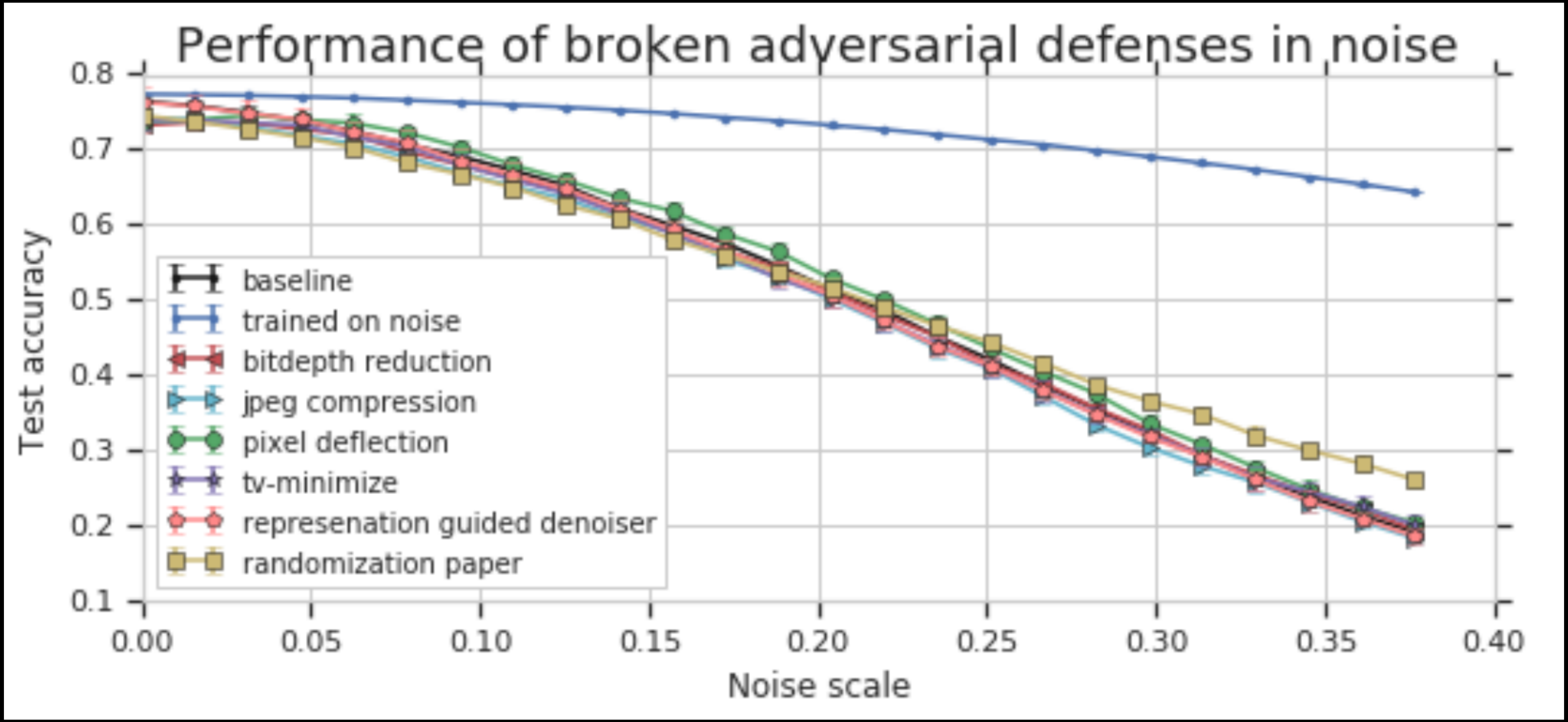
Plot accuracy vs distortion

MaxIter	Model1	Model2	Model3	Model4
Natural	99.1%	98.5%	98.7%	98.2%
100	70.2%	91.7%	77.6%	75.6%
1000	0.05%	51.5%	20.3%	24.4%
10K	0%	16.0%	20.1%	24.4%
100K	0%	9.8%	20.1%	24.4%
1M	0%	7.6%	20.1%	24.4%

Verify enough iterations of gradient descent

By using a gradient-free method, we are able to attack the end-to-end model, despite the lack of an analytic gradient.

Try gradient-free attack algorithms



Try random noise

Conclusion

Conclusion

To understand adversarial examples, repeatedly *attack* and *defend*, optimizing for lessons learned.

Questions?

nicholas@carlini.com https://nicholas.carlini.com