

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



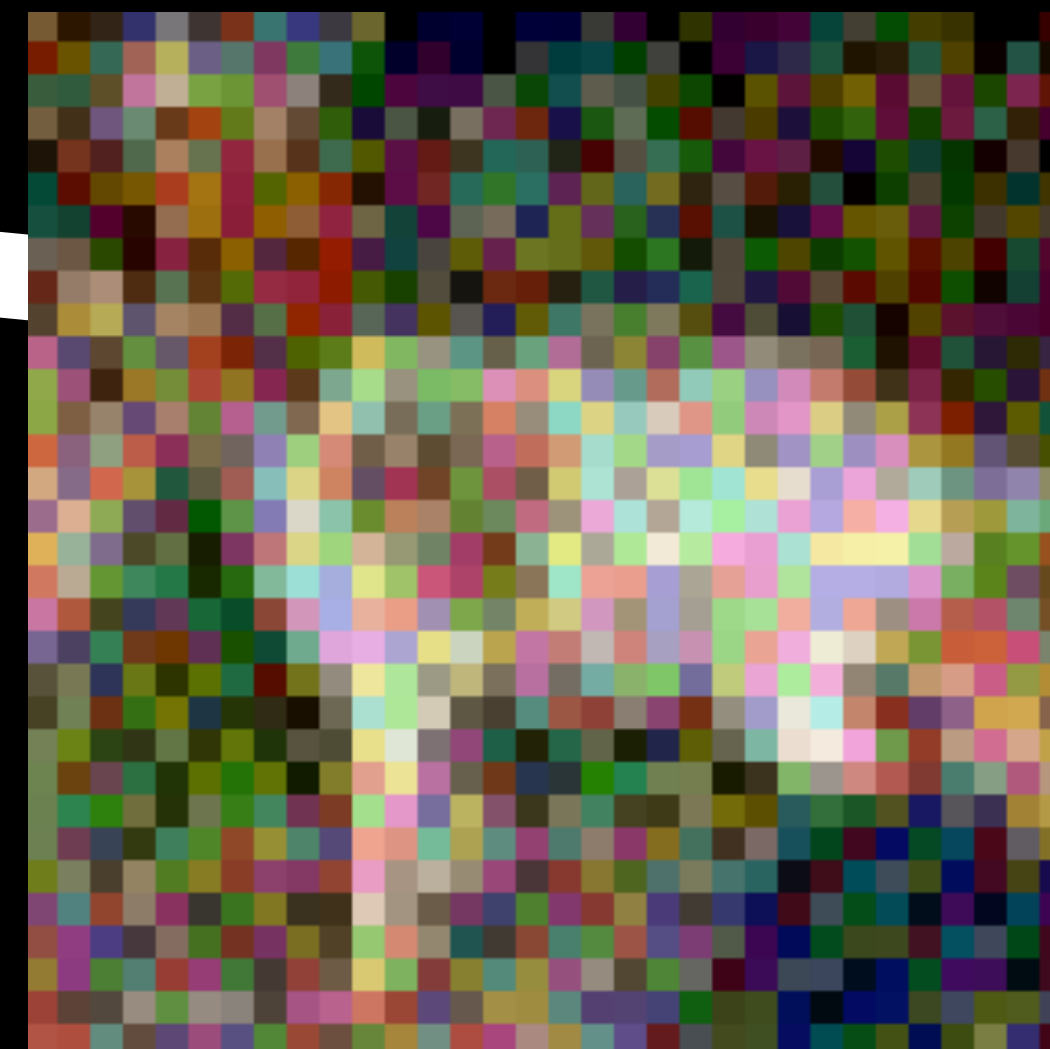
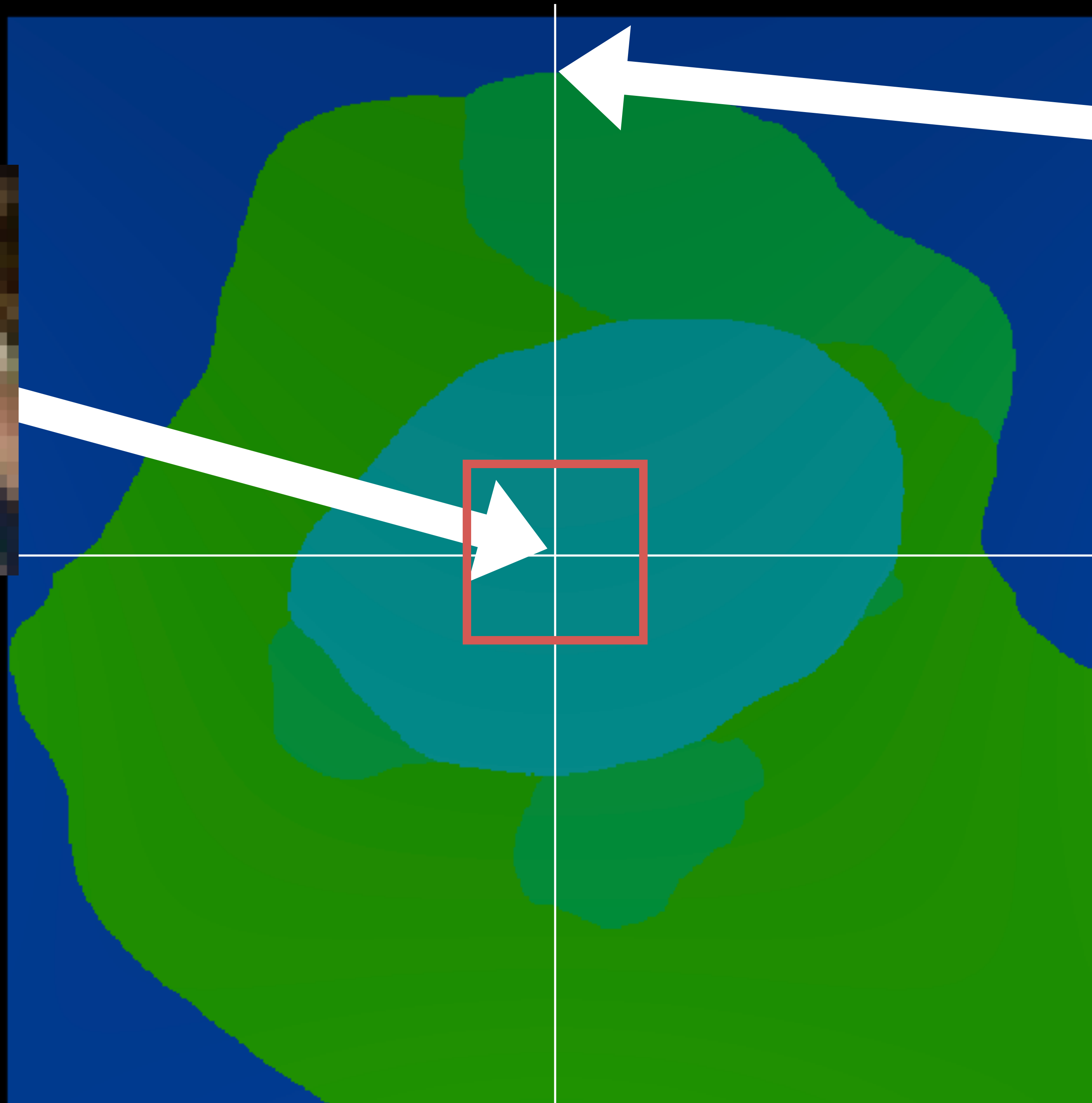
88% **tabby cat**

99% **guacamole**

How do we generate
adversarial examples?



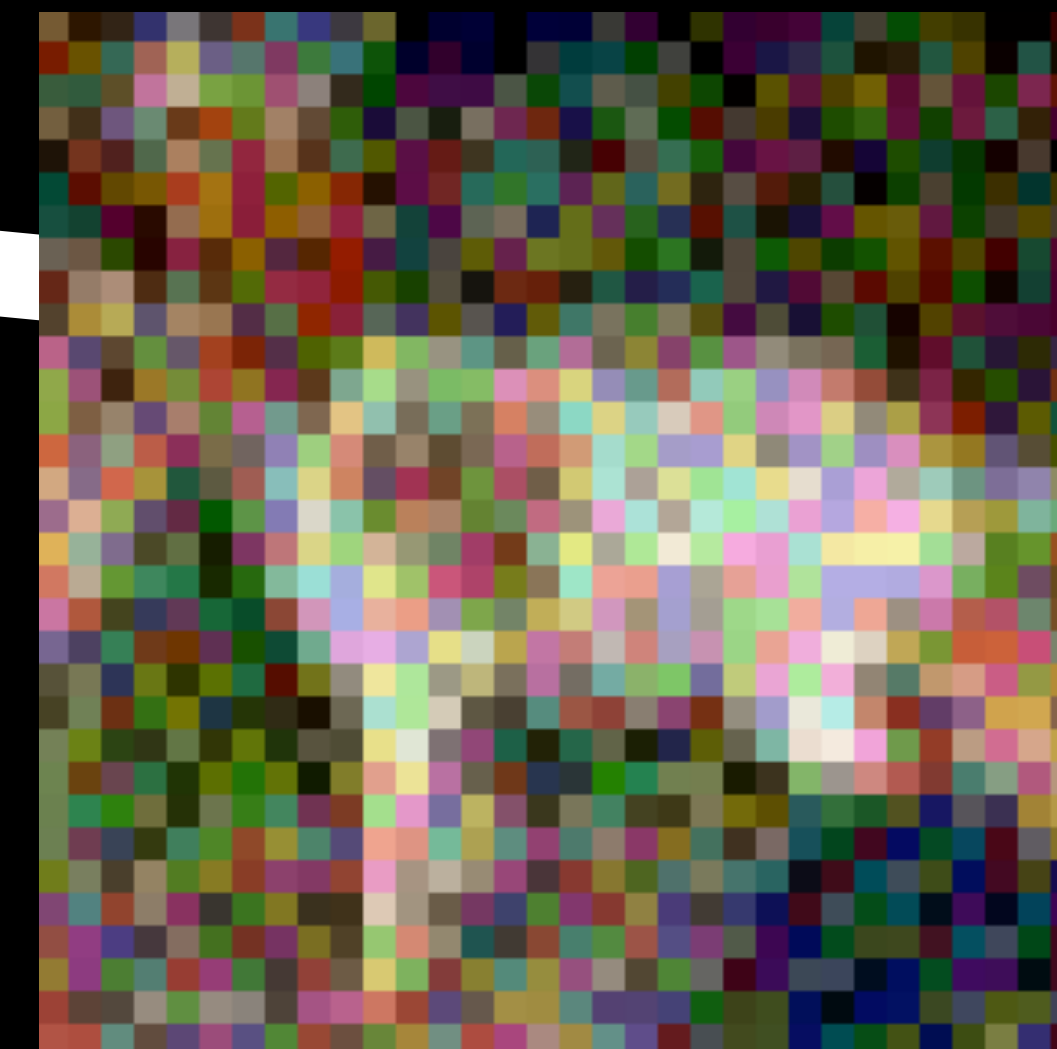
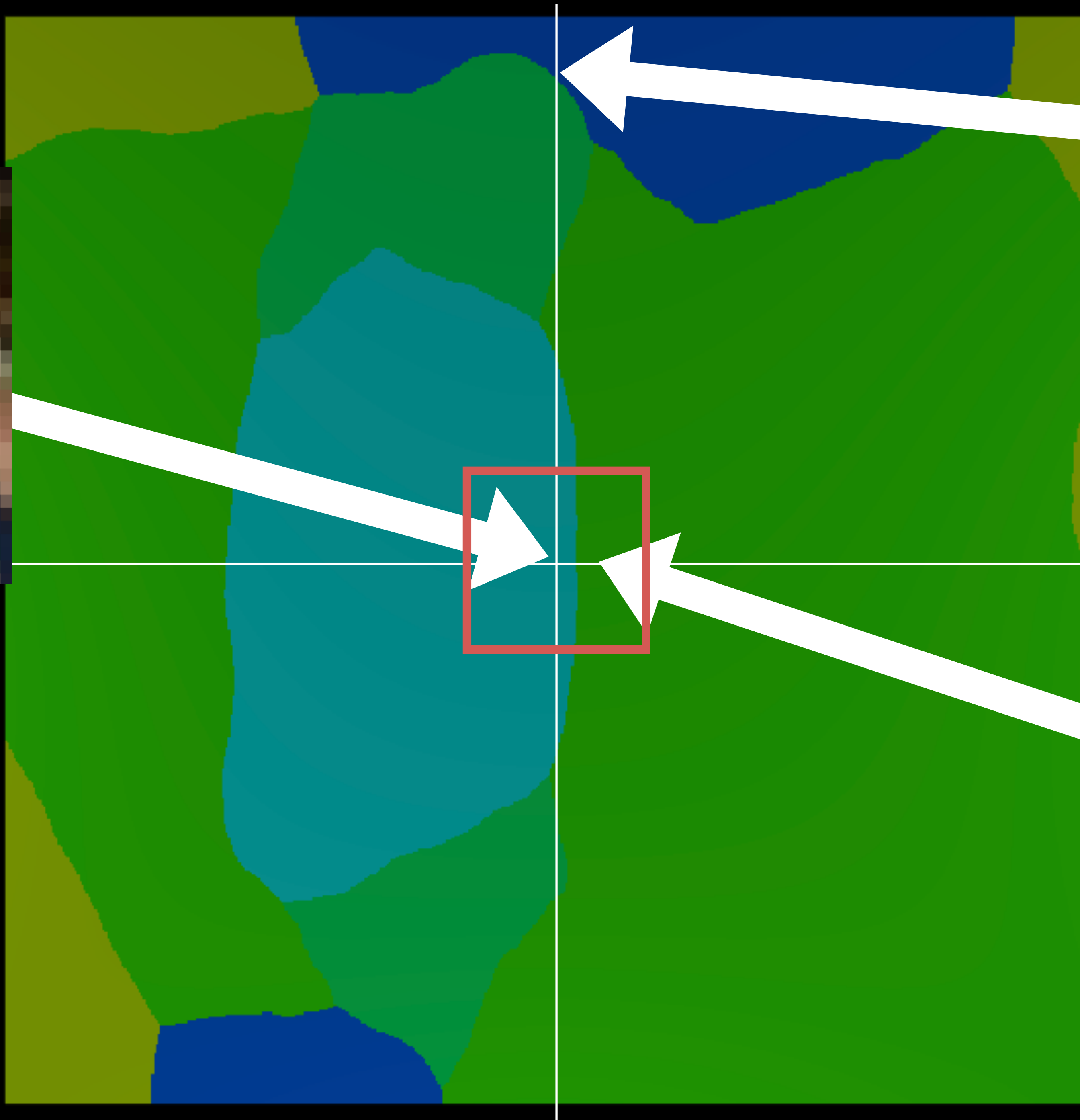
Dog



Truck



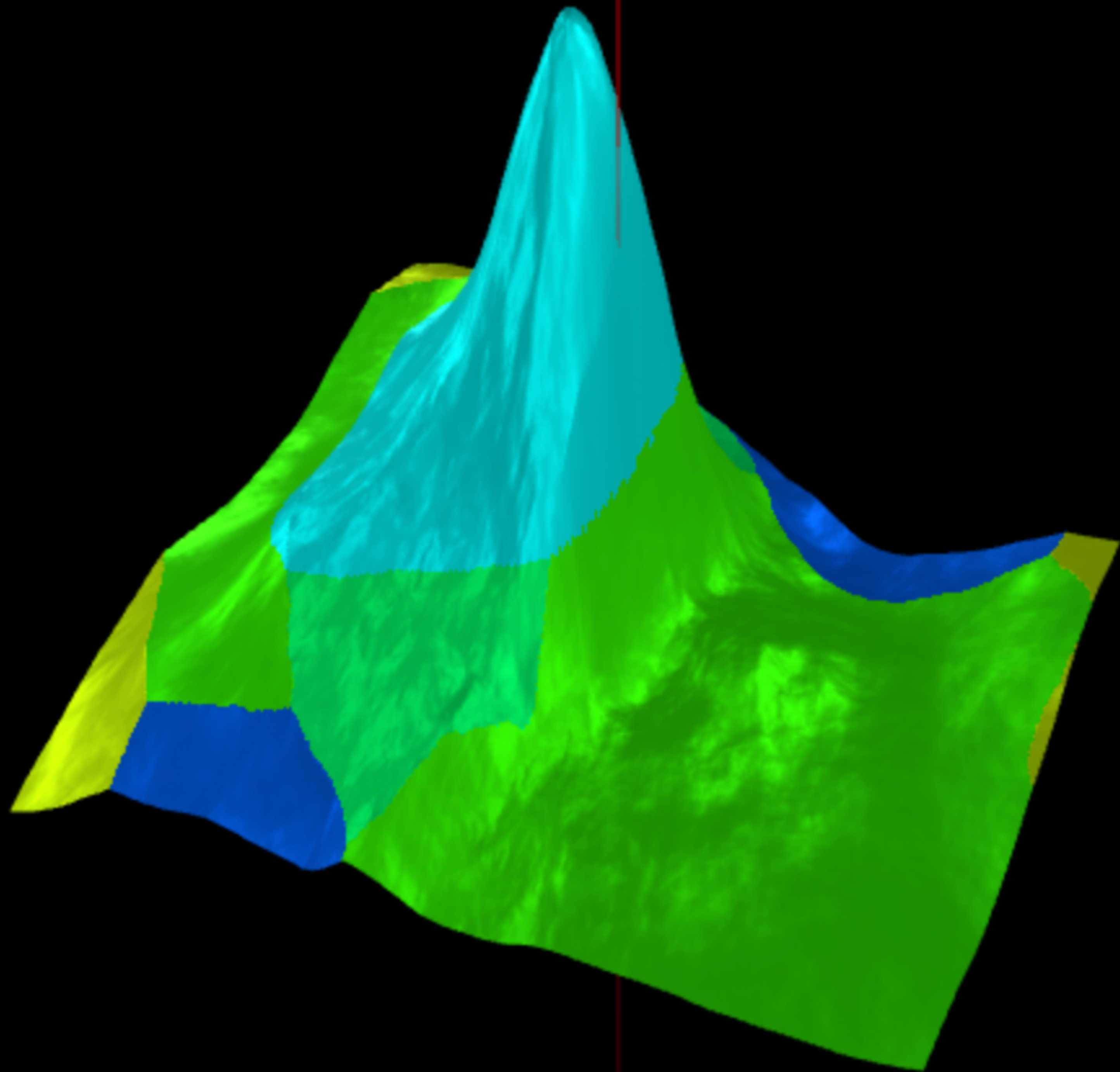
Dog

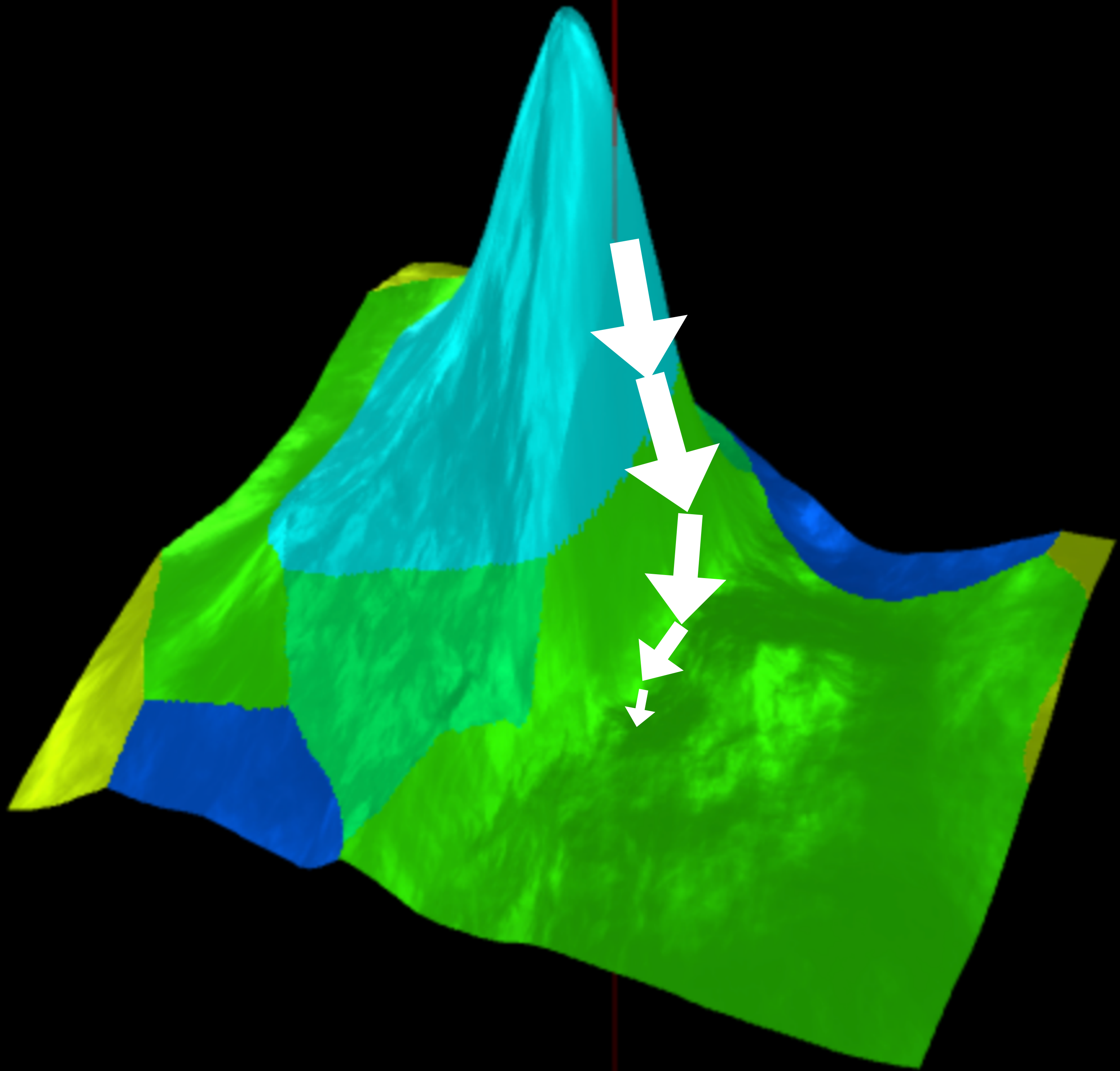


Truck



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?

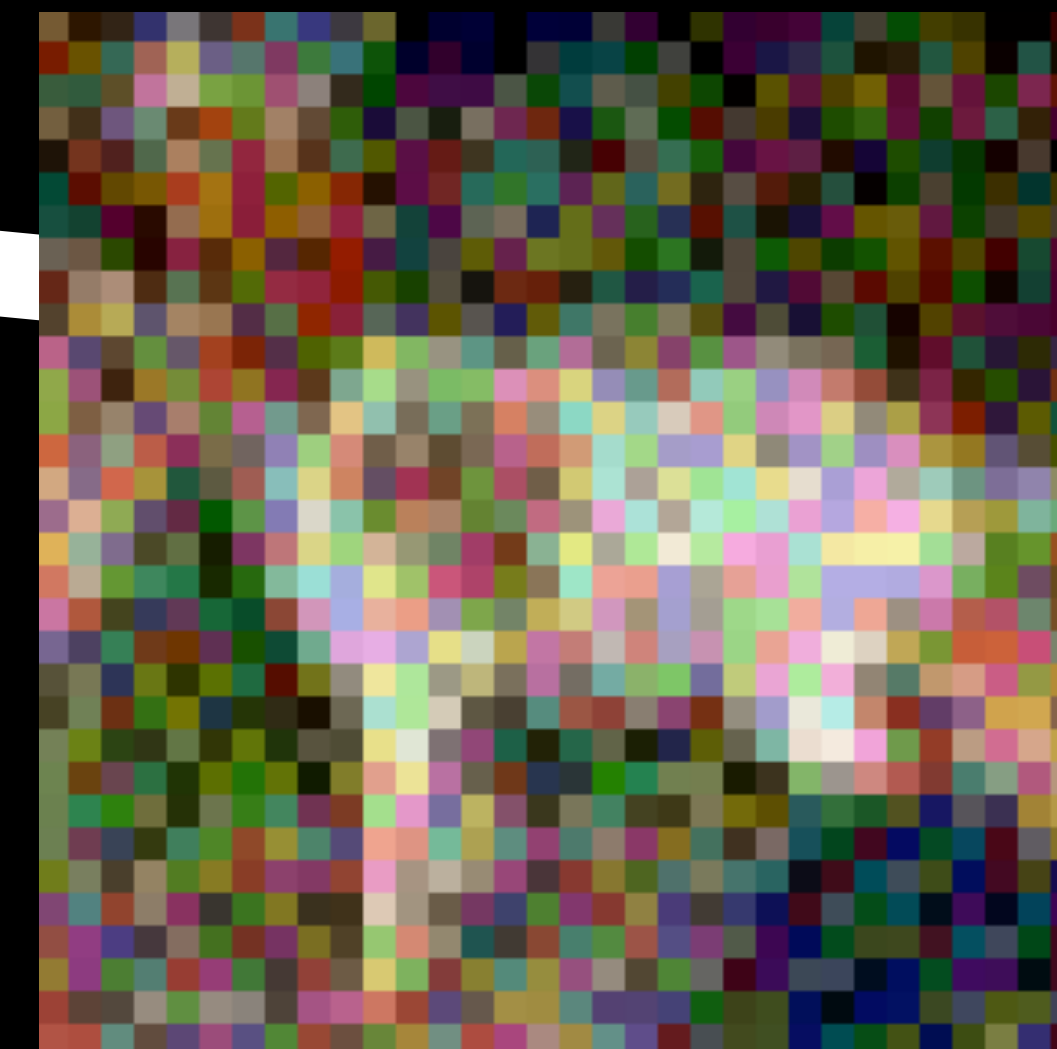
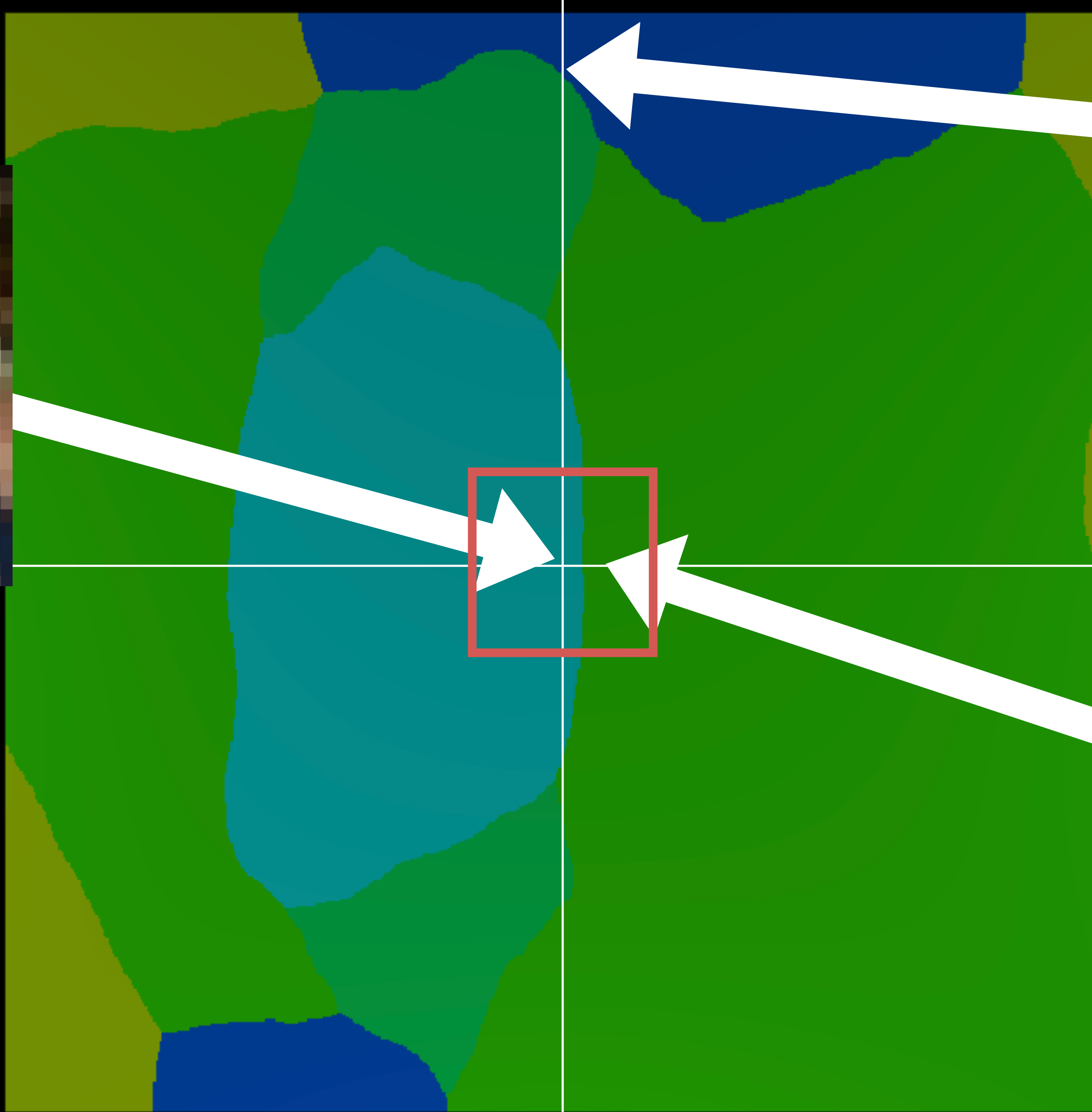
Good Threat Model:

"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, 7)

(8, 3)

Training

Adversarial Training (1)

(7, 7)

(8, 3)

(7, 7)

(8, 3)

Attack

Adversarial Training (2)

(7, 7)

(8, 3)

(7, 7)

(8, 3)

Training

Or:

Thermometer Encoding

Claim:

Neural networks are "overly linear"

Solution

$$T(0.13) = 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

$$T(0.66) = 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0$$

$$T(0.97) = 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1$$

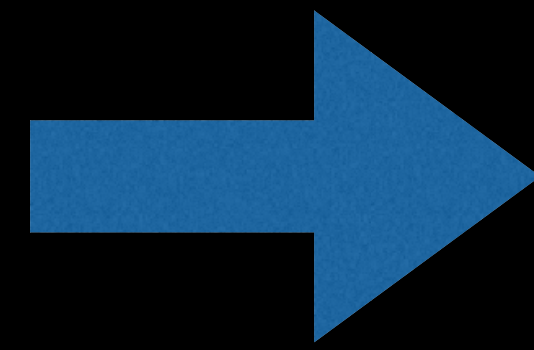
Or:

Input Transformations

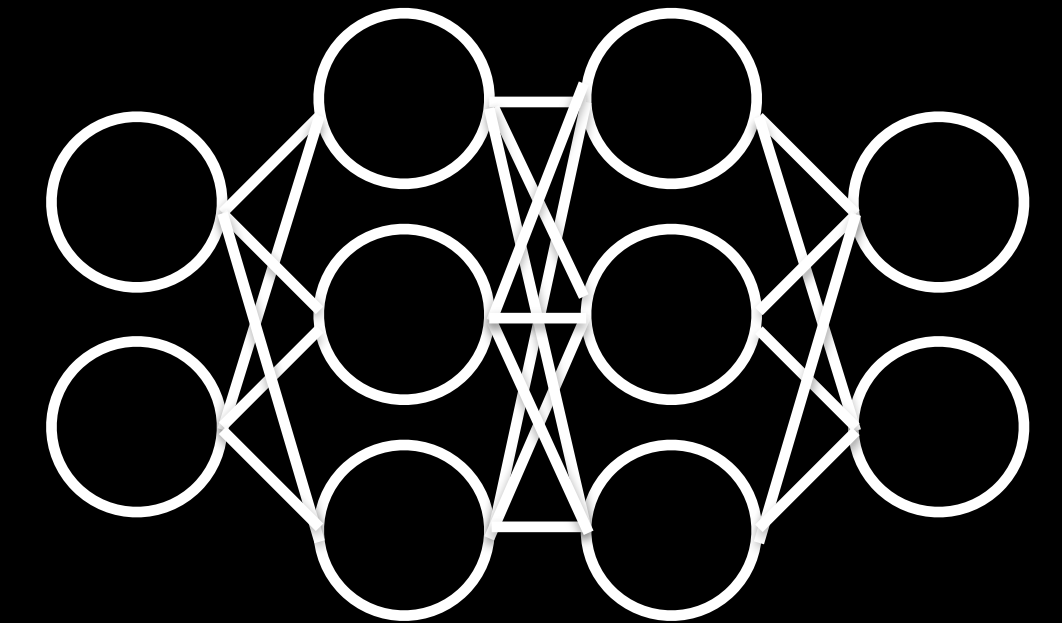
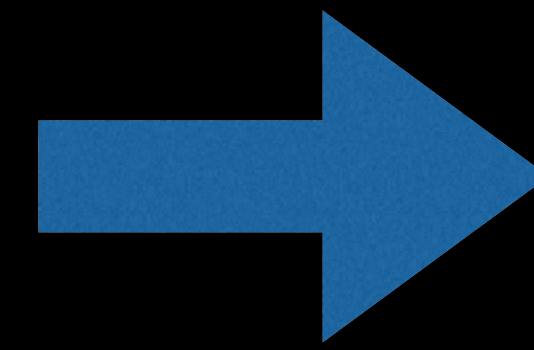
Claim:

Perturbations are brittle

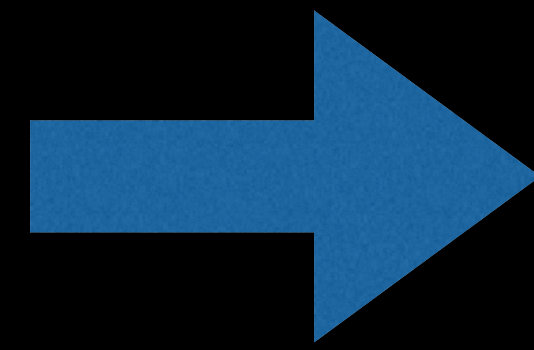
Solution



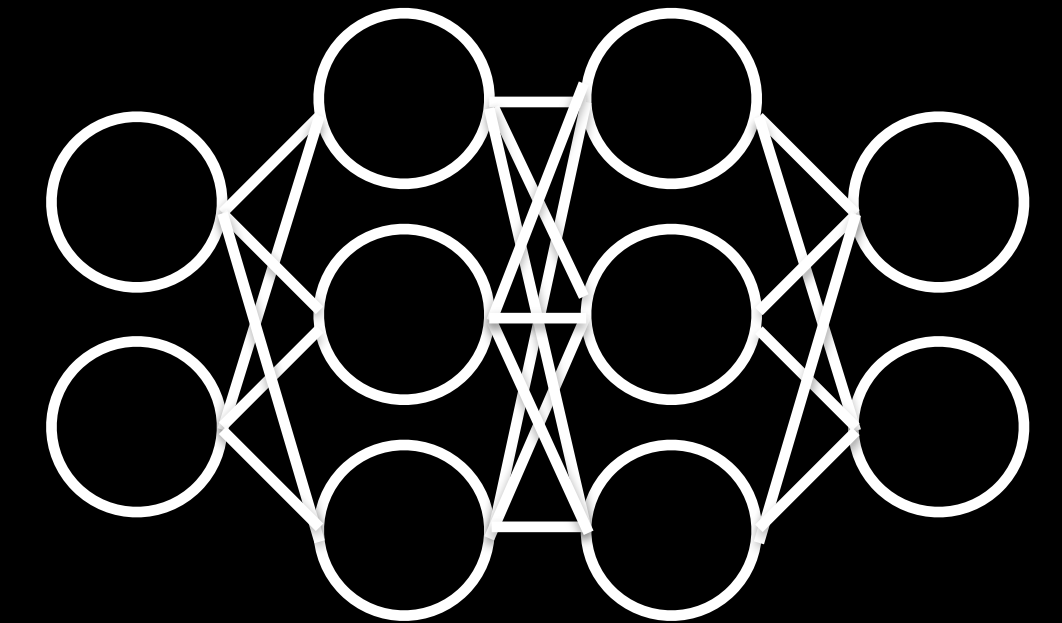
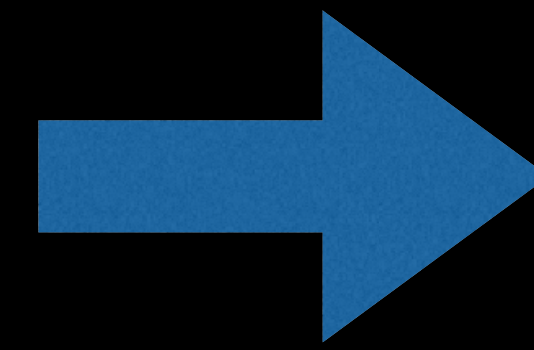
Random
Transform



Solution



JPEG
Compress



Lessons Learned from
Evaluating the Robustness of
Defenses to Adversarial Examples

What does it mean 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")
```

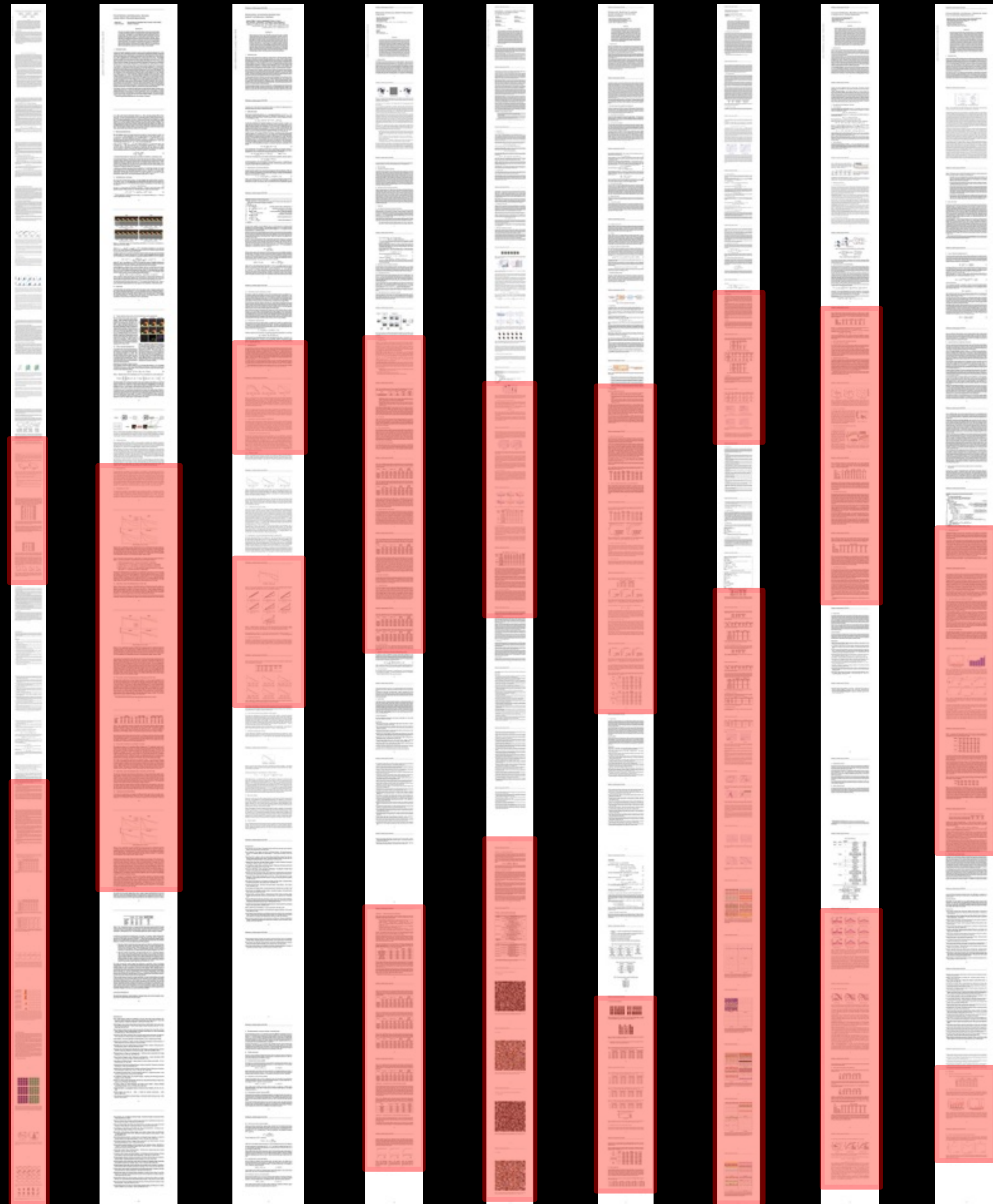
Adversarial ML Evaluations

```
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



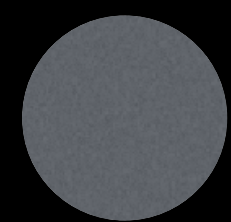
Serious effort
to evaluate

By space, most
papers are $\frac{1}{2}$
evaluation

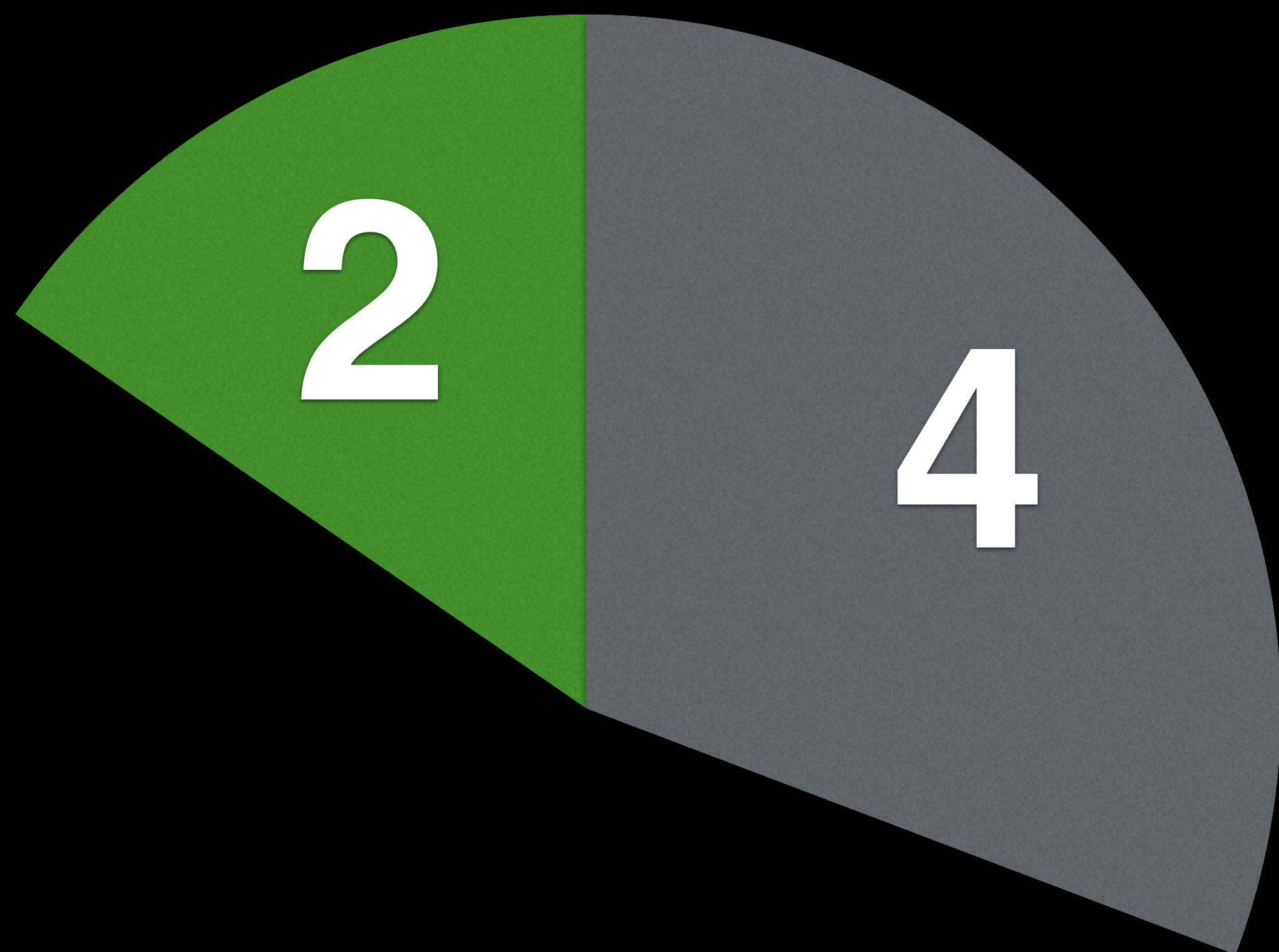
We re-evaluated
these defenses ...



4

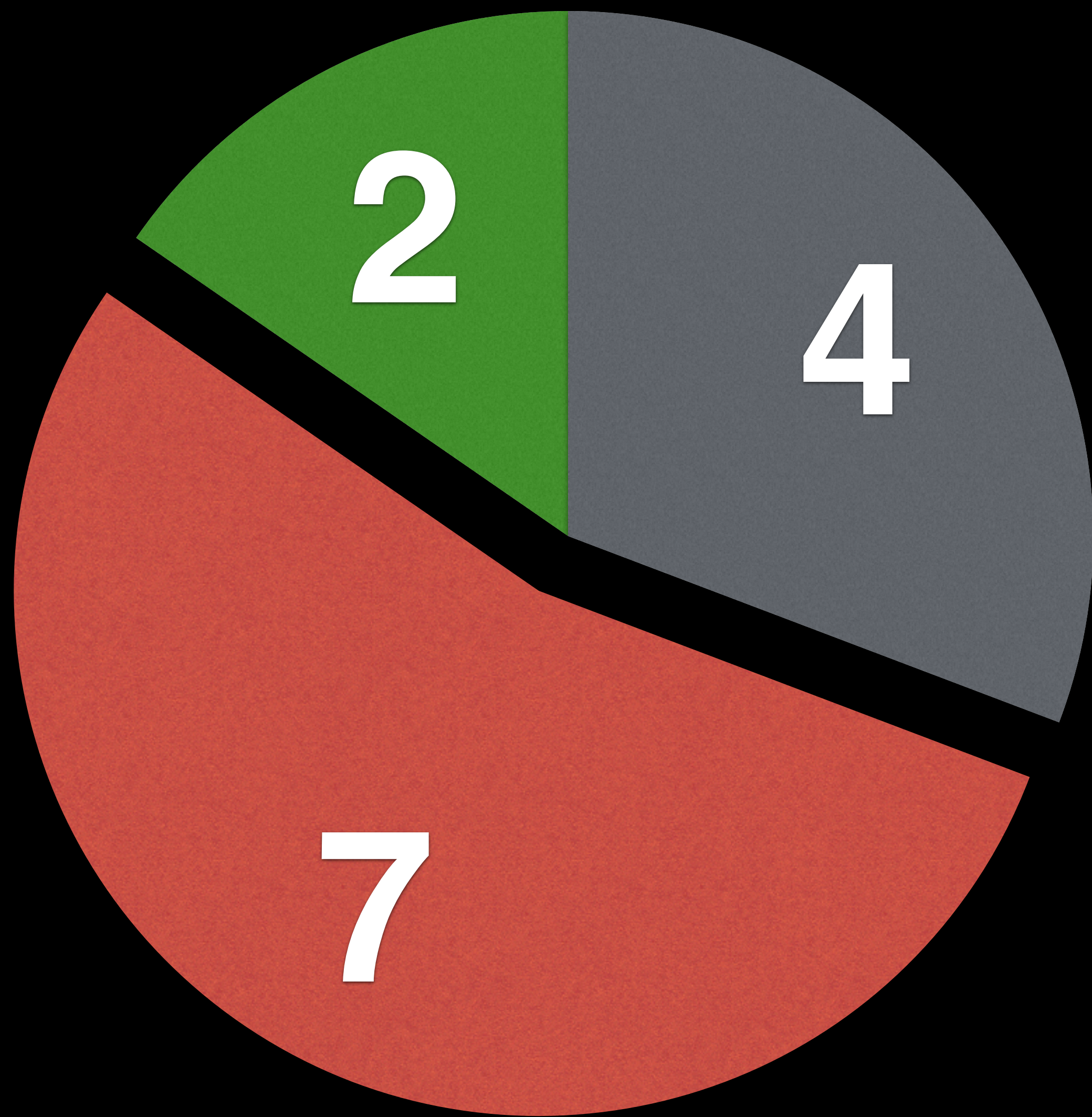


Out of scope



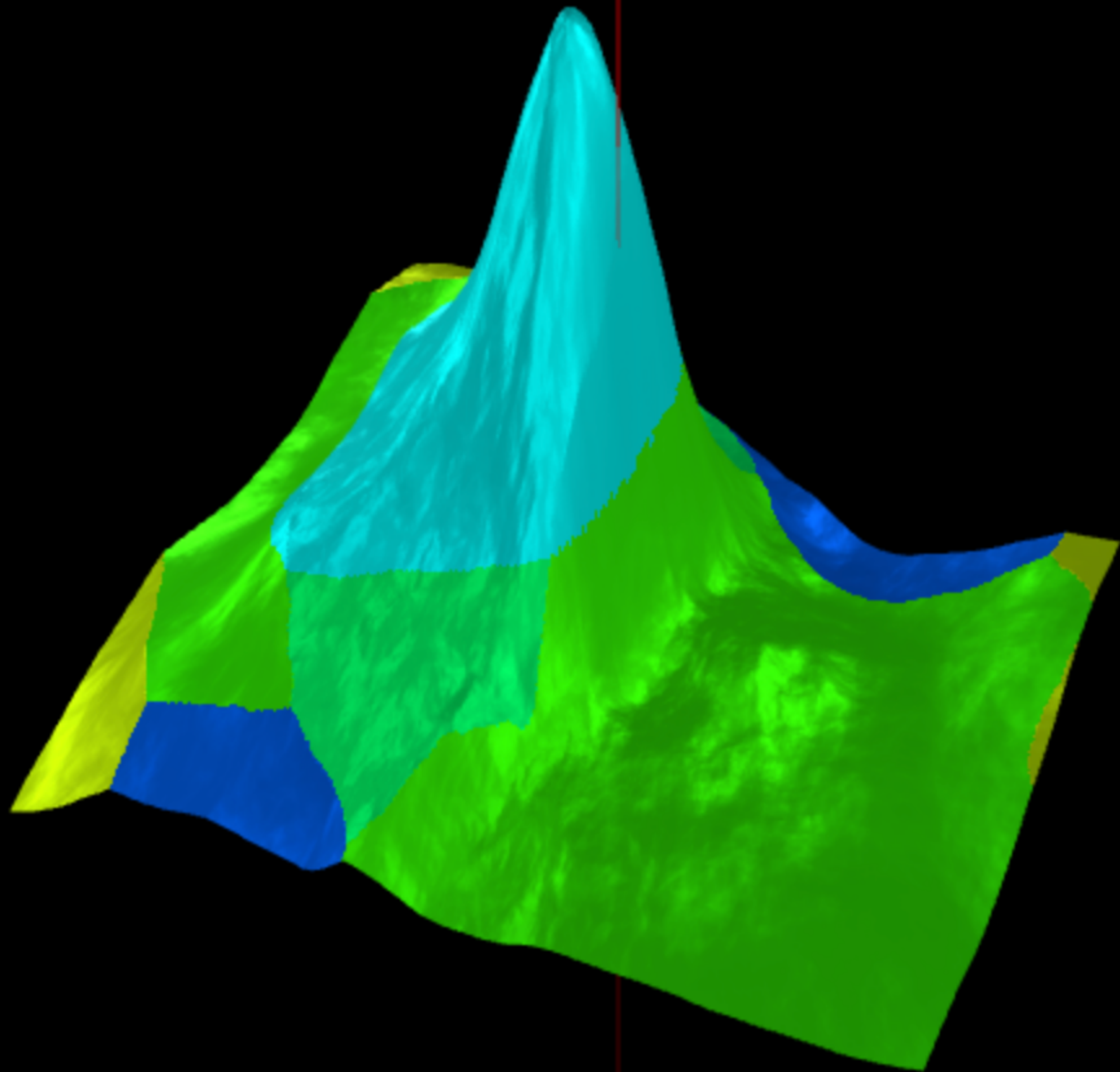
● **Out of scope**

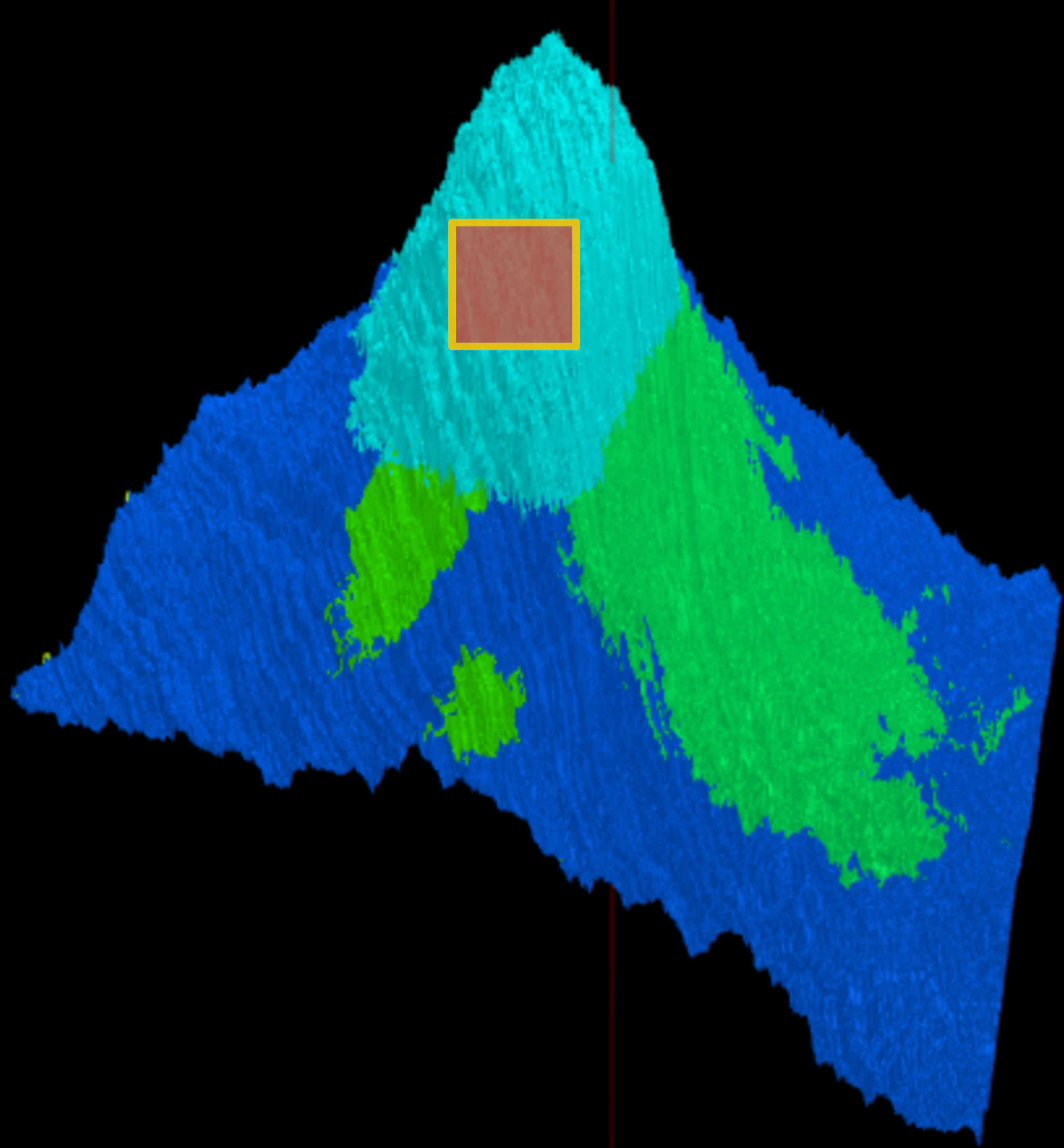
● **Correct Defenses**

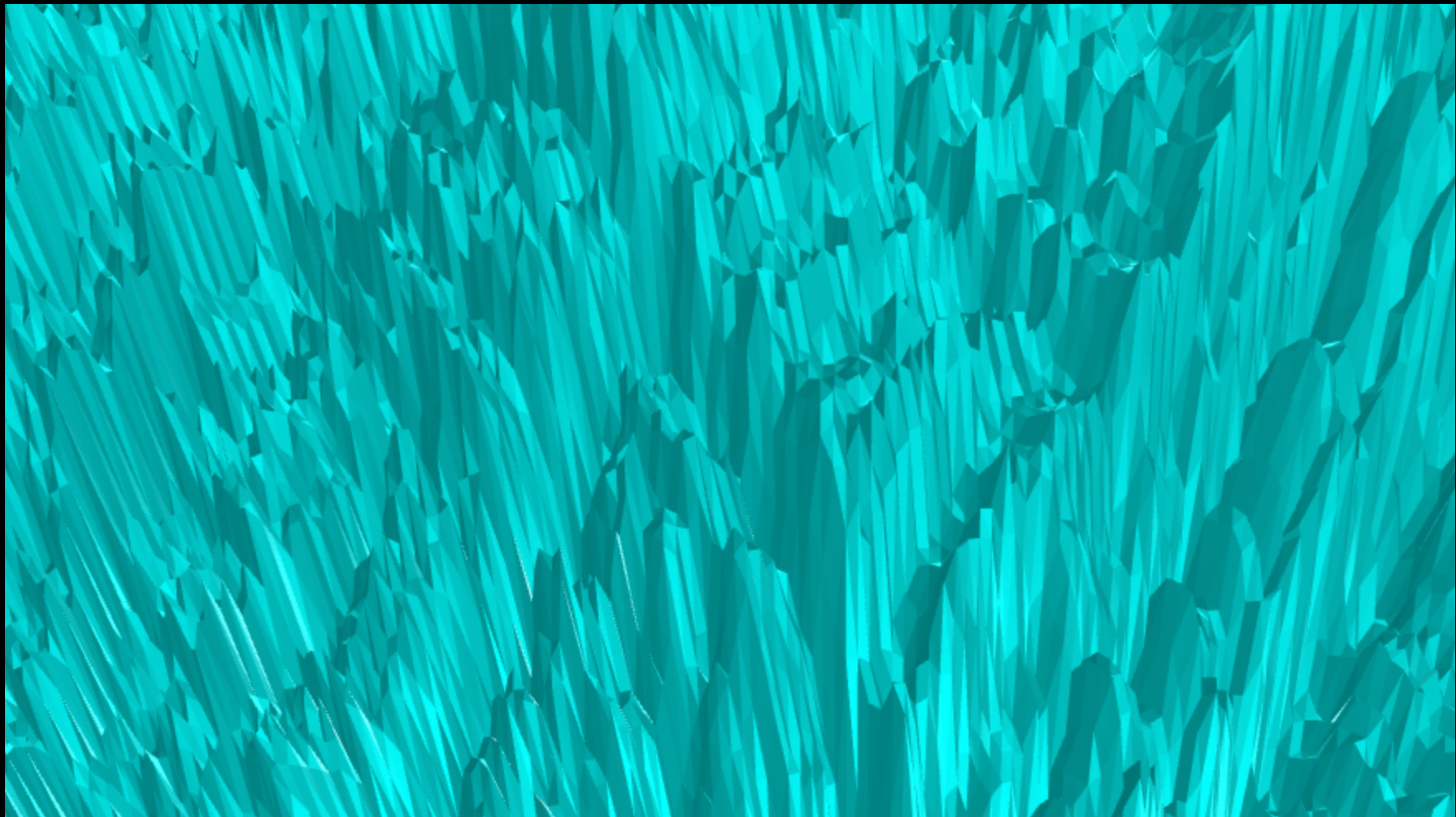


- **Out of scope**
- **Broken Defenses**
- **Correct Defenses**

So what did
defenses do?







Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods

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

Abstract

MagNet and proposed as a defense we can consider defenses with

1 Introduction

It is an open question they will be recently, three networks robust

- MagNet neural network through adversarial examples to lie on the data classification the which MagNet adversarial the parameter

- An efficient neural network during function but claim

- Adversarial

We identify a new sense of adversarial examples based on gradients. When they are used, the effect can be characteristic and for each gradient we to overcome certified we find otherwise, with gradients. We present 6 counterexamples that threaten most

1. Introduction

In response to adversarial examples there has been significant progress has been made against adversarial the adversary solution has not

As benchmark attacks (e.g., Kurakin & Wagner)

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

Neural network adversarial examples two white-box 2018 and find existing techniques of the defenses

1. Introduction

Training neural network adversarial examples (Two defenses that this problem: “Interception” (Practical Adversarial Attacks Denoiser” (Liao)

In this note, we present in the white-box adversarial examples that fool ImageNet datasets a small ℓ_∞ perturbation considered in the targeted adversarial

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 code² which we build on³. We hope that future work will continue to release source code by publication time to accelerate progress in this field.

A. Evaluation



Abst

We sh more unpro

1 I

It is a they v sive d to mal versar

In t tillatio ificati ample strate

Dis advers inputs attack magni two of achiev of ima

2 B

2.1

We as sarial briefly

Let param last la layer $F(\theta, x)$ cation

probab $C(\theta, x)$ tion o we are greysc

ABSTRACT Neural network examples: inputs that In order to better survey ten recent papers to compare their effectiveness. We propose a new loss function that is significantly harder to optimize than the properties believed to make adversarial examples effective. Finally, we propose a future defense

1 INTRODUCTION Recent years have seen a proliferation of neural network defenses. A major driving force behind this has been demonstrated by the success of [38], to beat the state-of-the-art defenses on ImageNet cars [6].

In this paper, we propose a new defense. When they will be recently, three networks robust to adversarial examples. We find that MagNet, proposed as a defense we can consider defenses with obfuscated gradients, and proposed as a defense we can consider defenses with obfuscated gradients, and proposed as a defense we can consider defenses with obfuscated gradients.

Due to this, we propose a new defense. We find that MagNet, proposed as a defense we can consider defenses with obfuscated gradients, and proposed as a defense we can consider defenses with obfuscated gradients.

Permission to make personal or classroom use is granted by ACM for profit or commercial resale. This permission is granted without fee except for copying beyond the limits permitted by Sections 107 and 108 of the 1976 Copyright Act. For more information, contact the ACM Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923. Copyright © 2017 Copyright ACM ISBN 978-1-4503-4328-1/17/0000-0000 \$15.00 https://doi.org/10.1145/3123456

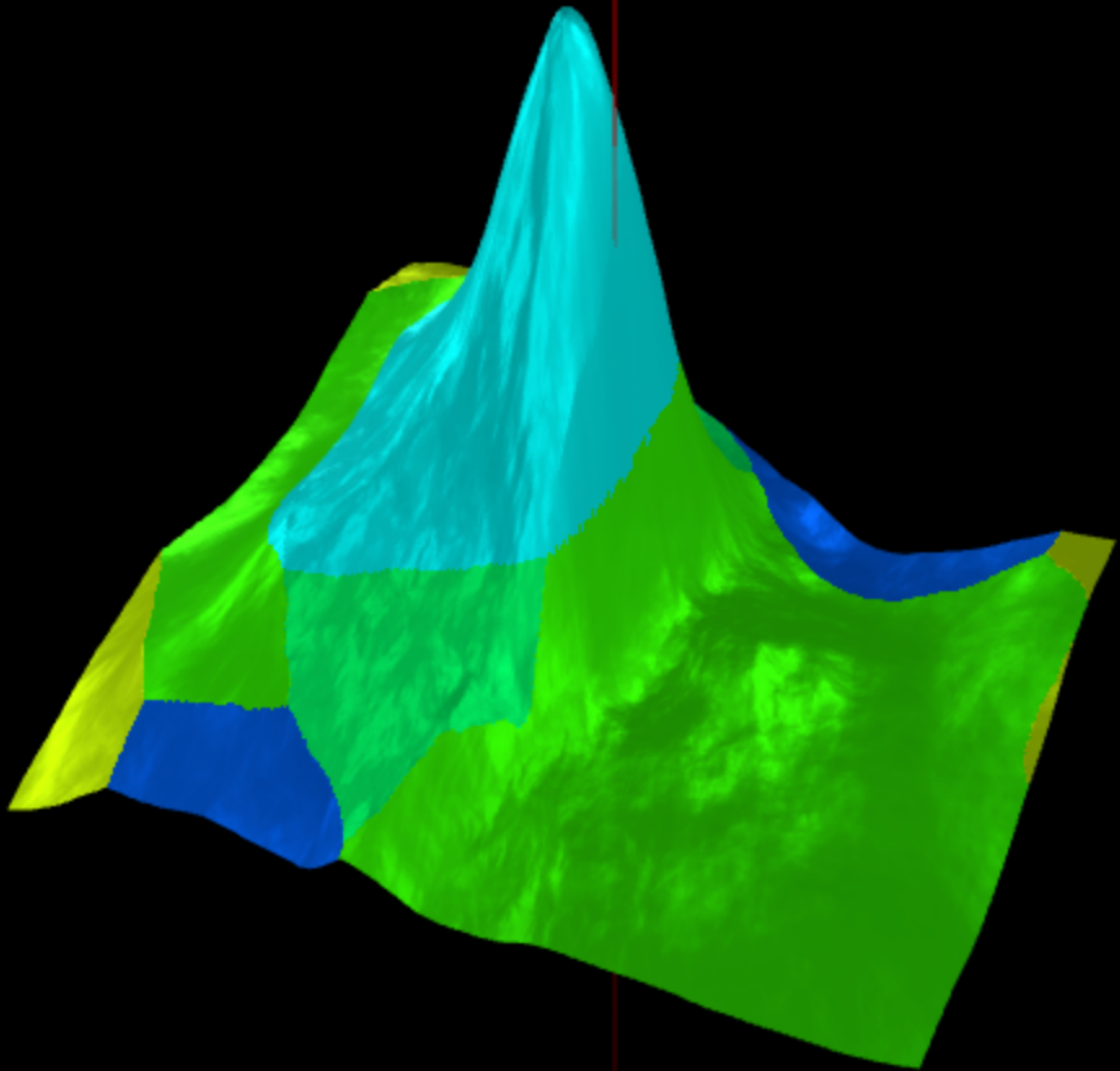
Lessons Learned from
Evaluating the Robustness of
Defenses to Adversarial Examples

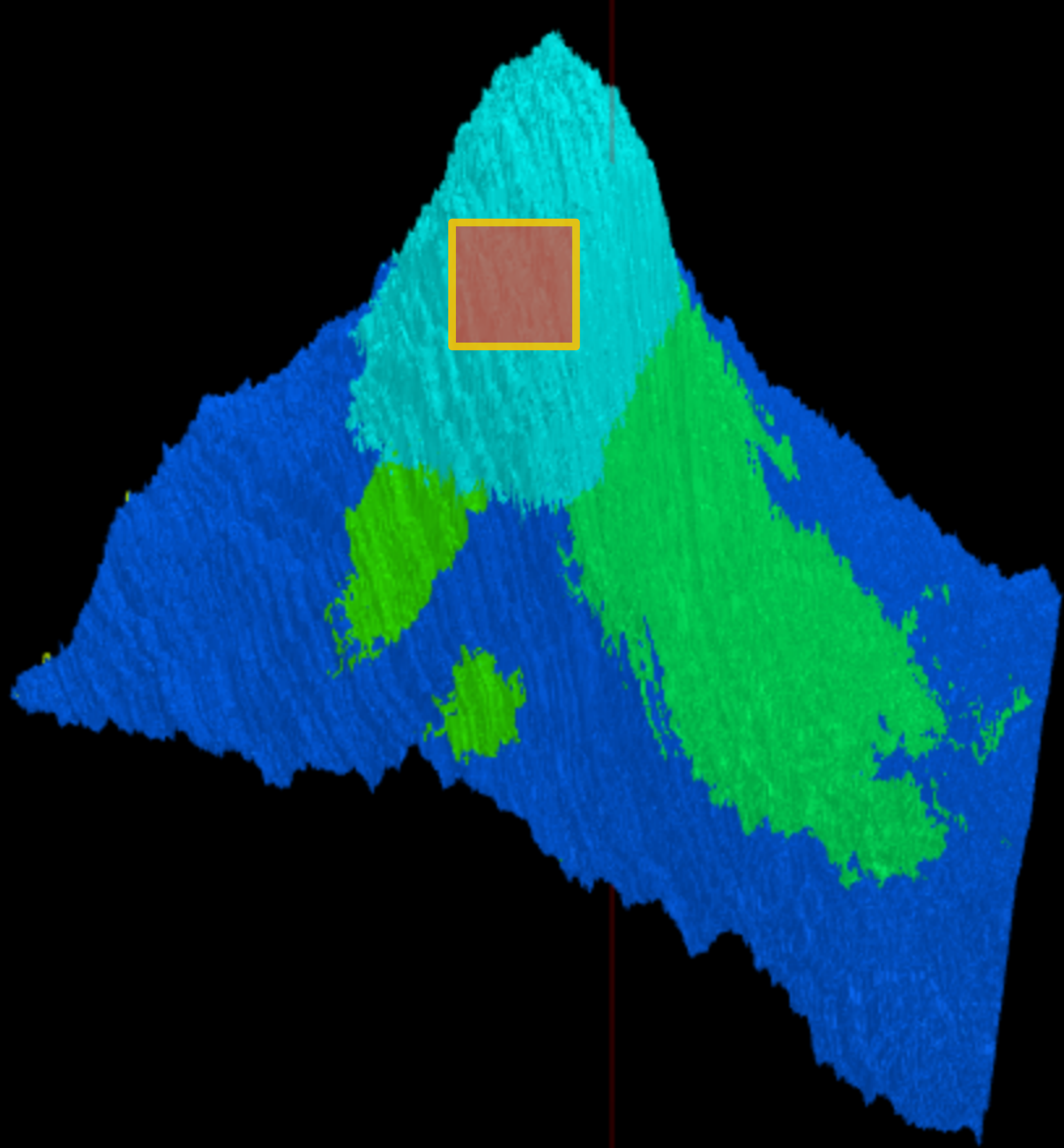
Lessons (1 of 2)

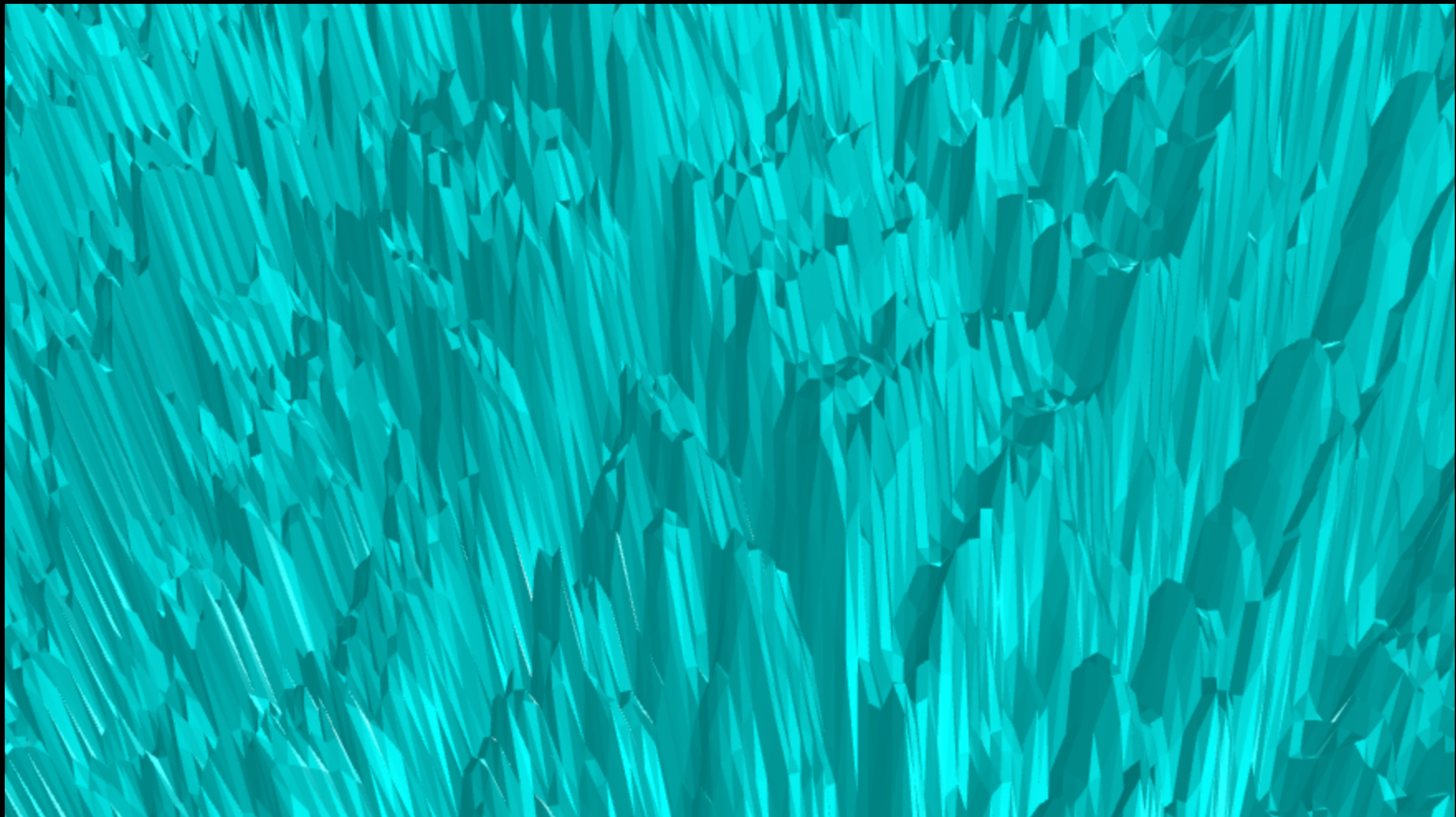
*what we learn from evaluations
(and why to evaluate thoroughly)*

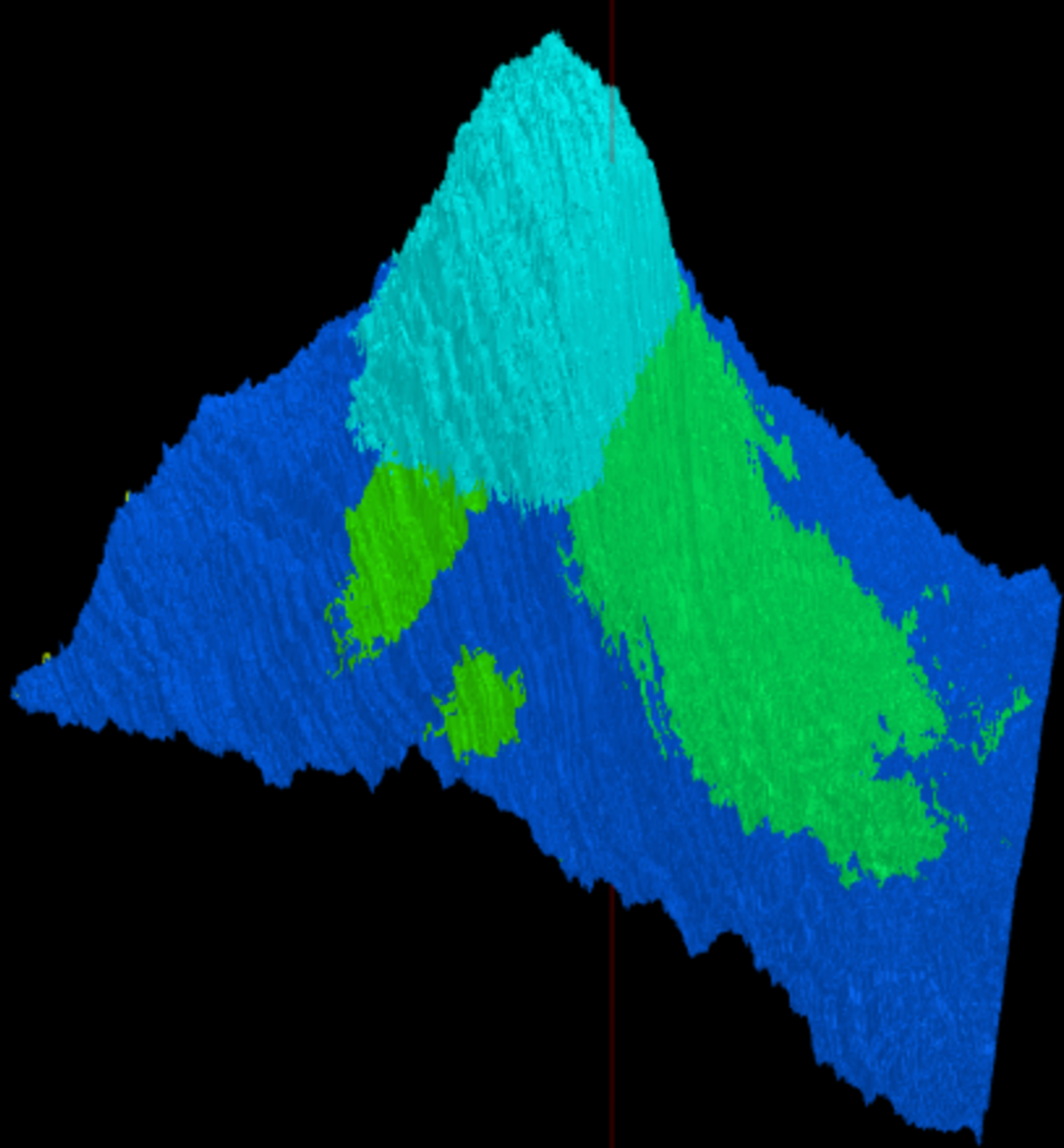
A Brief History of ~~Time~~ Defenses

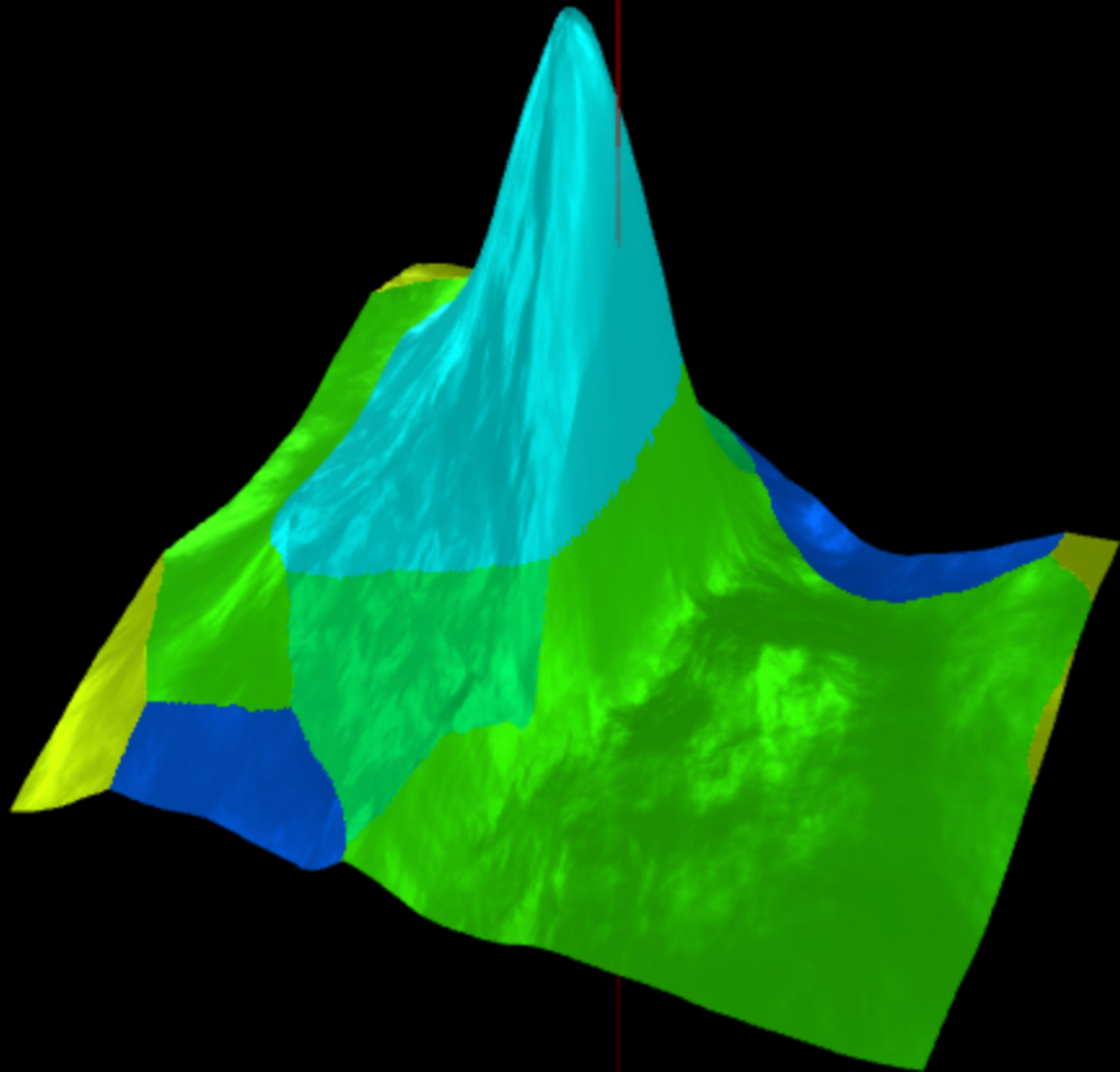
- S&P'16 - *gradient masking*
- ICLR'17 - *attack objective functions*
- CCS'17 - *transferability of examples*
- ICLR'18 - *obfuscated gradients*



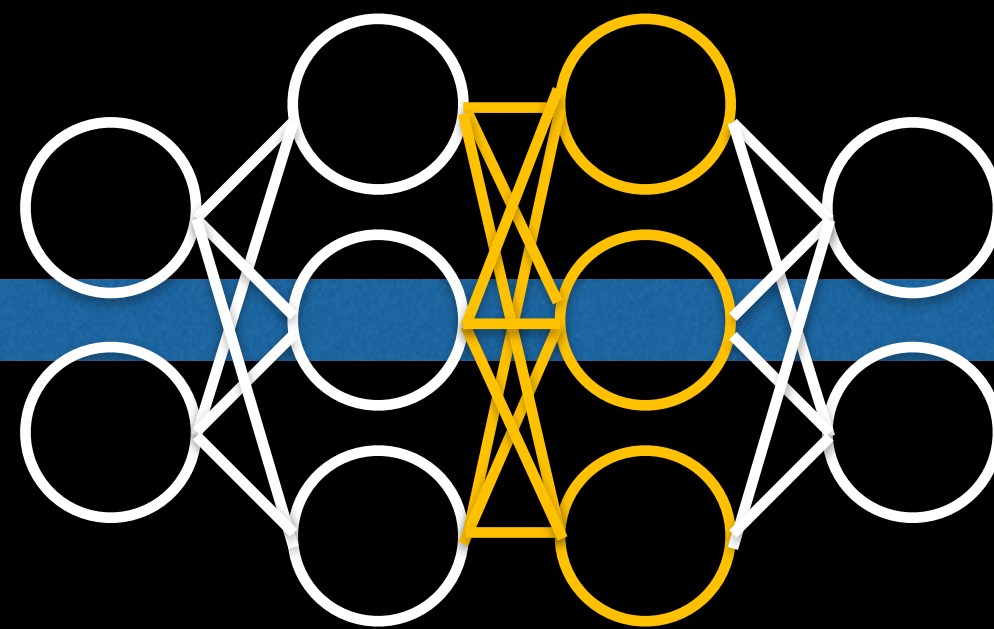
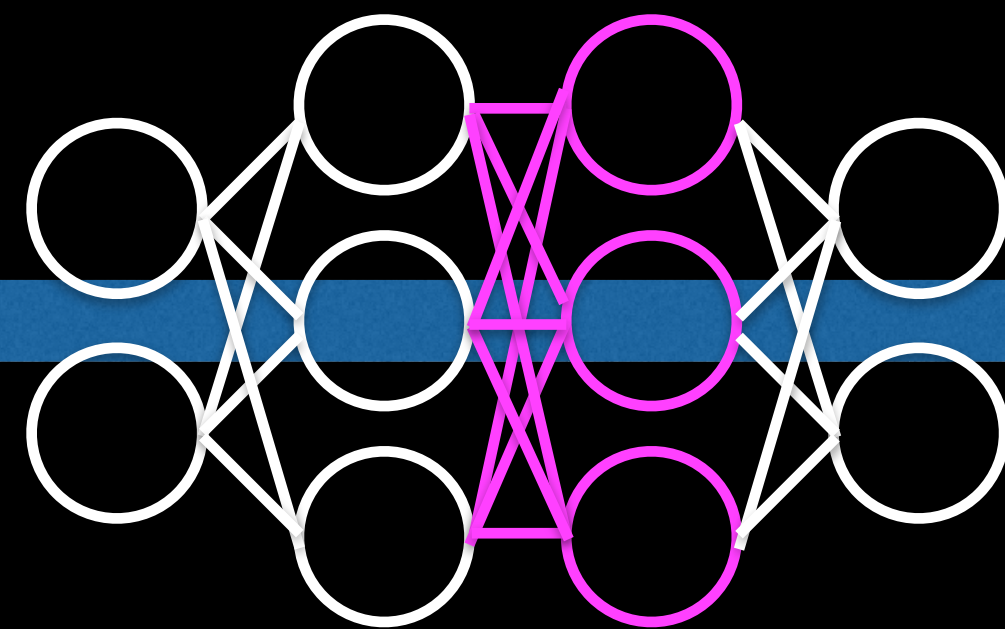




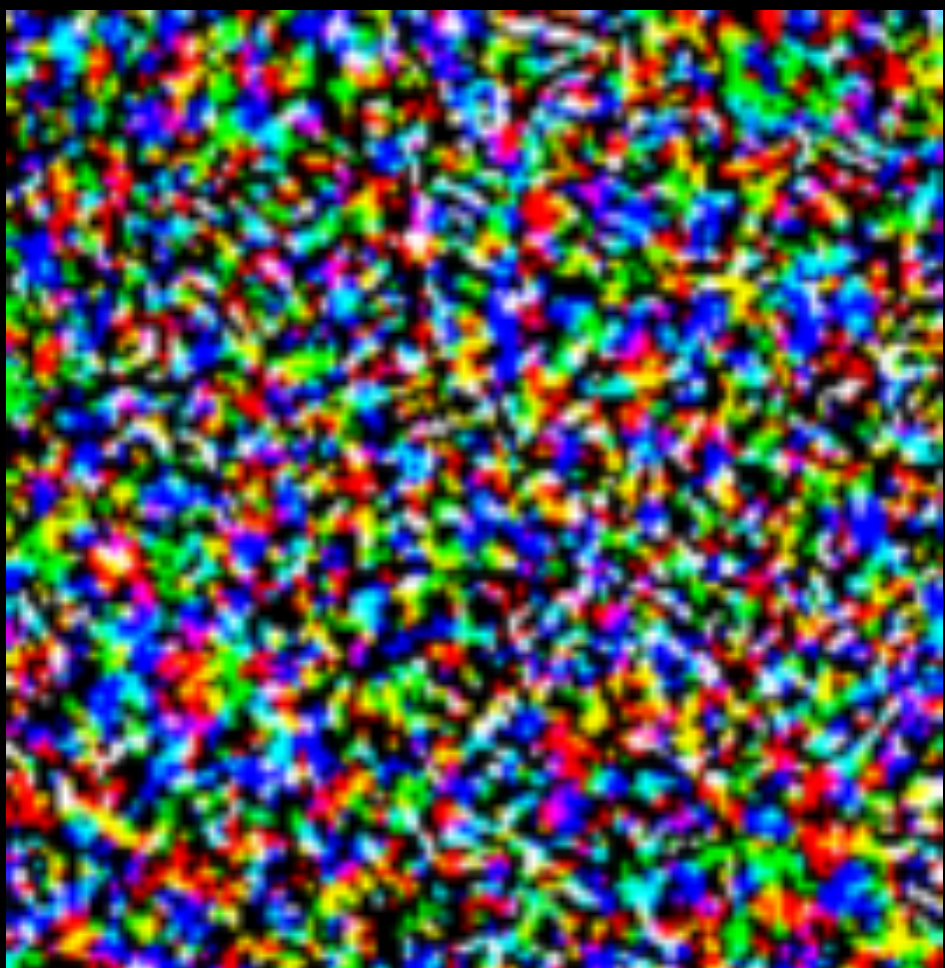




"Fixing" Gradient Descent



**[0.1,
0.3,
0.0,
0.2,
0.4]**

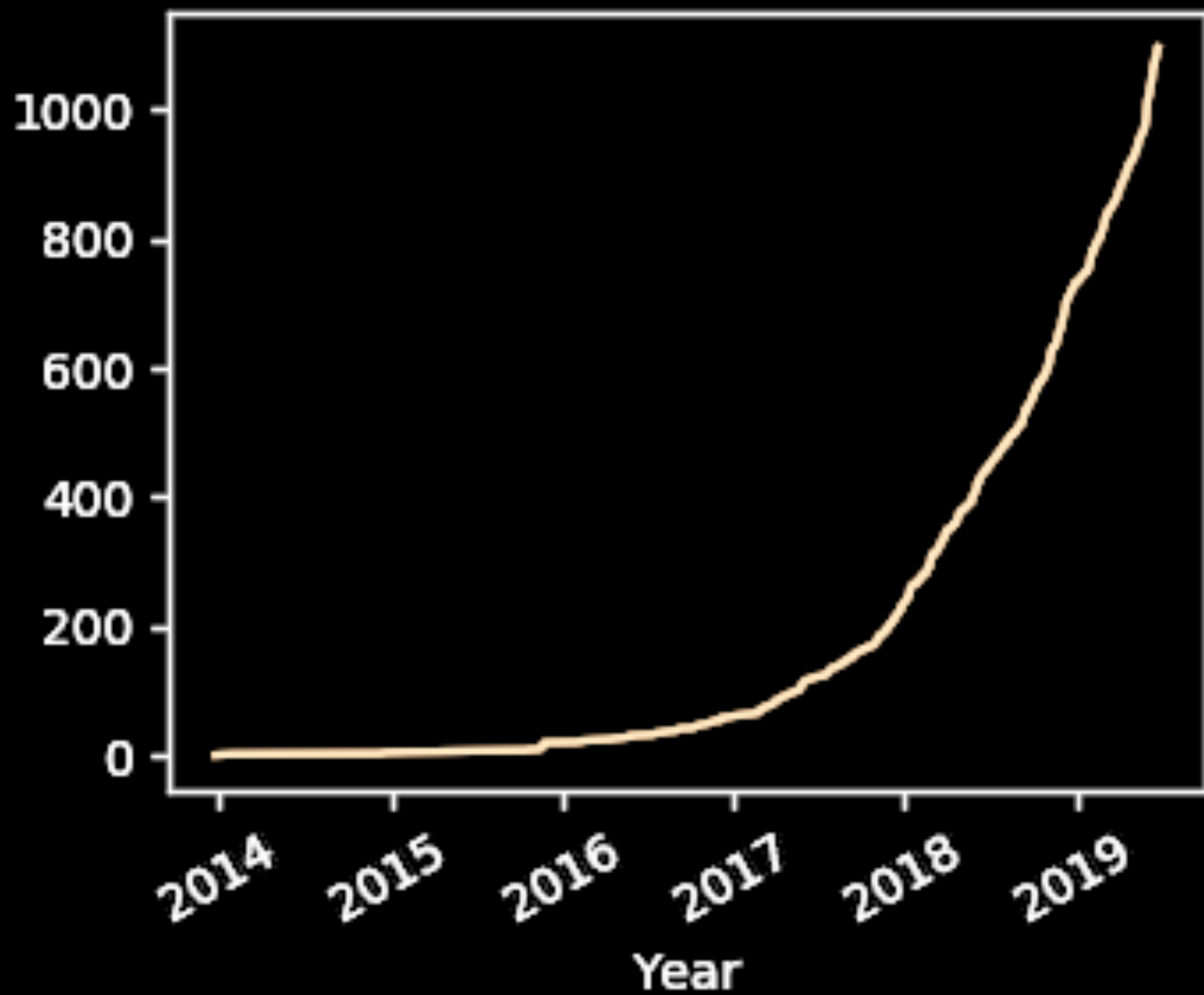


Disentangling
true robustness
from
apparent robustness
is nontrivial

Lessons (2 of 2)

performing better evaluations

Cumulative Number of Adversarial Example Papers



ON EVALUATING ADVERSARIAL ROBUSTNESS

Nicholas Carlini¹, Anish Athalye², Nicolas Papernot¹, Wieland Brendel³, Jonas Rauber³,
Dimitris Tsipras², Ian Goodfellow¹, Aleksander Mądry², 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).

²The 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

3.1. Effectiveness

Adversarial Attacks. We test on the following attacks:

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.

4. Related Work

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:

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

Here, y^* is the label-vector, $\gamma \in [10^{-3}, 10^0]$ is a scalar found through a bisection search, L_{fp} is the fingerprint-loss

5. Discussion and Future Work

3.4. Robustness to Adaptive Whitebox-Attackers

Adversarial Attacks. We test on the following attacks:

3.1. Effective

4. Related Work

3.4. Robustness to Adaptive Whitebox-Attackers

5. Discussion and Future Work

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

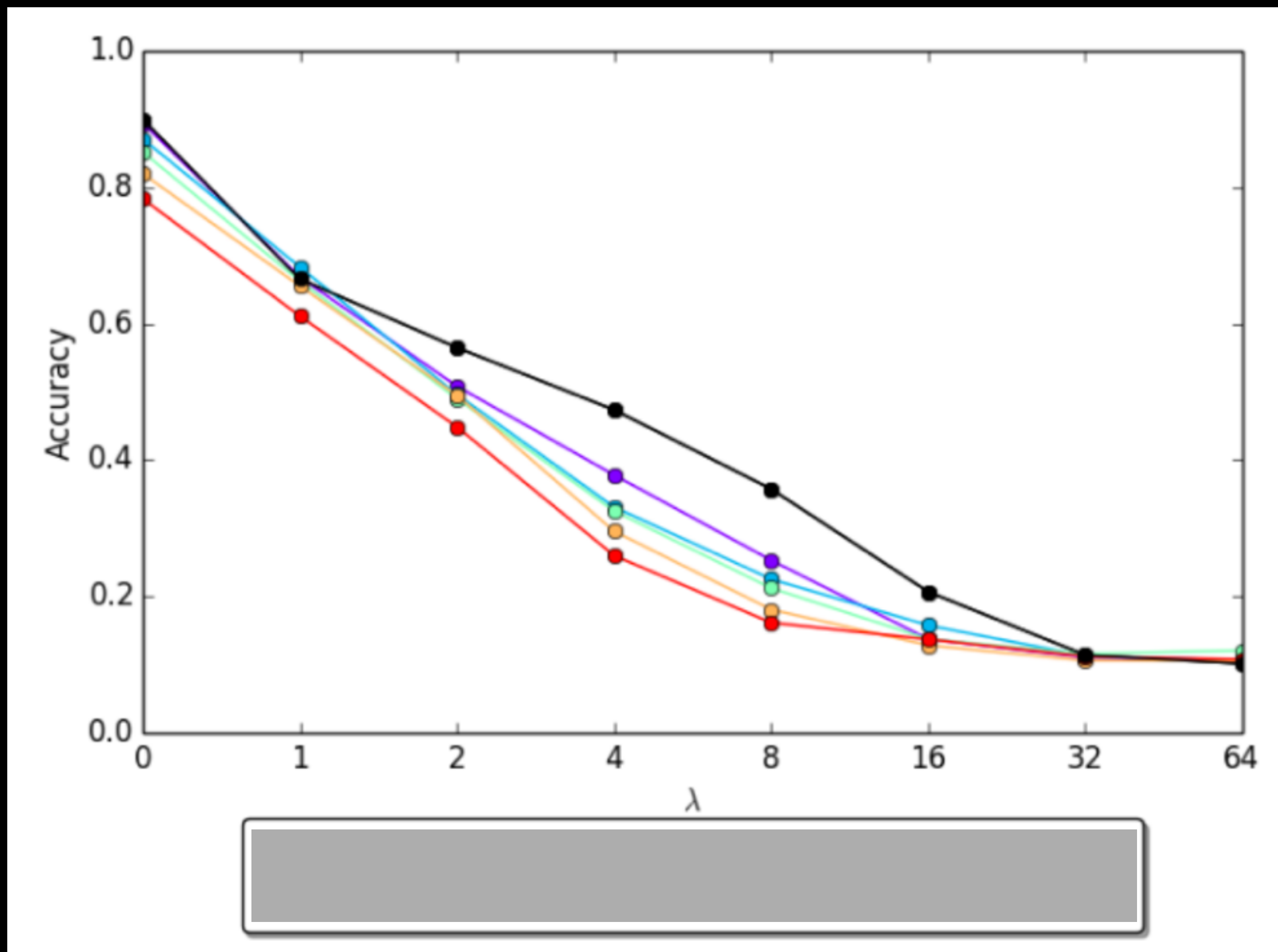
	Model	FGSM	PGD
<i>Clean</i>		25.10	4.10
		46.15	1.66
		43.89	3.57
		52.07	53.11
		48.50	50.50



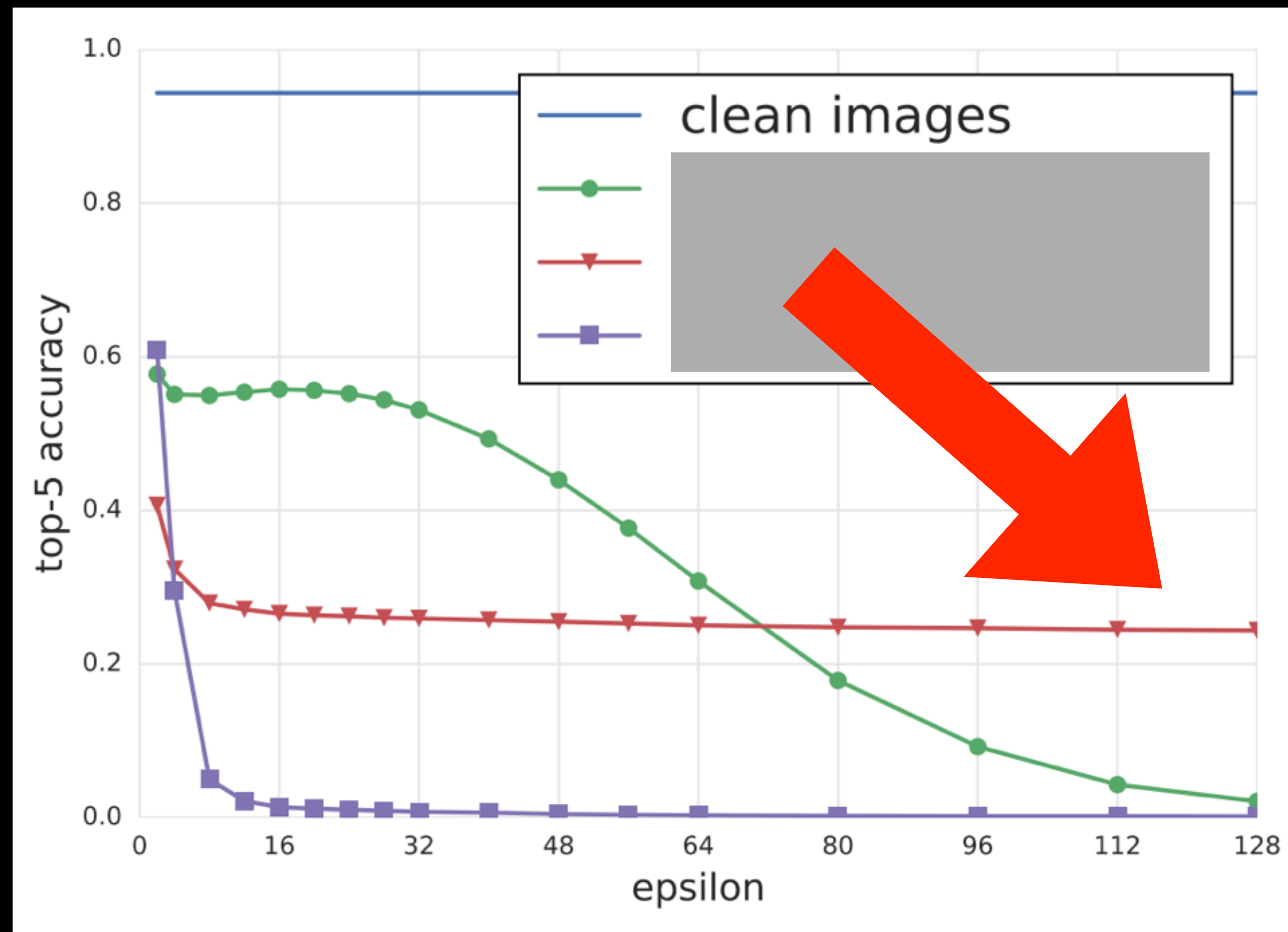
Iterative attacks should always do better than single step attacks.

Attack	Parameter	Fooling Rate	Detection Rate
DeepFool		99.35%	97.83%
Carlini	$\kappa=0.0$	100.0%	95.66%

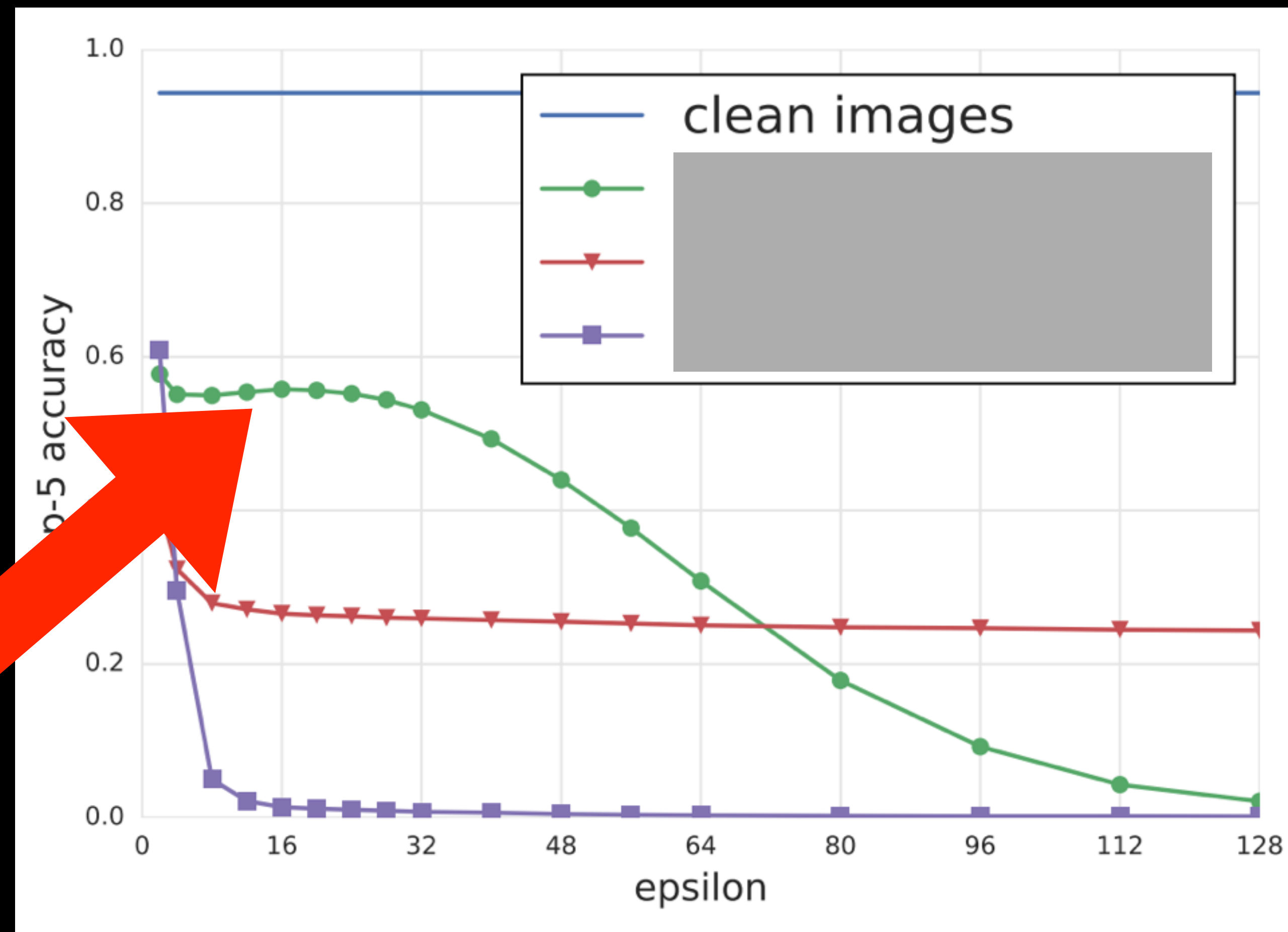
Unbounded optimization attacks should eventually reach in 0% accuracy



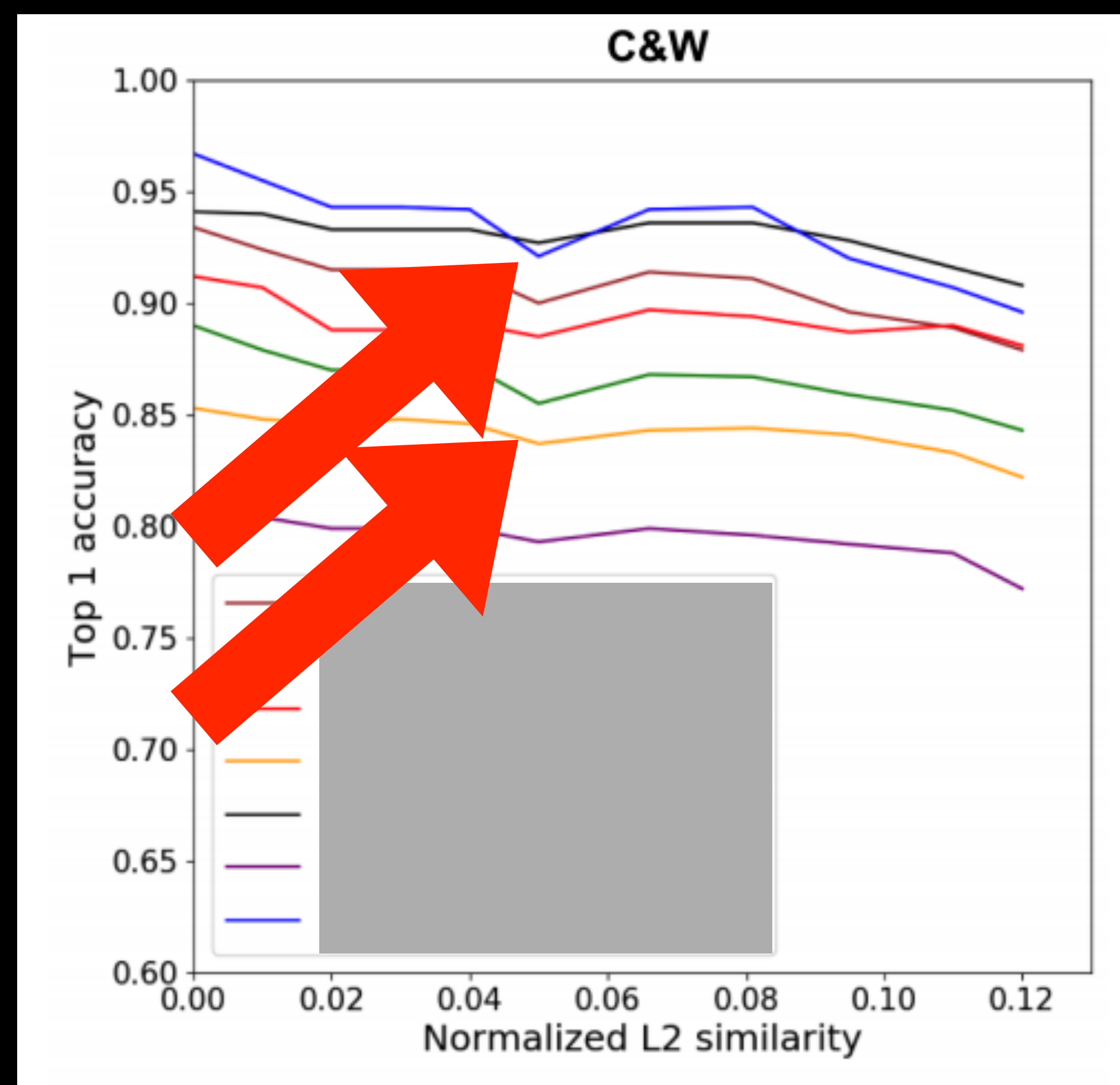
Unbounded optimization attacks should eventually reach in 0% accuracy



Unbounded optimization attacks should eventually reach in 0% accuracy



Model accuracy should be monotonically decreasing

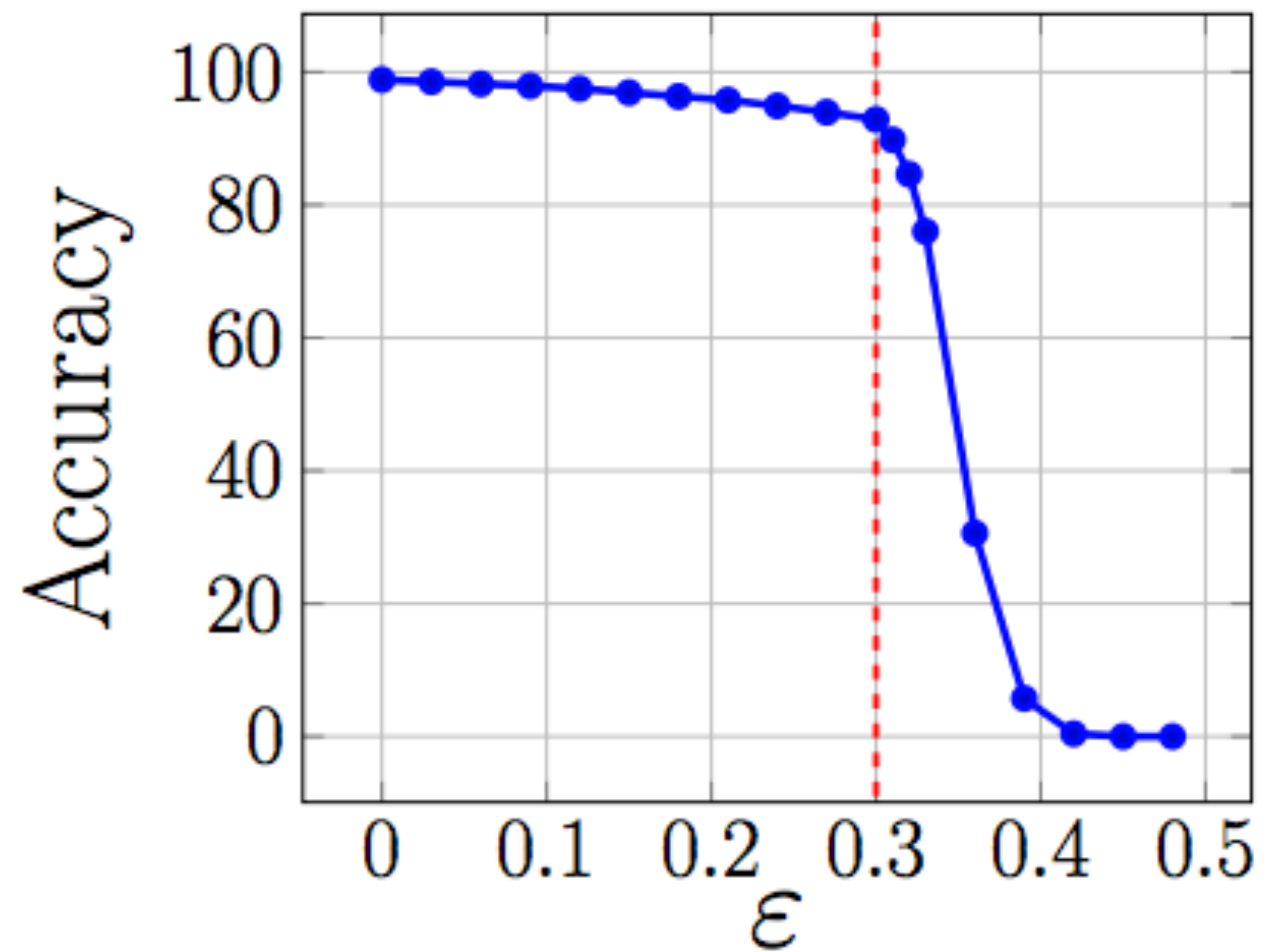


Model accuracy should be monotonically decreasing



Model	clean	step_ll		step_FGSM		iter_FGSM		CW	
		$\epsilon=2$	$\epsilon=16$	$\epsilon=2$	$\epsilon=16$	$\epsilon=2$	$\epsilon=4$	$\epsilon=2$	$\epsilon=4$
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

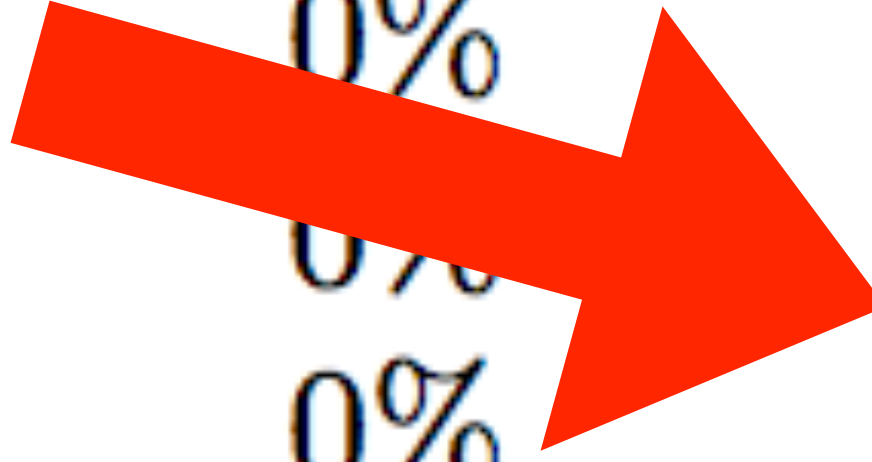
Evaluate against the
worst attack



(a) MNIST, ℓ_∞ norm

Plot accuracy vs distortion

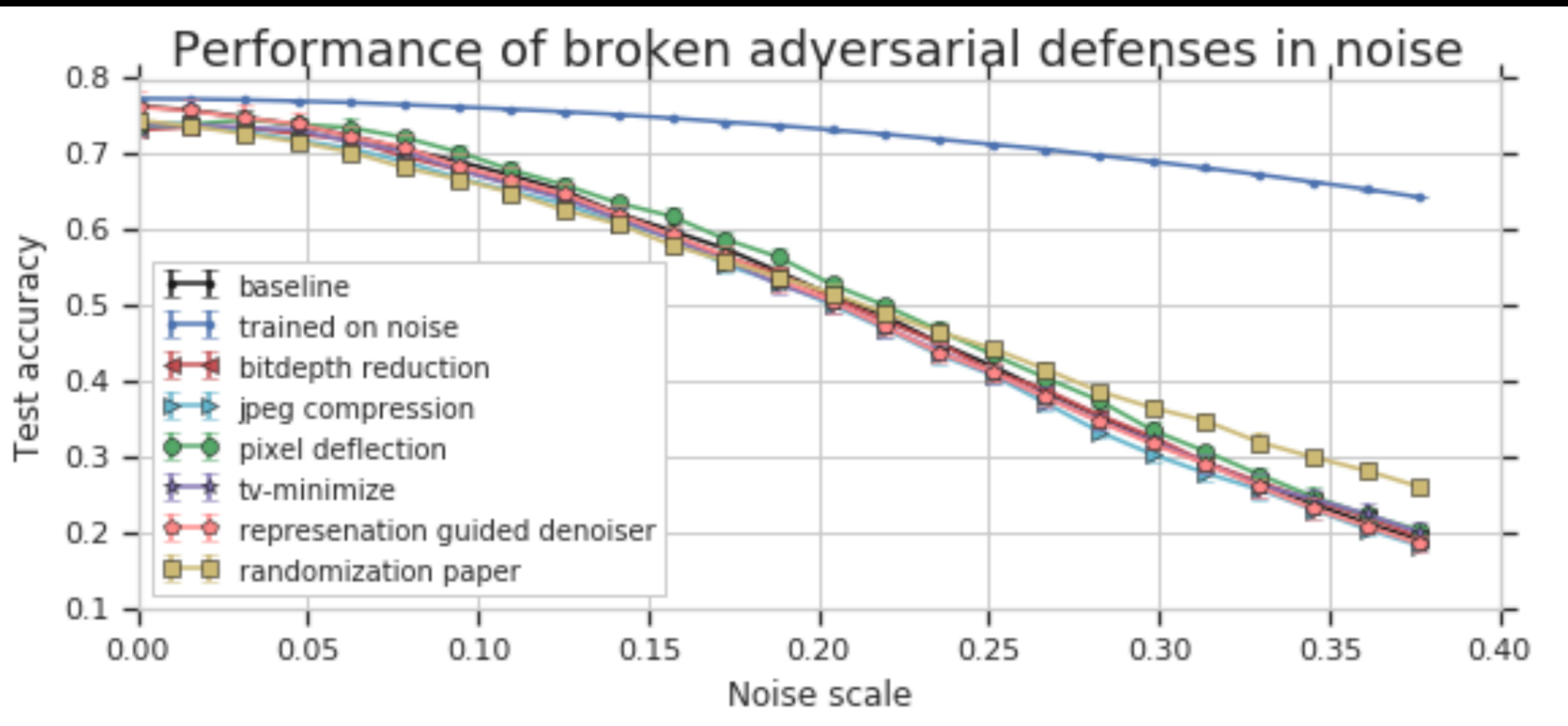
MaxIter	Model1	Model2	Model3	Model4
Natural	99.1%	98.5%	98.7%	98.2%
100	70.2%	91.7%	77.6%	75.6%
1000	0.05%	51.5%	20.3%	24.4%
10K	0%	16.0%	20.1%	24.4%
100K	0%	9.8%	20.1%	24.4%
1M	0%	7.6%	20.1%	24.4%



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>