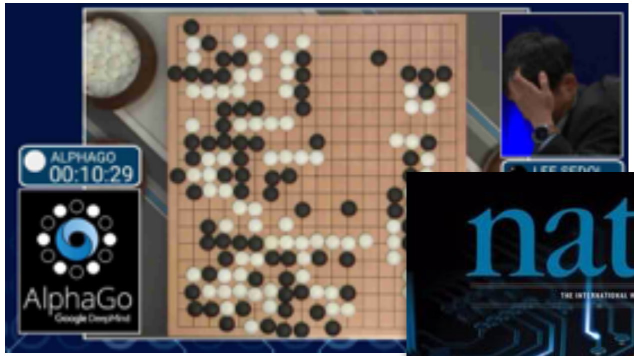
Machine Learning: A Robustness Perspective

Aleksander Mądry

madry-lab.ml

Machine Learning: The Success Story



Machine Learning: The Success Story

IS "DEEP LEARNING" A REVOLUTION IN ARTIFICIAL INTELLIGENCE?

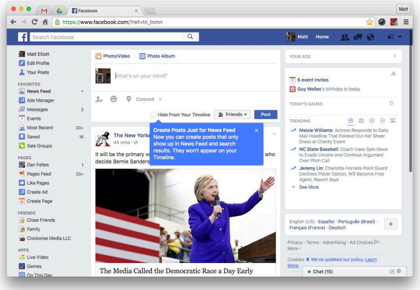
Trump Signs Executive Order Promoting Artificial Intelligence

2016: The Year That Deep Learning Took Over the Internet

WHY DEEP LEARNING IS SUDDENLY CHANGING YOUR LIFE

Andrew Ng @AndrewYNg Follow

"AI is the new electricity!" Electricity transformed countless industries; AI will now do the same.



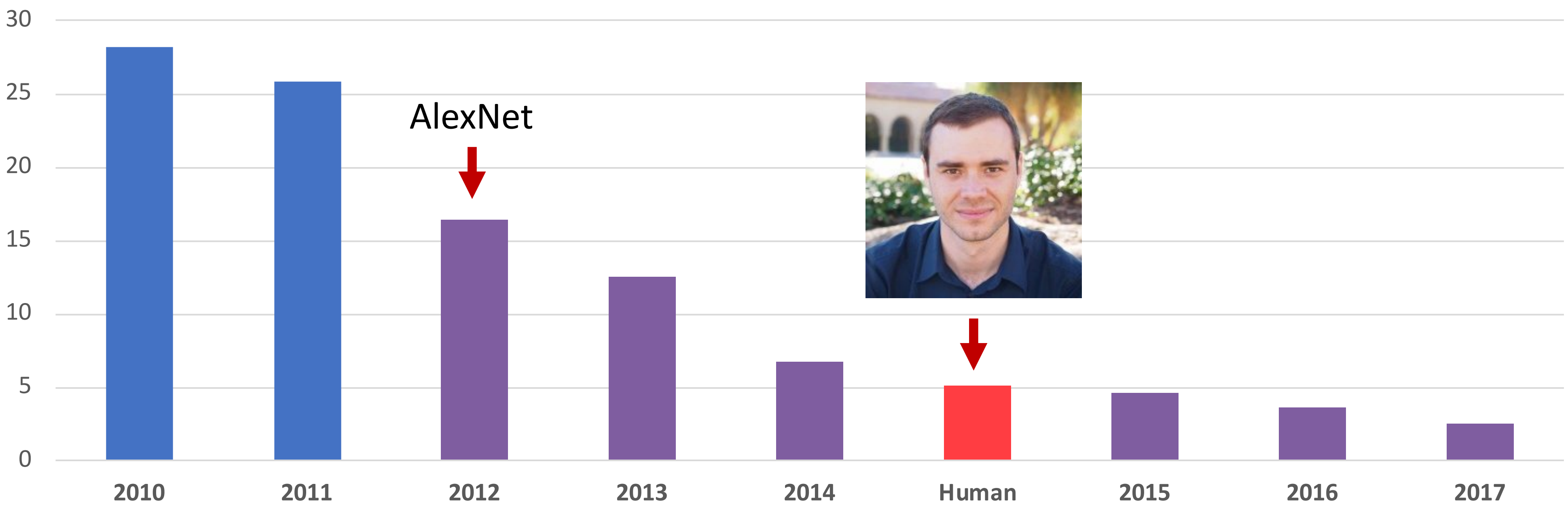
Is ML **truly** ready for real-world deployment?

Can We Truly Rely on ML?

Robust ML:
The Challenges

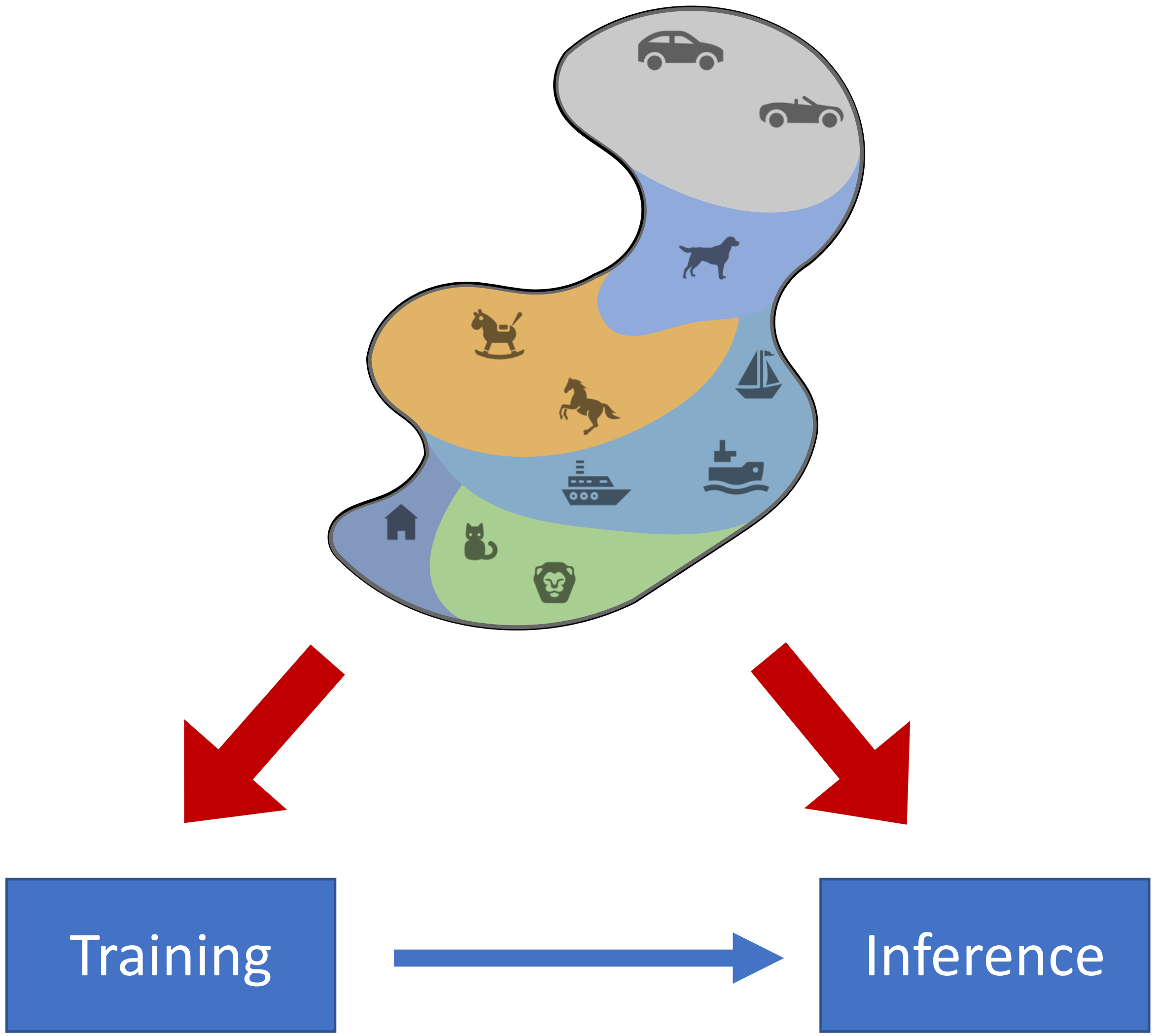
ImageNet: An ML Home Run

ILSVRC top-5 Error on ImageNet



But what do these results *really* mean?

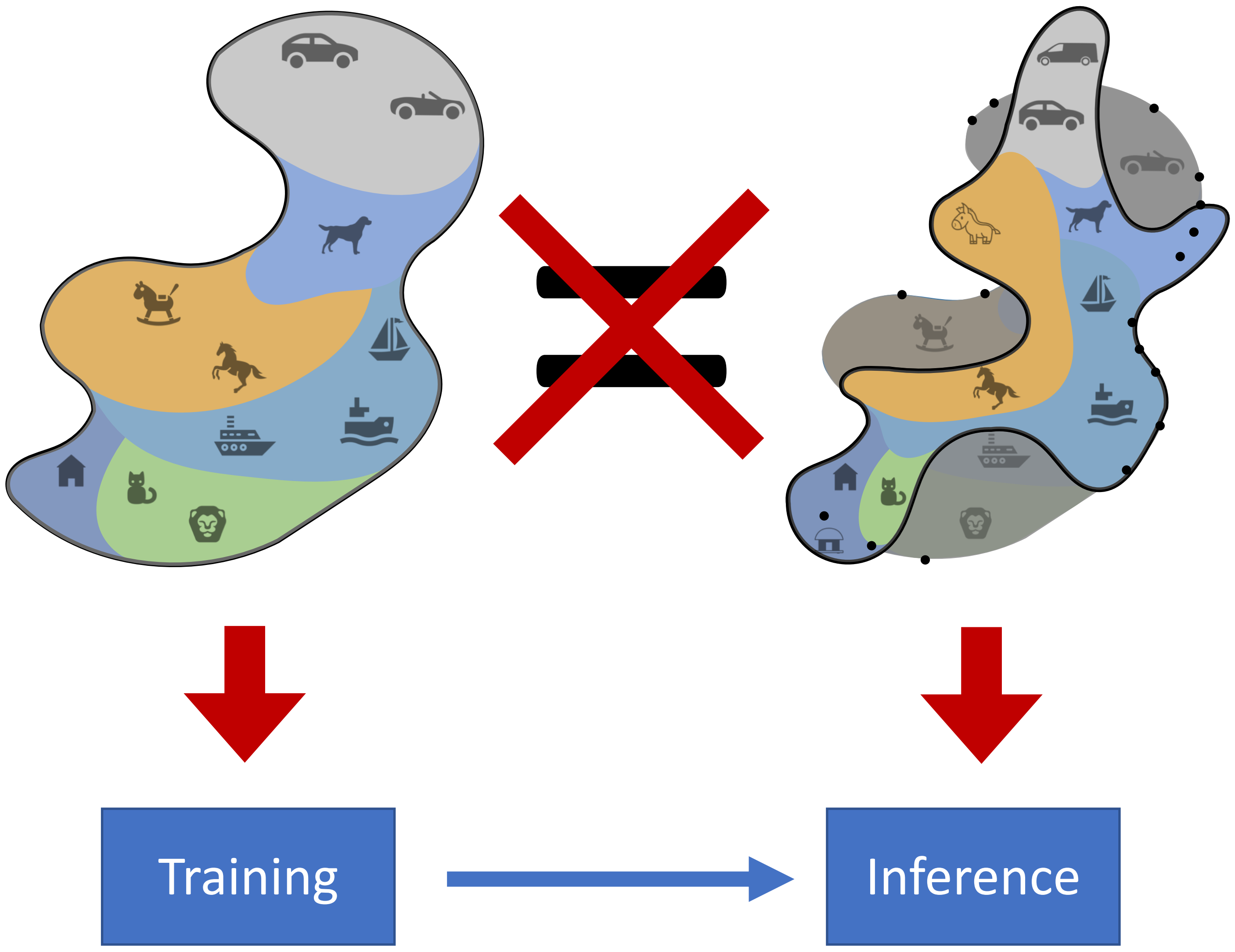
A Limitation of the (Supervised) ML Framework



Measure of performance:
Fraction of mistakes during testing

But: In reality, the distributions we **use** ML on are NOT the ones we **train** it on

A Limitation of the (Supervised) ML Framework



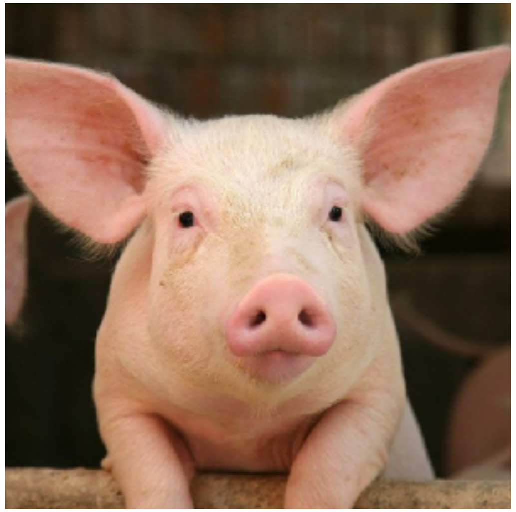
Measure of performance:
Fraction of mistakes during testing

But: In reality, the distributions we **use** ML on are NOT the ones we **train** it on

What can go wrong?

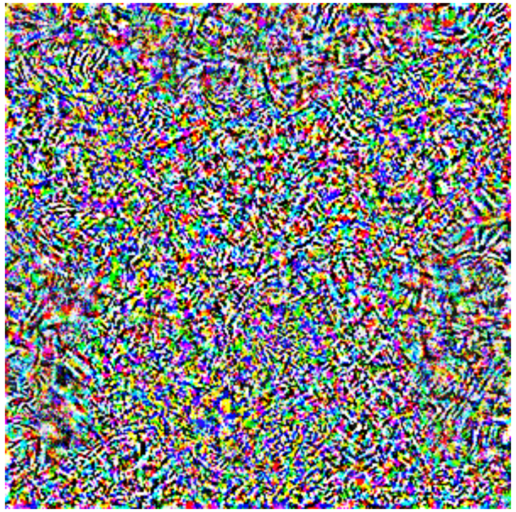
ML Predictions Are (Mostly) Accurate but Brittle

“pig” (91%)



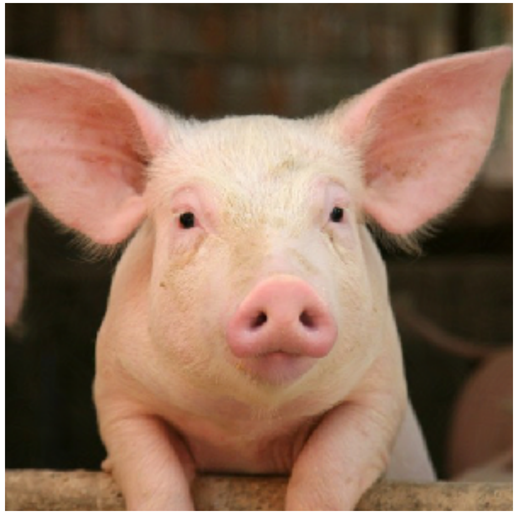
+ 0.005 x

noise (NOT random)



=

“airliner” (99%)



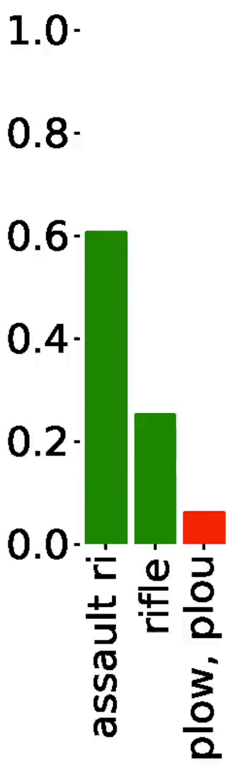
[Szegedy Zaremba Sutskever Bruna Erhan Goodfellow Fergus 2013]
[Biggio Corona Maiorca Nelson Srndic Laskov Giacinto Roli 2013]

But also: [Dalvi Domingos Mausam Sanghai Verma 2004][Lowd Meek 2005]
[Globerson Roweis 2006][Kolcz Teo 2009][Barreno Nelson Rubinstein Joseph Tygar 2010]
[Biggio Fumera Roli 2010][Biggio Fumera Roli 2014][Srndic Laskov 2013]

ML Predictions Are (Mostly) Accurate but Brittle

[Athalye Engstrom Ilyas Kwok 2017]

ML Predictions Are (Mostly) Accurate but Brittle



[Fawzi Frossard 2015]
[Engstrom Tran Tsipras Schmidt M 2018]:
Rotation + Translation suffices to fool state-of-the-art vision models

→ Data augmentation does **not** seem to help here either

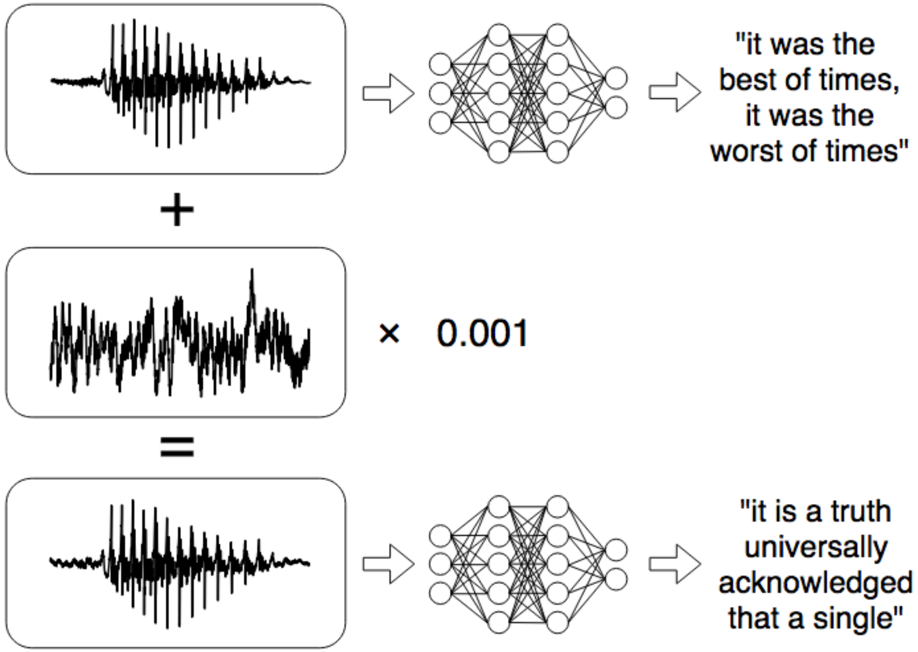
So: Brittleness of ML is a thing

Should we be worried?

Why Is This Brittleness of ML a Problem?

→ Security

[Carlini Wagner 2018]:
Voice commands that are unintelligible to humans

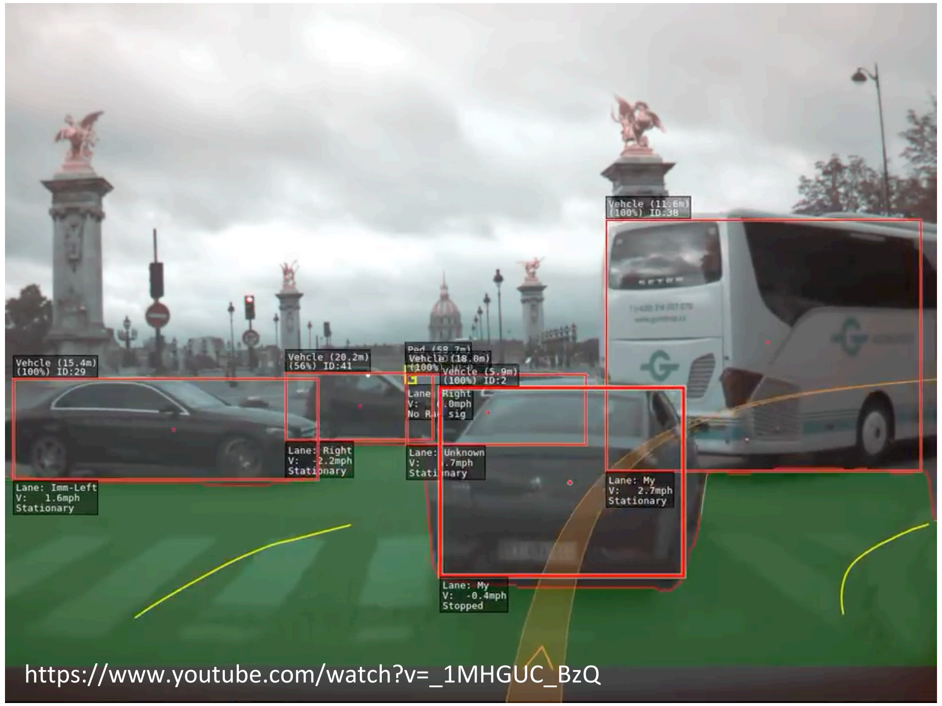


[Sharif Bhagavatula Bauer Reiter 2016]:
Glasses that fool face recognition

Why Is This Brittleness of ML a Problem?

→ Security

→ Safety

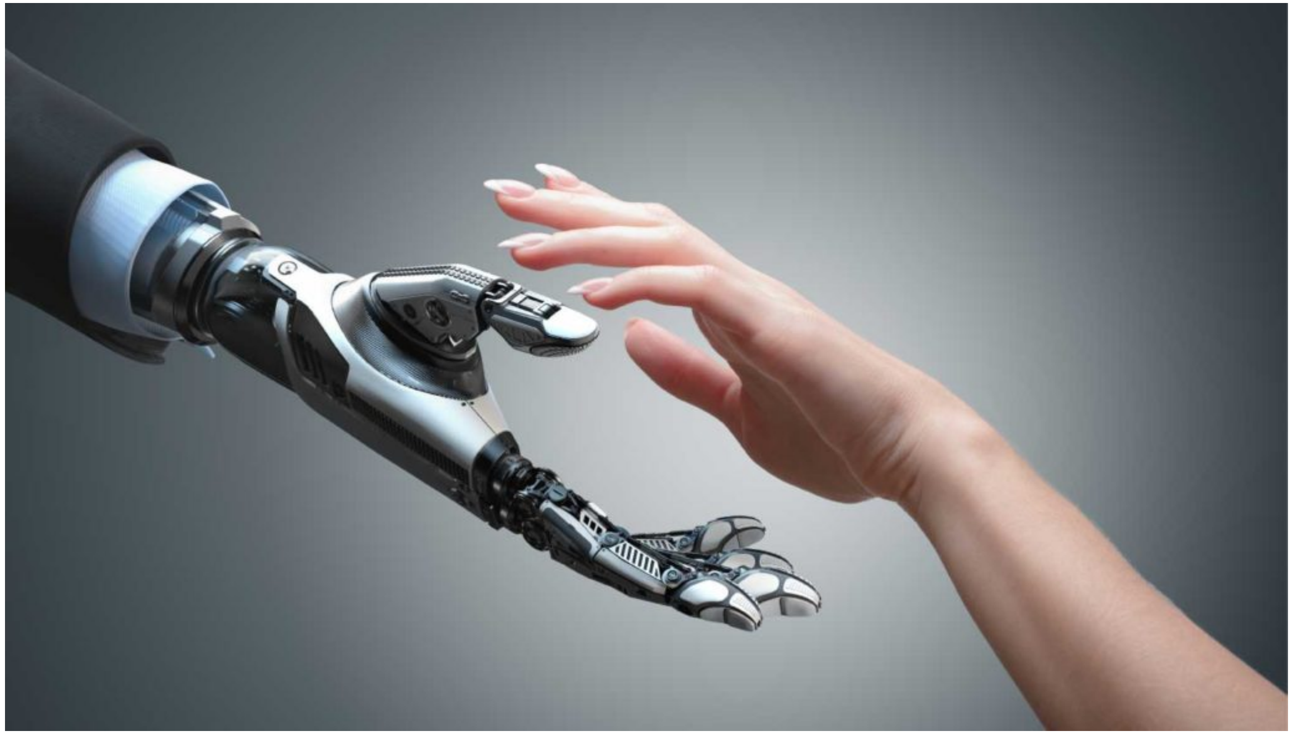


Why Is This Brittleness of ML a Problem?

→ Security

→ Safety

→ ML Alignment



Need to understand the
"failure modes" of ML

Is That It?

Training → Inference

Data poisoning

Adversarial Examples

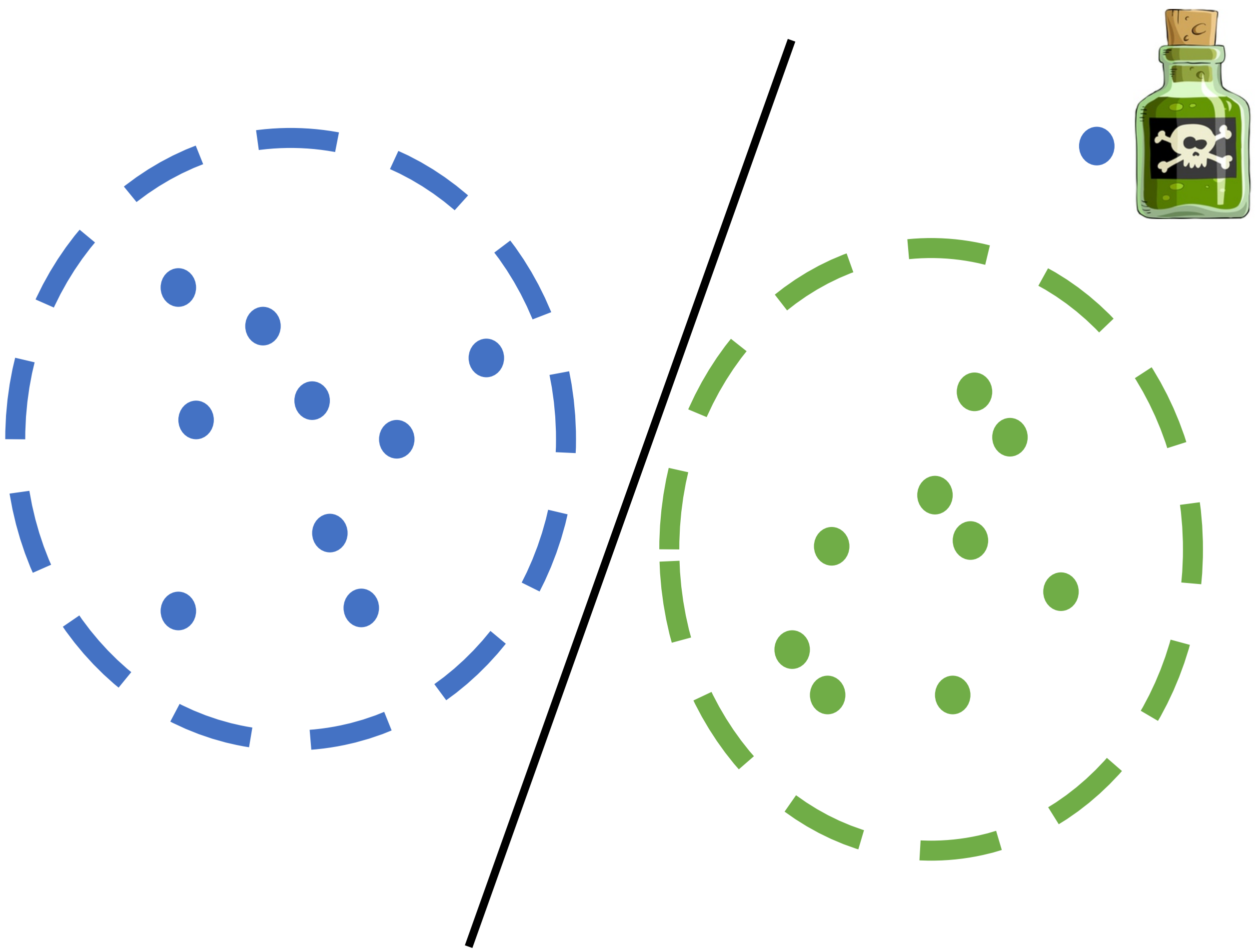
(Deep) ML is ”data hungry”

→ Can’t afford to be too picky about where we get the training data from

What can go wrong?

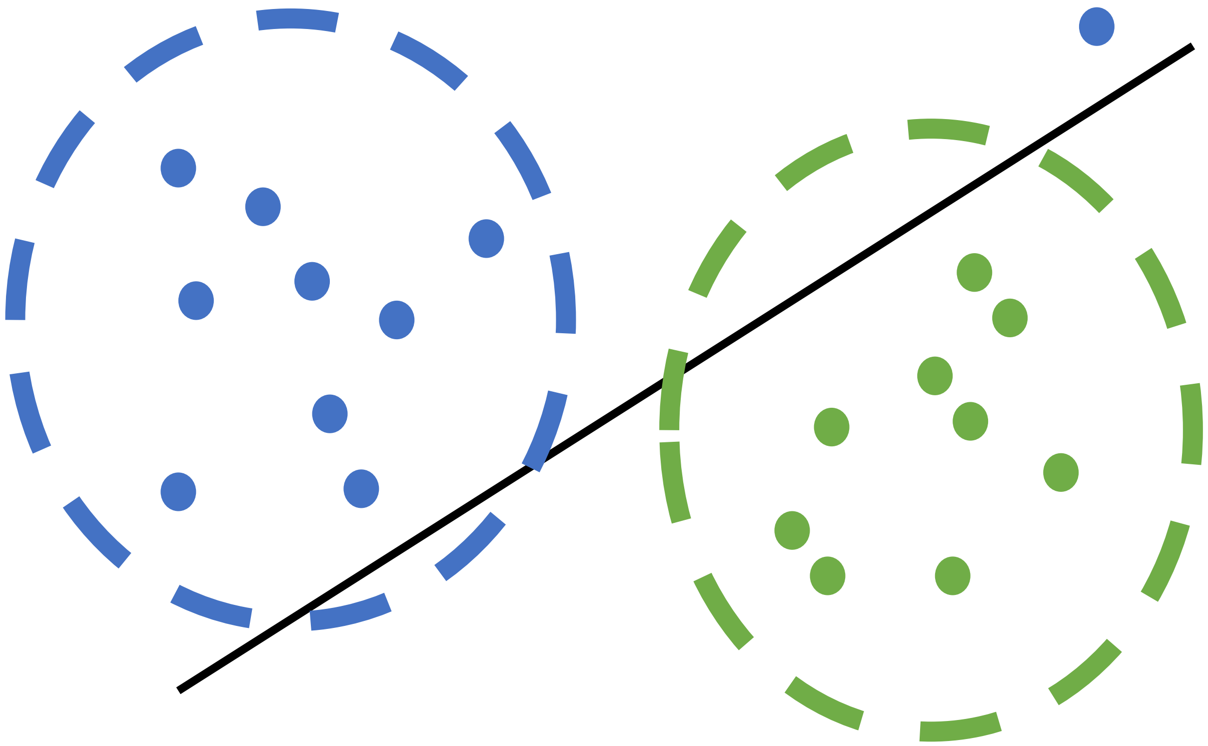
Data Poisoning

Goal: Maintain training accuracy but hamper generalization



Data Poisoning

Goal: Maintain training accuracy but hamper generalization



→ Fundamental problem in "classic" ML (robust statistics)

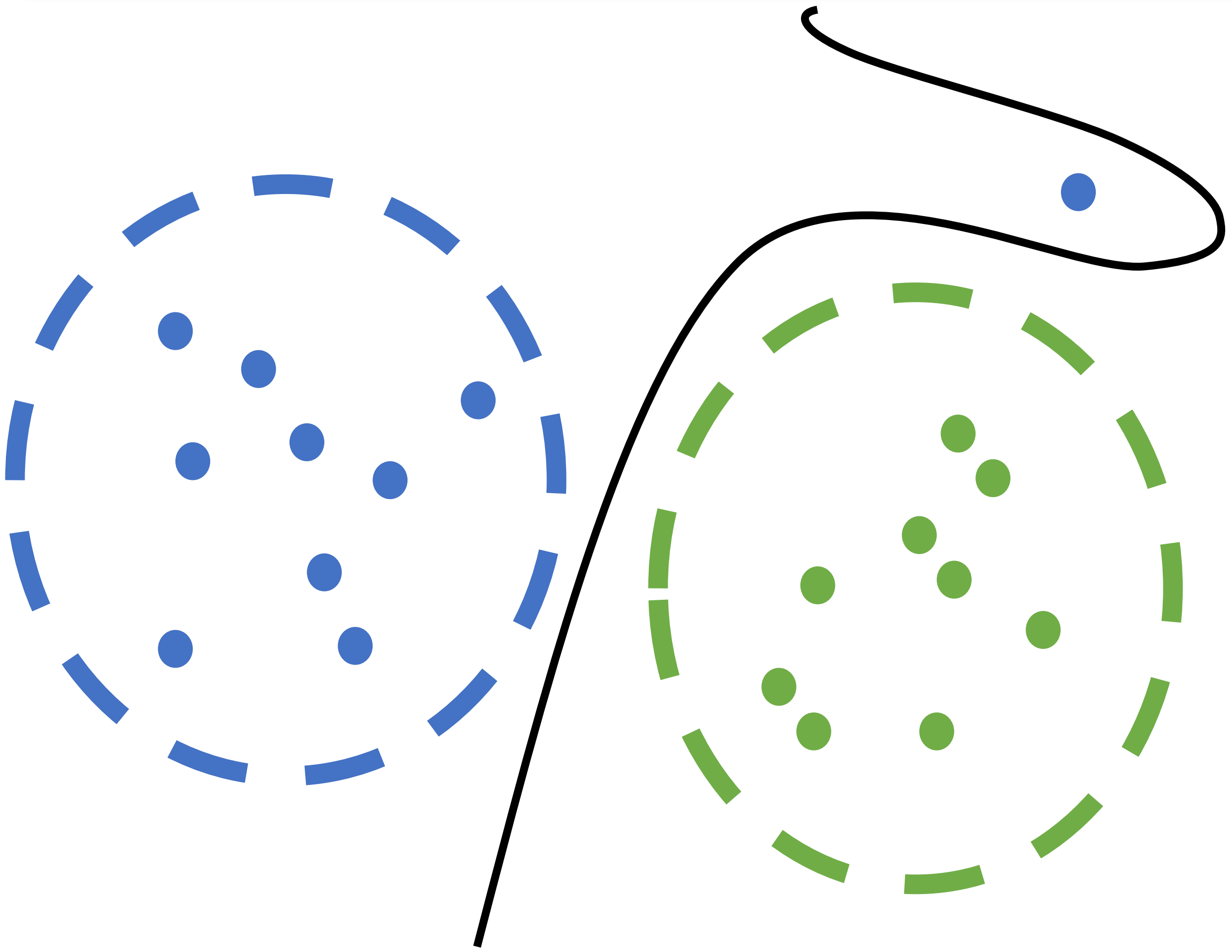
→ **But:** seems less so in deep learning

→ **Reason:** Memorization?

Data Poisoning

classification of **specific** inputs

Goal: Maintain training accuracy but hamper ~~generalization~~



→ Fundamental problem in “classic” ML (robust statistics)

→ **But:** seems less so in deep learning

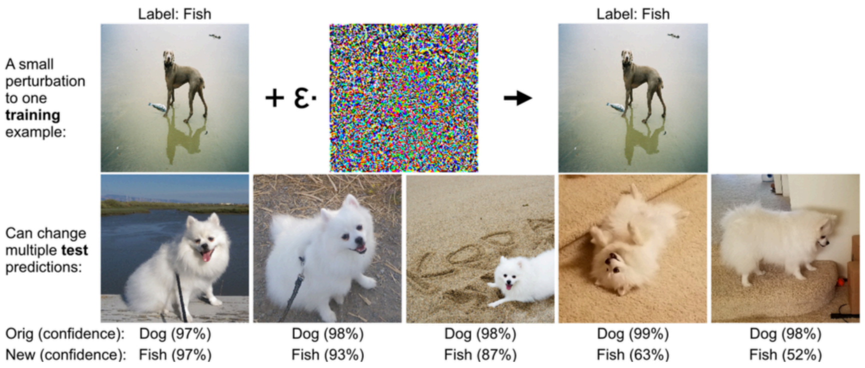
→ **Reason:** Memorization?

Is that it?

Data Poisoning

classification of **specific** inputs

Goal: Maintain training accuracy but hamper ~~generalization~~



[Koh Liang 2017]: Can manipulate **many** predictions with a **single** "poisoned" input

But: This gets (much) worse

[Gu Dolan-Gavitt Garg 2017][Turner Tsipras M 2018]: Can plant an **undetectable backdoor** that gives an almost **total** control over the model

Some defense mechanisms exist but not there (yet?) **[Tran Li M 2018]**

Is That It?

Microsoft Azure (Language Services)

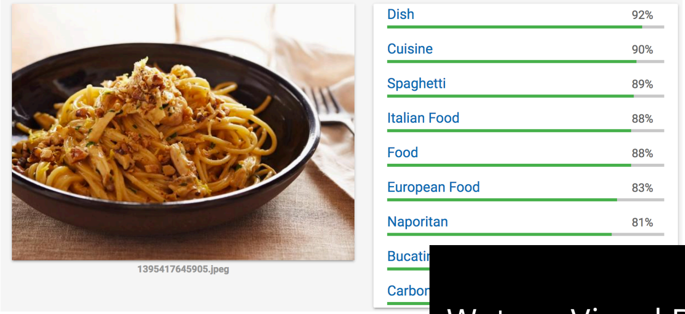
Language Understanding (LUIS)
Teach your apps to understand commands from your users
Try Language Understanding (LUIS) | Use with an Azure subscription

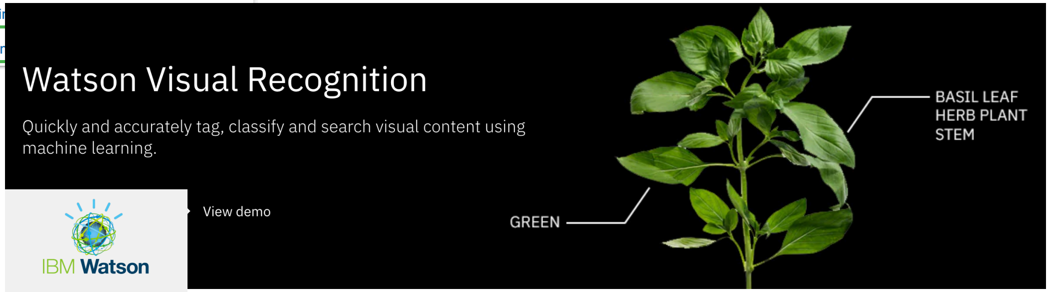
Text Analytics API
Easily evaluate sentiment and topics to understand what users want
Try Text Analytics API | Use with an Azure subscription

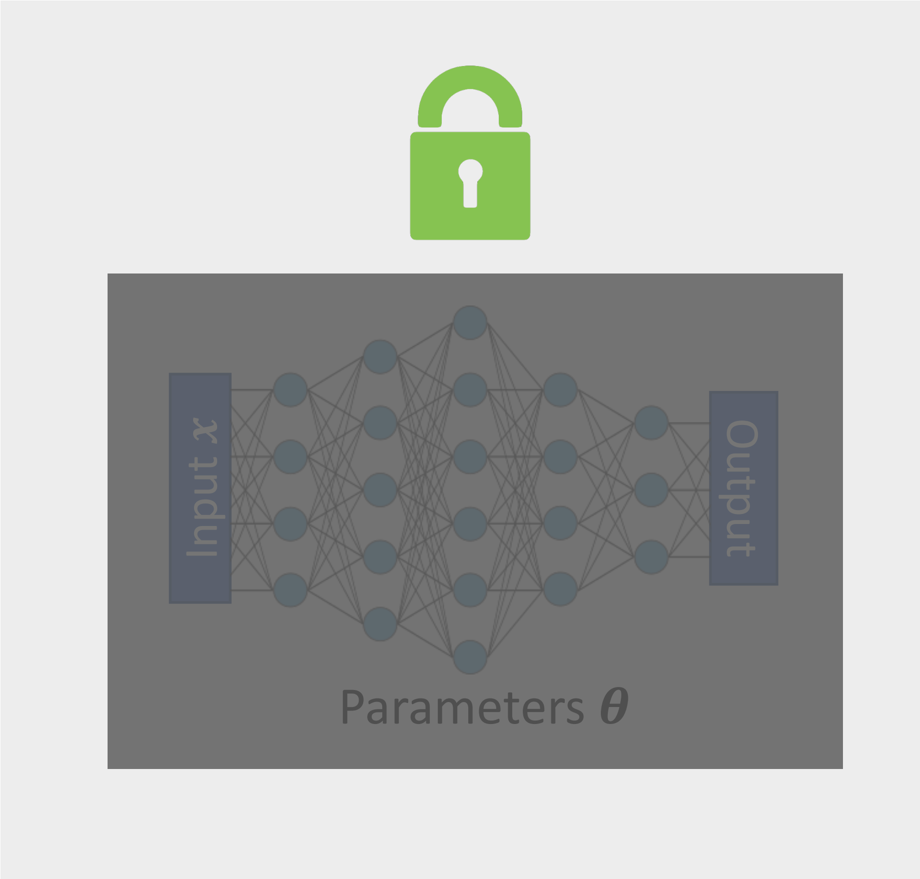
Bing Spell Check API
Detect and correct spelling mistakes in your app
Try Bing Spell Check API | Use with an Azure subscription

Translator Text API
Easily conduct machine translation with a simple REST API call
Use with an Azure subscription

Google Cloud Vision API





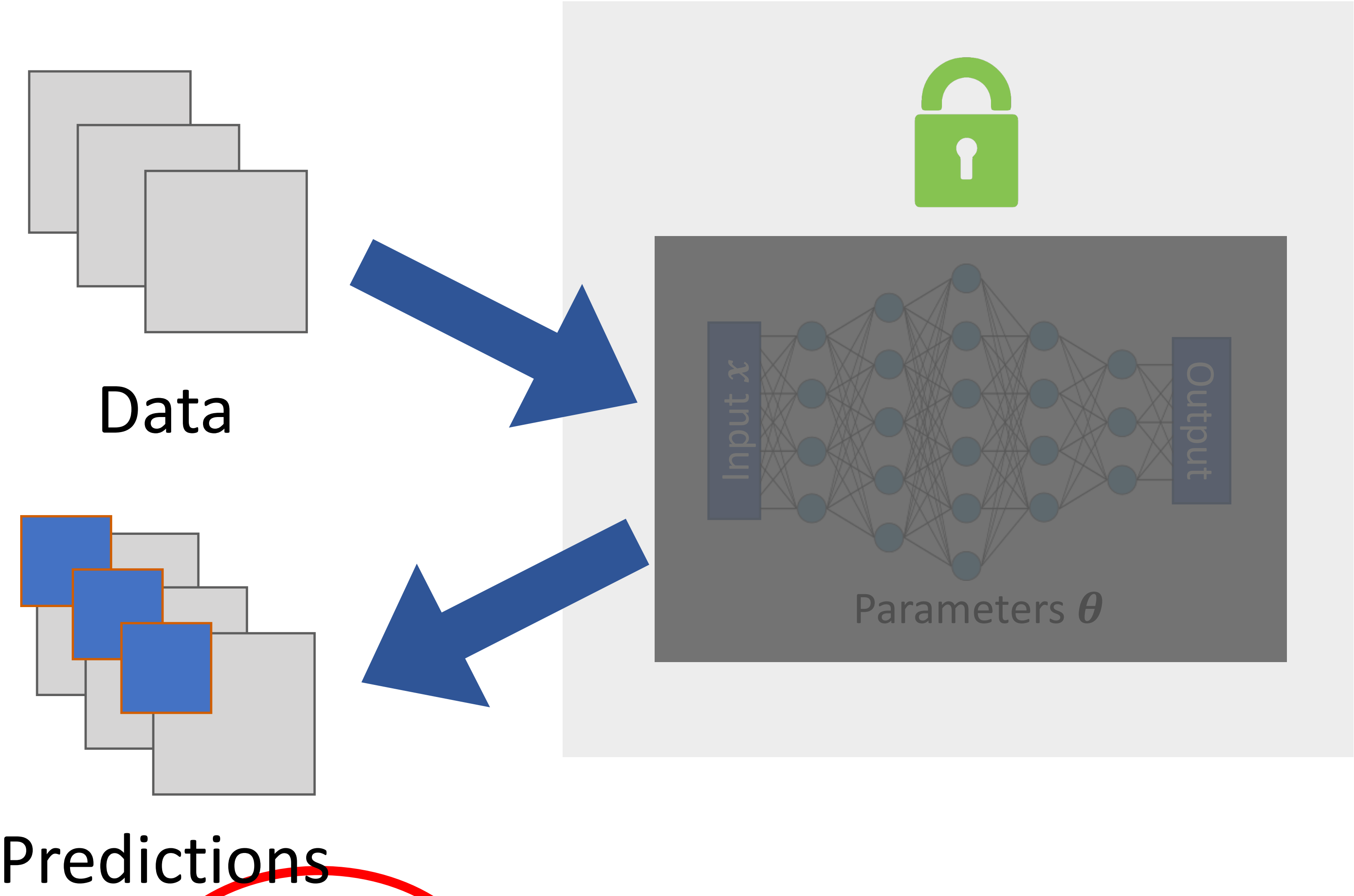


Training → Inference → Deployment

Is That It?

Does limited access give security?

In short: No



Training → Inference → Deployment

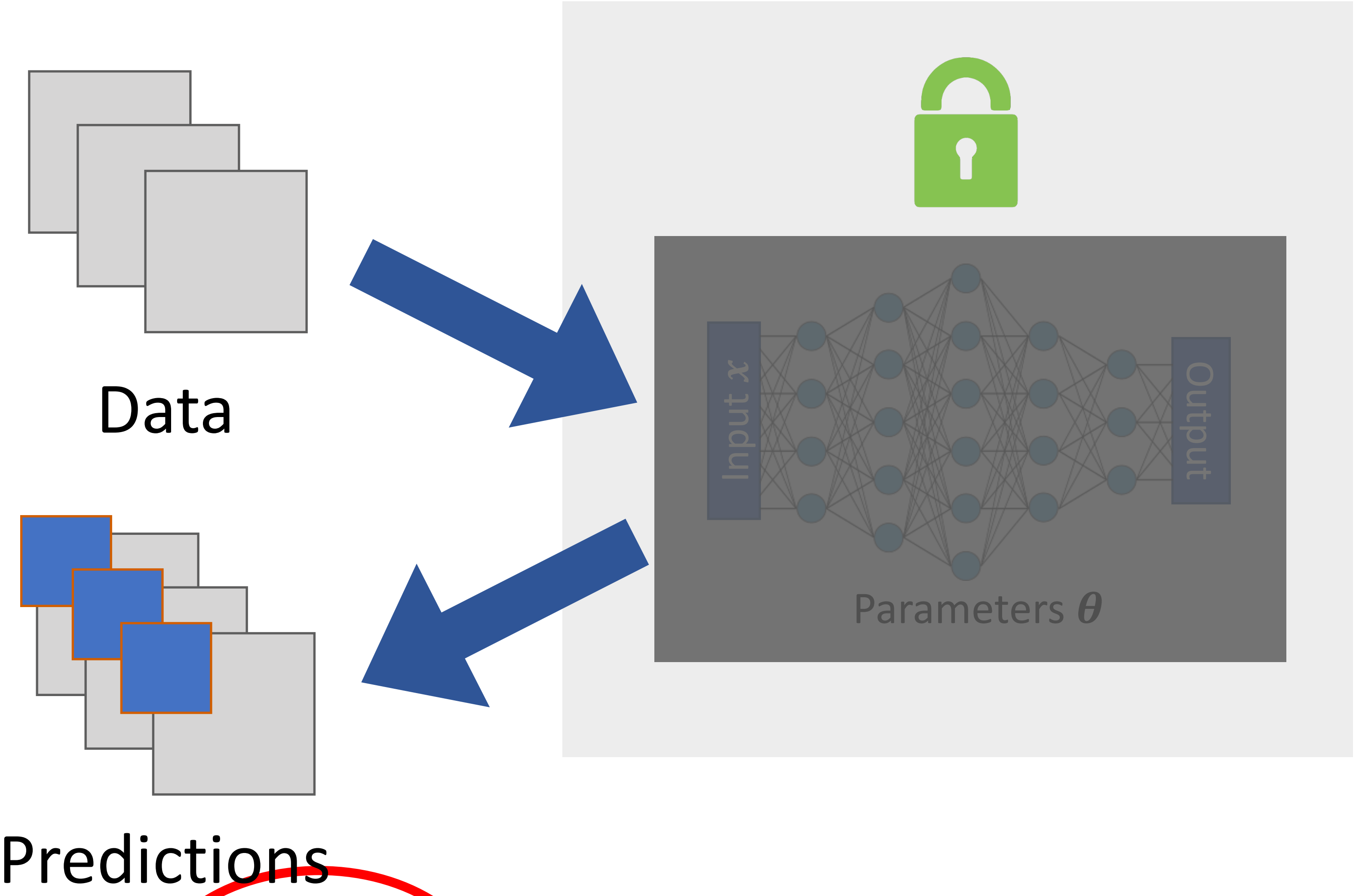
⚠ Black box attacks

Is That It?

Does limited access give security?

Model stealing: “Reverse engineer“ the model
[Tramer Zhang Juels Reiter Ristenpart 2016]

Black box attacks: Construct adv. examples from queries
[Chen Zhang Sharma Yi Hsieh 2017][Bhagoji He Li Song 2017][Ilyas Engstrom Athalye Lin 2017][Brendel Rauber Bethge 2017][Cheng Le Chen Yi Zhang Hsieh 2018][Ilyas Engstrom M 2018]



Training → Inference → Deployment

Black box attacks

Three commandments of Secure/Safe ML

I. Thou shall not train on data you don't fully trust

(because of data poisoning)

II. Thou shall not let anyone use your model (or observe its outputs) unless you completely trust them

(because of model stealing and black box attacks)

III. Thou shall not fully trust the predictions of your model

(because of adversarial examples)

Are we doomed?

(Is ML inherently not reliable?)

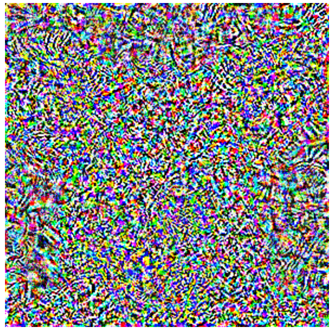
No: But we need to re-think how we do ML

(**Think:** adversarial aspects = stress-testing our solutions)

Towards Adversarially Robust Models

"pig" (91%)

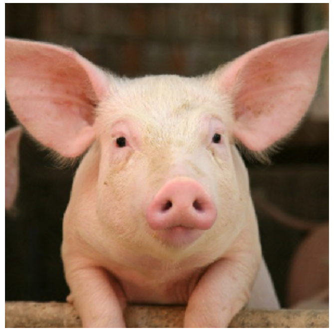
+ 0.005 x



=

"pig"

~~"airliner"~~ (99%)



Where Do Adversarial Examples Come From?

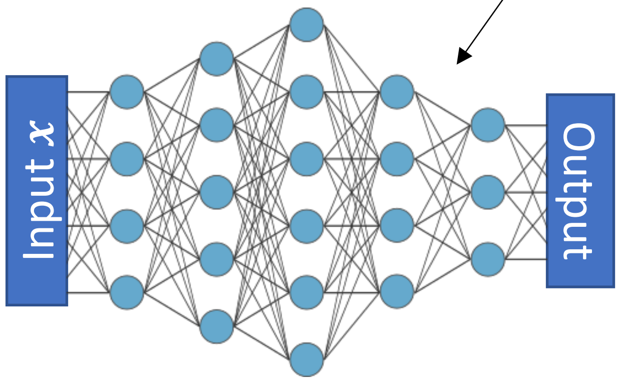
To get an adv. example

~~**Goal of training:**~~

Model Parameters　Input　Correct Label

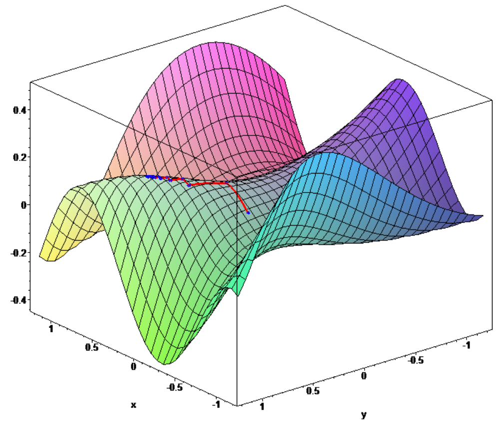
$$min_{\theta}\ loss(\theta, x, y)$$

Differentiable



Parameters $\boldsymbol{\theta}$

Can use gradient descent method to find good θ



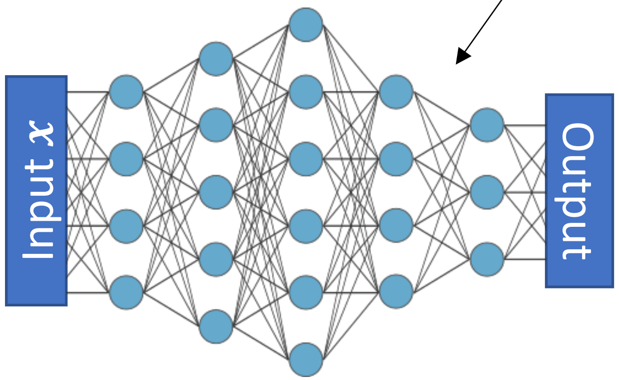
Where Do Adversarial Examples Come From?

To get an adv. example

~~**Goal of training:**~~

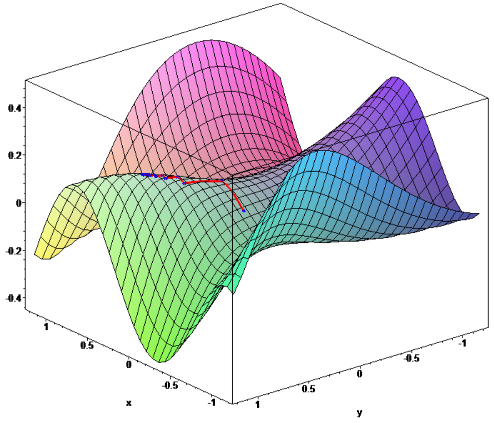
$$loss(\theta, x + \delta, y)$$

Differentiable



Parameters $\boldsymbol{\theta}$

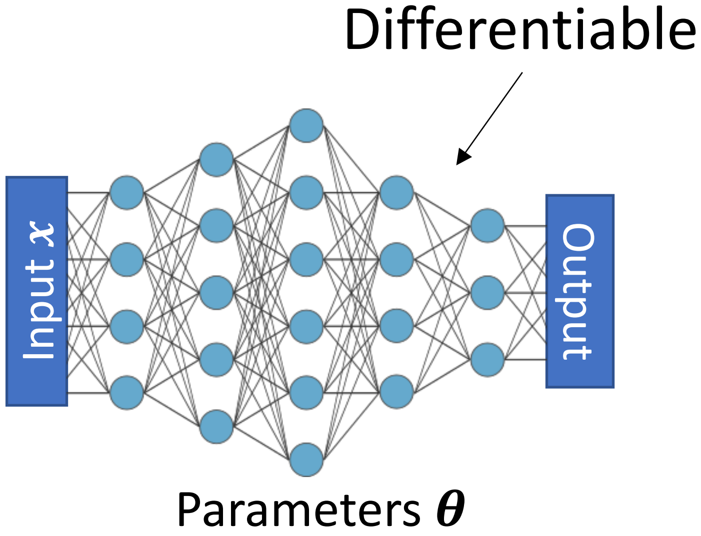
Can use gradient descent method to find good θ



Where Do Adversarial Examples Come From?

To get an adv. example

~~**Goal of training:**~~



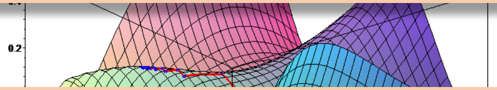
$$max_{\delta}\ loss(\theta, x + \delta, y)$$

Which δ are allowed?

Examples: δ that is small wrt

- ℓ_p-norm
- Rotation and/or translation
- VGG feature perturbation
- (add the perturbation you need here)

Can use gradient descent



This choice is important (but we put it aside)

In any case: We have to confront (small) ℓ_p-norm perturbations

Towards ML Models that Are Adv. Robust

[M Makelov Schmidt Tsipras Vladu 2018]

Key observation: Lack of adv. robustness is **NOT** at odds with what we currently want our ML models to achieve

~~Standard~~ generalization: $\mathbb{E}_{(x,y)\sim D}\left[loss(\theta, x, y)\right]$

Adversarially robust

But: Adversarial noise is a "needle in a haystack"

Towards ML Models that Are Adv. Robust

[M Makelov Schmidt Tsipras Vladu 2018]

Key observation: Lack of adv. robustness is **NOT** at odds with what we currently want our ML models to achieve

~~Standard~~ generalization: $\mathbb{E}_{(x,y)\sim D}\left[\max_{\boldsymbol{\delta}\in\boldsymbol{\Delta}} loss(\theta, x+\boldsymbol{\delta}, y)\right]$

Adversarially robust

But: Adversarial noise is a "needle in a haystack"

Towards ML Models that Are Adv. Robust

[M Makelov Schmidt Tsipras Vladu 2018]

Resulting training primitive:

$$\min_{\theta} \max_{\delta \in \Delta} \; loss(\theta, x + \boldsymbol{\delta}, y)$$

Finding a robust model

Finding a "bad" perturbation

To improve the model: Train on **perturbed** inputs
(aka as "adversarial training" **[Goodfellow Shlens Szegedy '15]**)

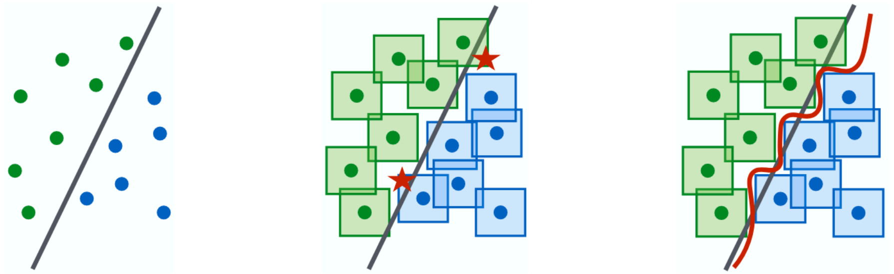
Does this work?

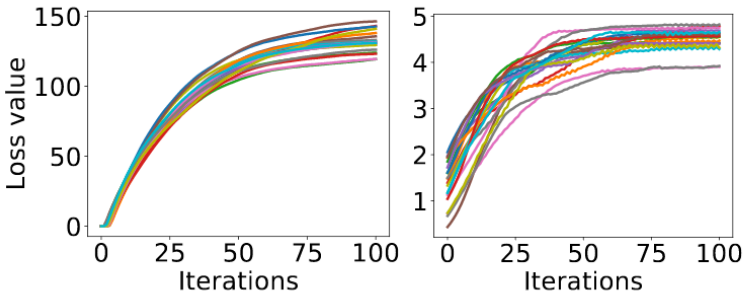
Yes! (In practice)
But certain care is required

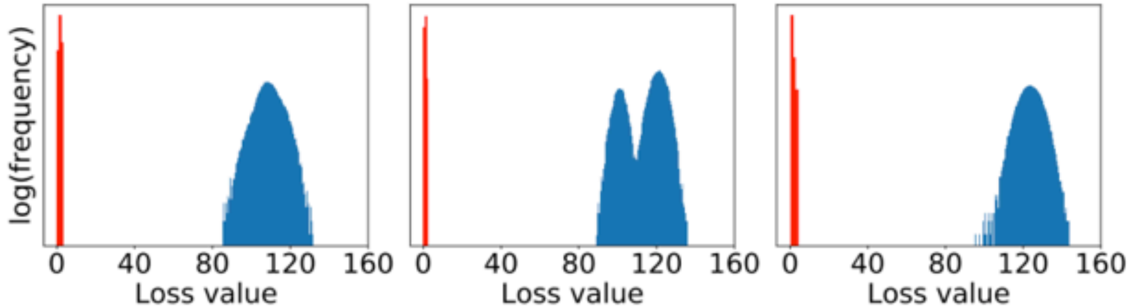
Key Components

→ Ability to **reliably** find "bad" perturbations

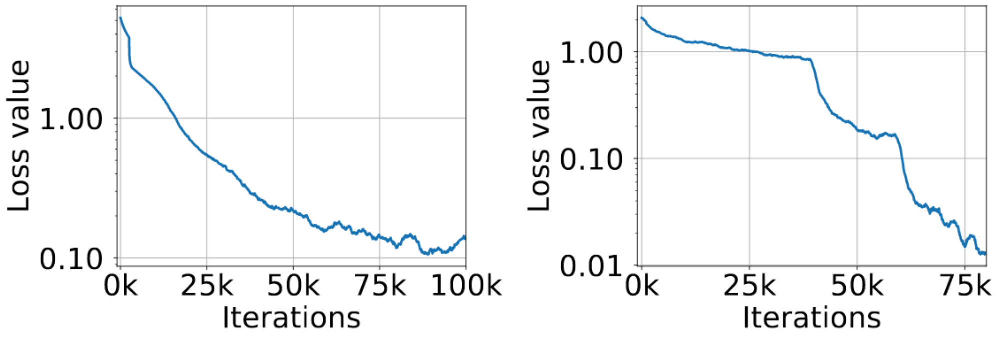
→ Sufficient model capacity



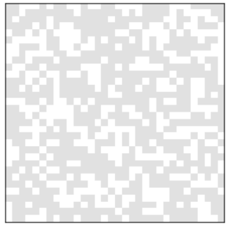
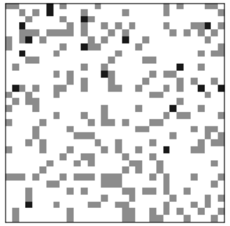
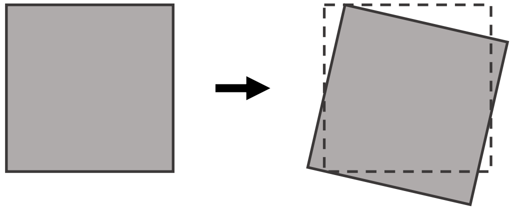




Result: Robustness increases steadily



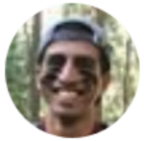
(a) MNIST (b) CIFAR10

	ℓ_∞-norm	ℓ_2-norm	Rotation + Translation
			
MNIST 	$\epsilon = 0.3/1$ 89%	$\epsilon = 2.5/1$ 66%	$\epsilon = \pm 3\ px, \pm 30°$ 98%
CIFAR-10 	$\epsilon = 8/255$ 47%	$\epsilon = 80/255$ 69%	$\epsilon = \pm 3\ px, \pm 30°$ 71% (+vote 82%)**
ImageNet 	$\epsilon = 16/255$ 4%	-	$\epsilon = \pm 30\ px, \pm 30°$ 53% (+vote 57%)**

**[Engstrom et al. 2018]

How do we know this really works?

→ Seems to be a recurring problem…



Anish Athalye @anishathalye · Feb 1
Defending against adversarial examples is still an unsolved problem; 7/8 defenses accepted to ICLR three days ago are already broken: github.com/anishathalye/o… (only the defense from @aleks_madry holds up to its claims: 47% accuracy on CIFAR-10)

Robustness by obscurity/complexity just does NOT work

→ Apply the standard security methodology:

- Evaluate with multiple **adaptive** attacks
- Use public security challenges

RobustML

(see robust-ml.org)

→ Use formal verification (where feasible):

- There is a steady progress on scaling these techniques up

[Katz et al '17, Wong Kolter '18, Tjeng et al '18, Dvijotham et al '18, Xiao Tjeng Shafiullah **M** '18]

Adversarial Robustness Beyond Security

ML via Adversarial Robustness Lens

Overarching question:
How does adv. robust ML differ from “standard” ML?

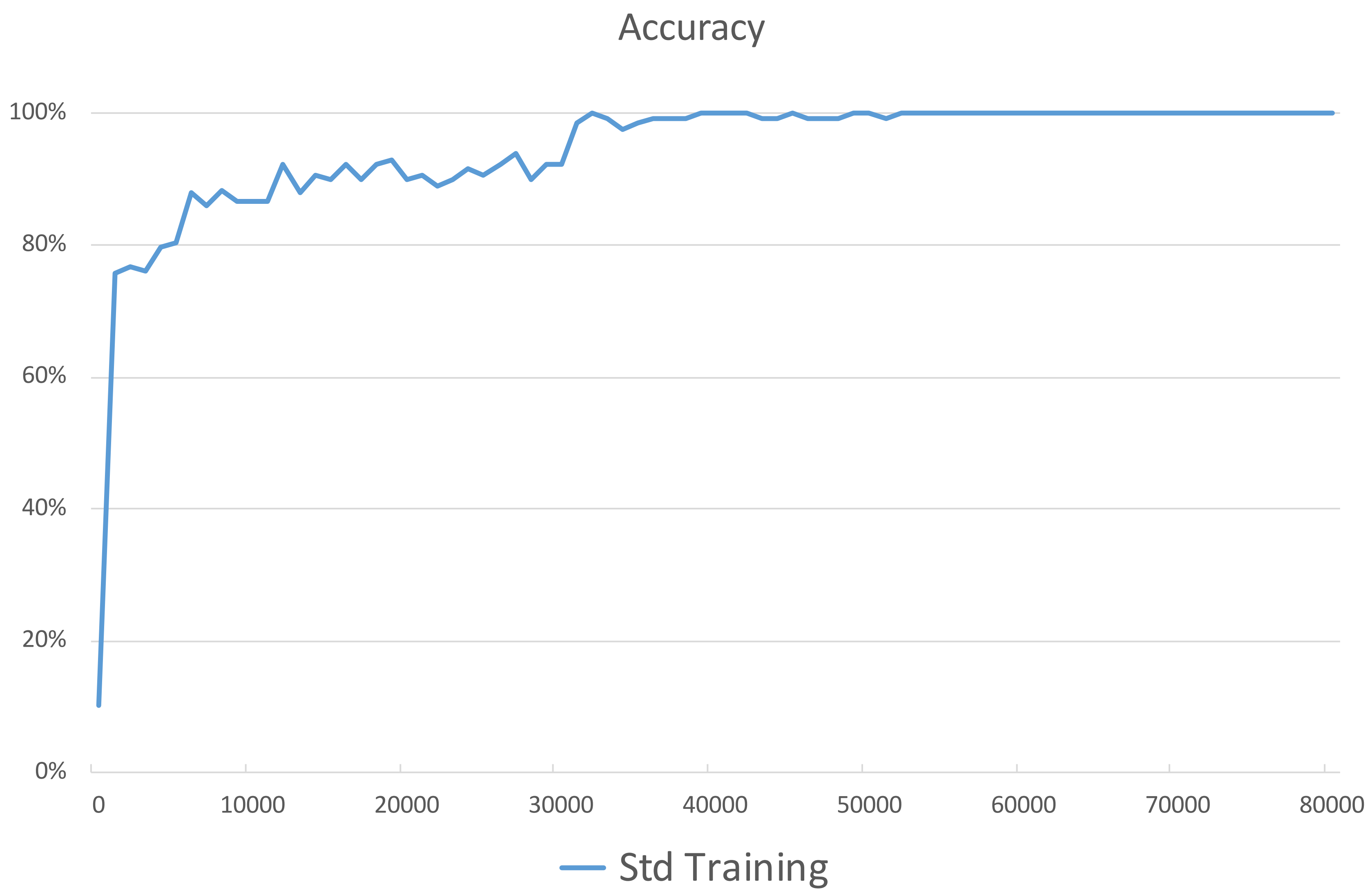
$$\mathbb{E}_{(x,y)\sim D}\left[loss(\theta, x, y)\right]$$

vs

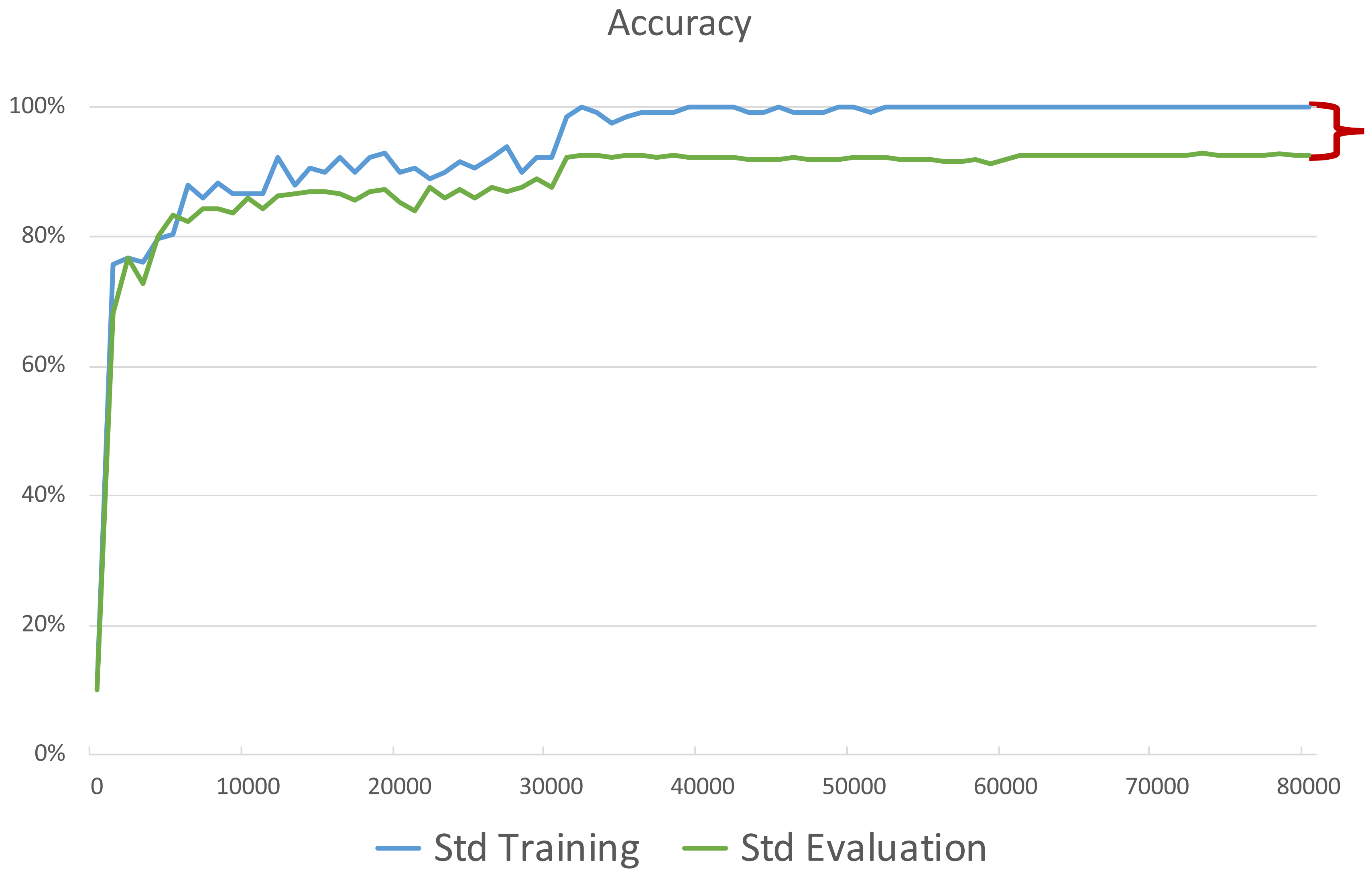
$$\mathbb{E}_{(x,y)\sim D}\left[\max_{\boldsymbol{\delta}\in\boldsymbol{\Delta}} loss(\theta, x + \boldsymbol{\delta}, y)\right]$$

(This goes **beyond** deep learning)

Do Robust Deep Networks Overfit?

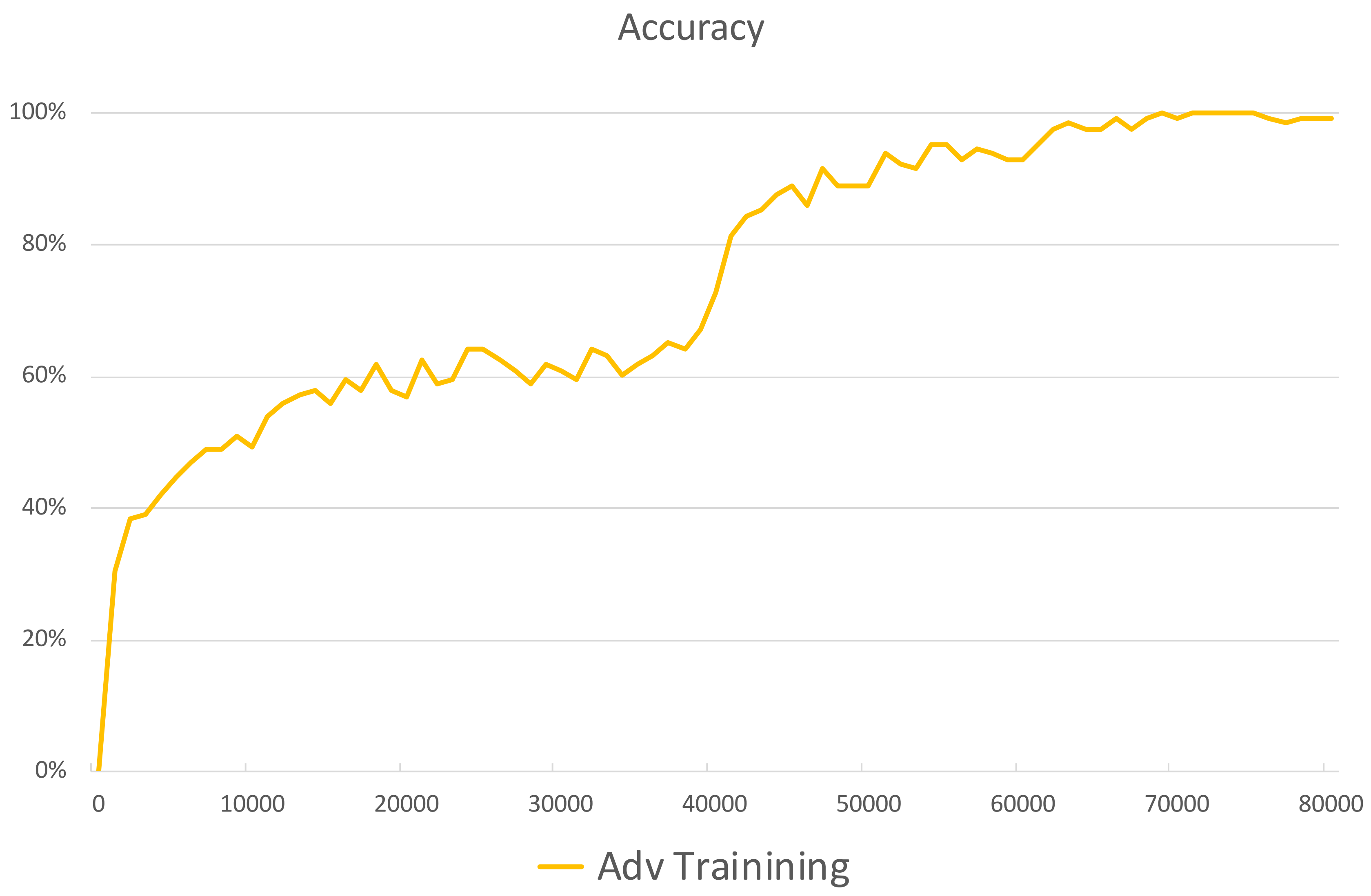


Do Robust Deep Networks Overfit?

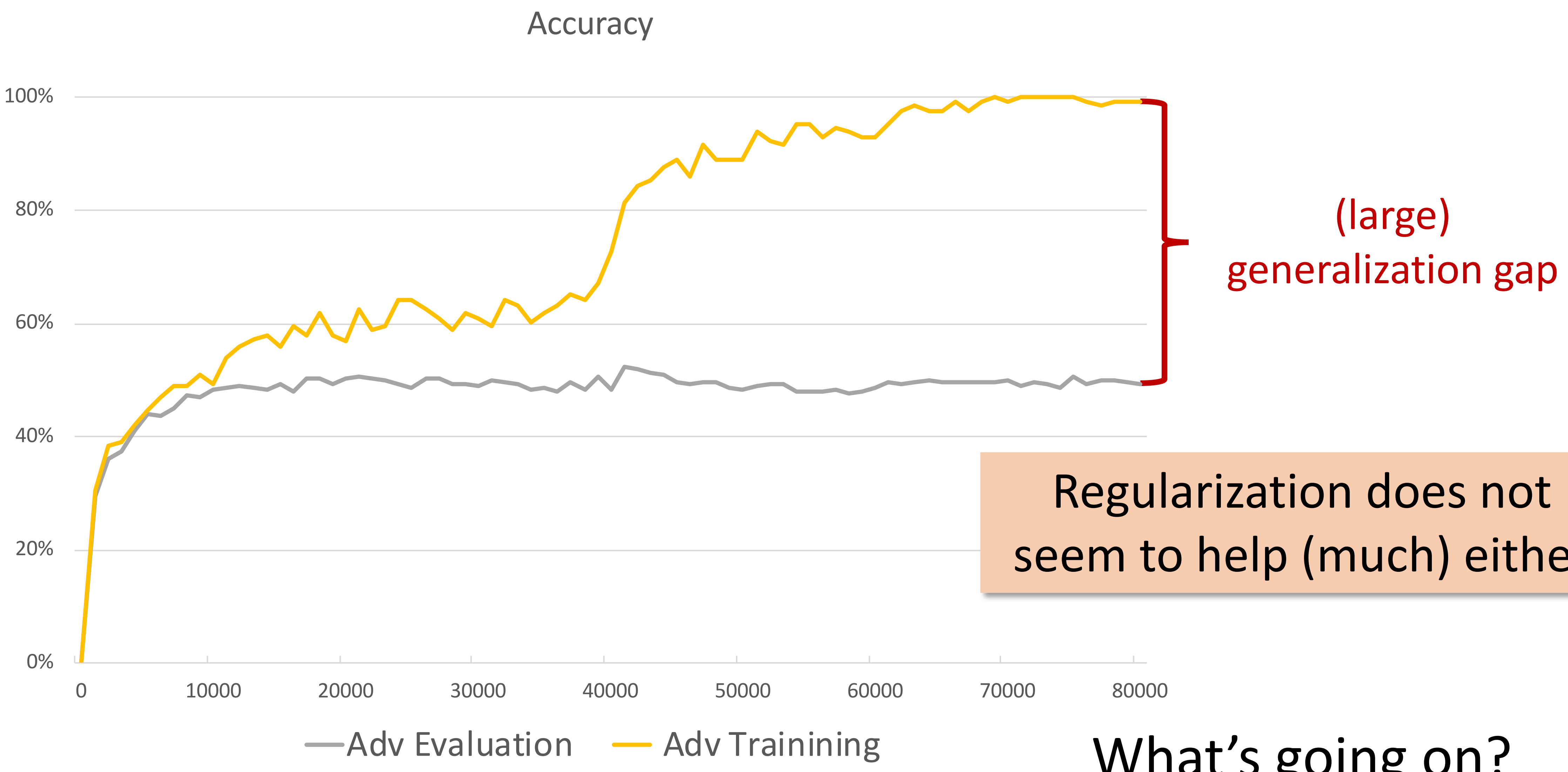


(small)
generalization gap

Do Robust Deep Networks Overfit?



Do Robust Deep Networks Overfit?



Regularization does not seem to help (much) either

What’s going on?

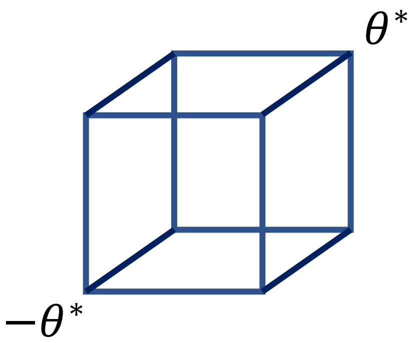
Adv. Robust Generalization Needs More Data

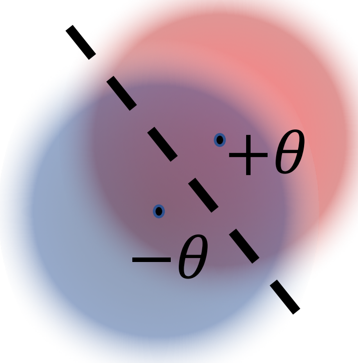
Theorem [Schmidt Santurkar Tsipras Talwar M 2018]:
Sample complexity of adv. robust generalization can be **significantly larger** than that of "standard" generalization

Specifically: There exists a **d**-dimensional distribution **D** s.t.:

→ A **single** sample is enough to get an **accurate** classifier (P[correct] > 0.99)

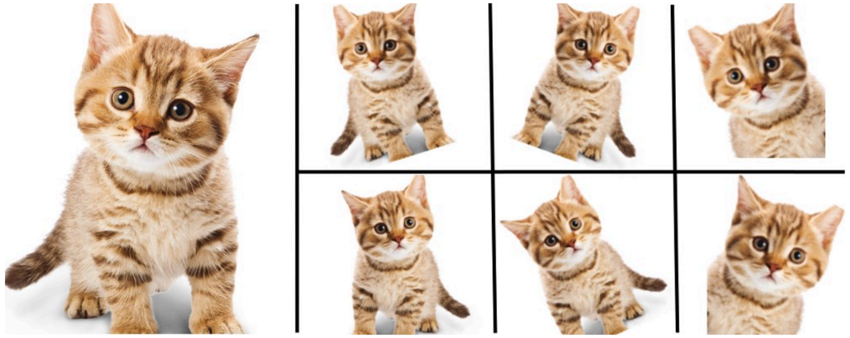
→ **But**: Need $\mathbf{\Omega}(\sqrt{\mathbf{d}})$ samples for better-than-chance **robust** classifier





Does Being Robust Help "Standard" Generalization?

Data augmentation: An effective technique to improve "standard" generalization



Adversarial training

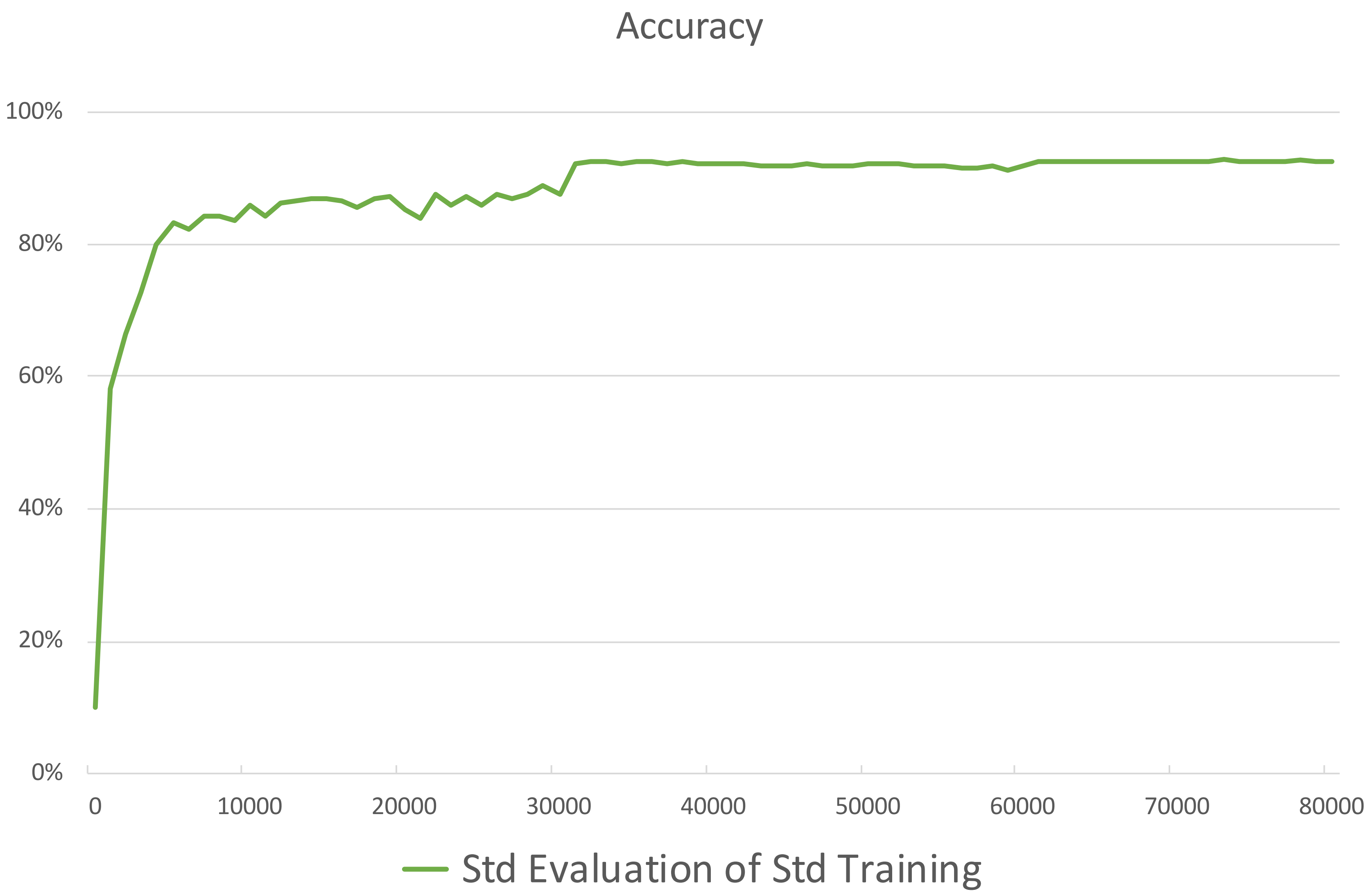
=

An "ultimate" version of data augmentation?

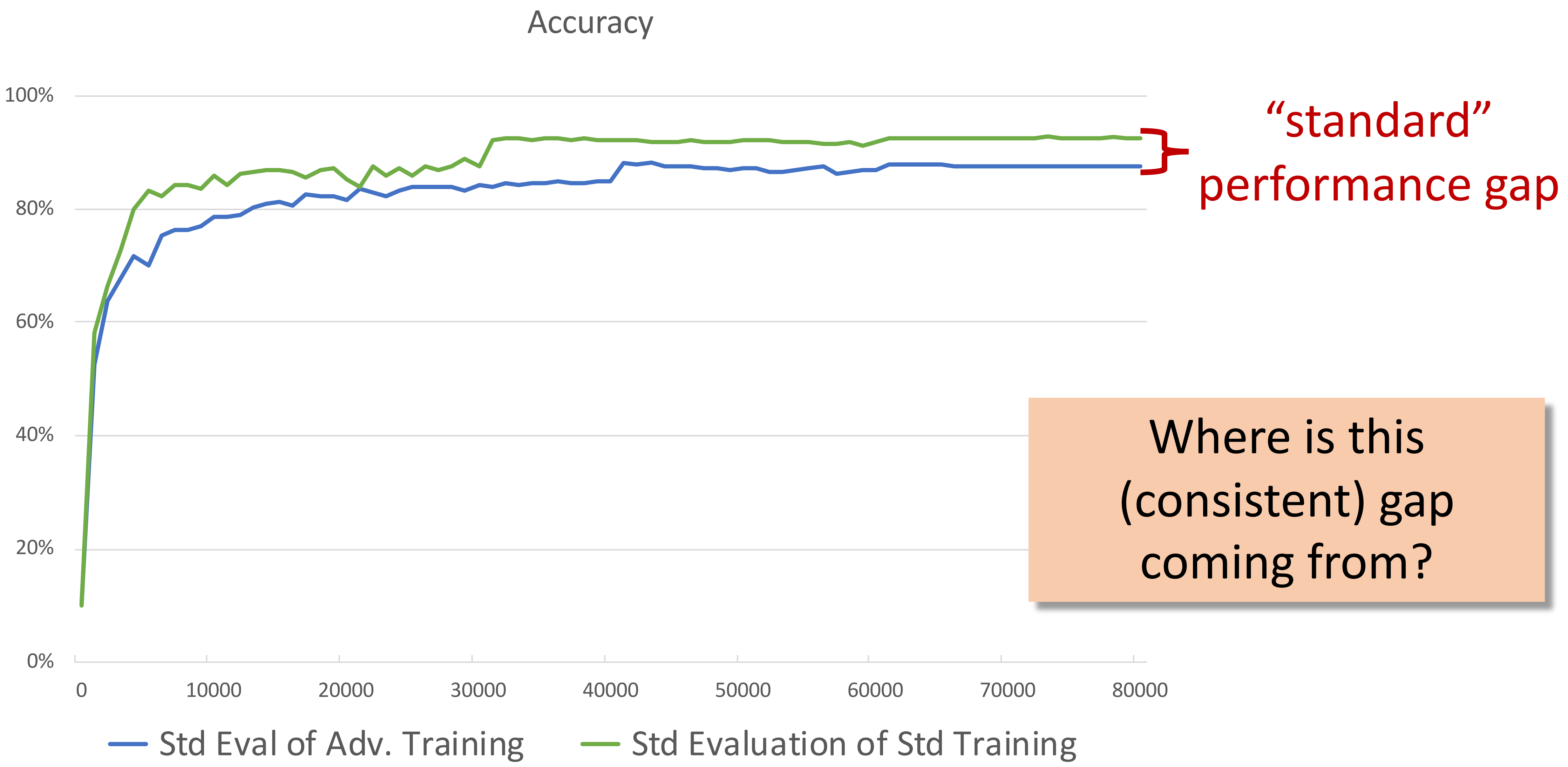
(since we train on the "most confusing" version of the training set)

Does adversarial training always improve "standard" generalization?

Does Being Robust Help “Standard” Generalization?



Does Being Robust Help “Standard” Generalization?



Where is this (consistent) gap coming from?

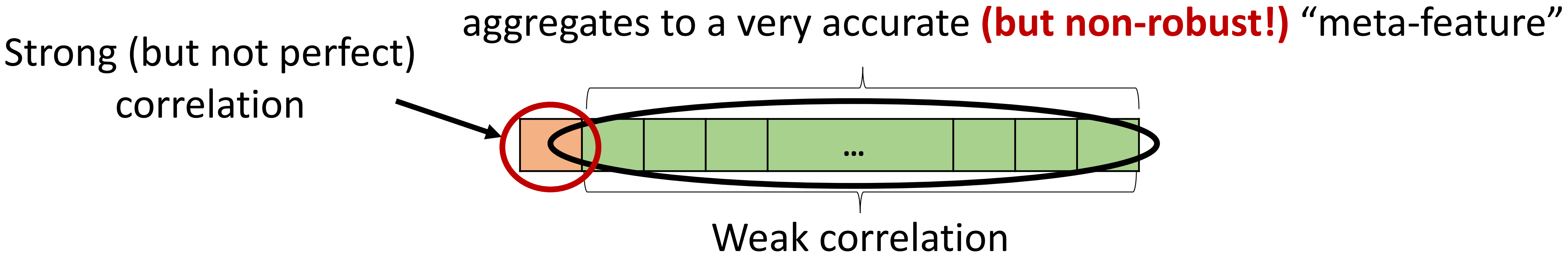
Does Being Robust Help "Standard" Generalization?

Theorem [Tsipras Santurkar Engstrom Turner M 2018]:
No "free lunch": can exist a trade-off between accuracy and robustness

Basic intuition:

→ In standard training, **all correlation is good correlation**

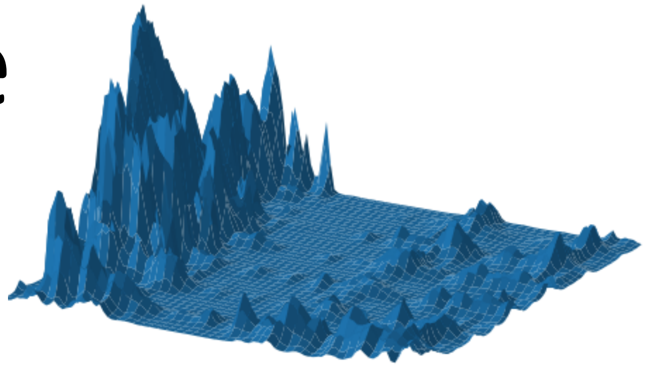
→ If we want robustness, **must avoid** weakly correlated features



Standard training: use all of features, maximize accuracy

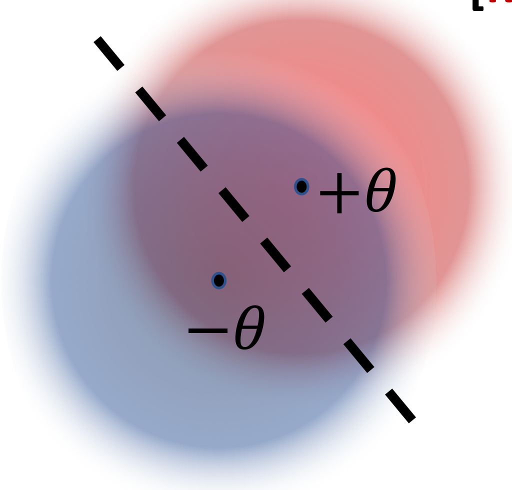
Adversarial training: use only single robust feature (**at the expense of accuracy**)

Adversarial Robustness is Not Free



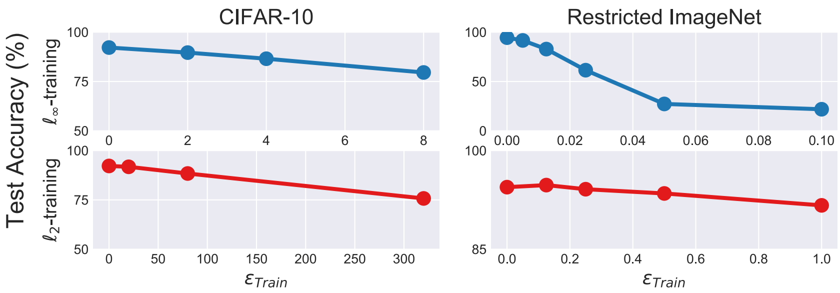
→ Optimization during training more difficult and models need to be larger

[M Makelov Schmidt Tsipras Vladu 2018]



→ More training data might be required

[Schmidt Santurkar Tsipras Talwar M 2018]



→ Might need to lose on "standard" measures of performance

[Tsipras Santurkar Engstrom Turner M 2018] (Also see: **[Bubeck Price Razenshteyn 2018]**)

But: "How"/"what" does not tell us "why"

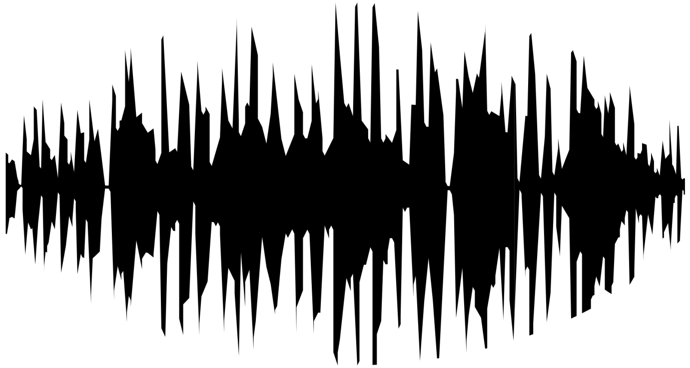
Why adversarial perturbations **exist**
(and **are so widespread**)?

Why these perturbations tend to **transfer**?

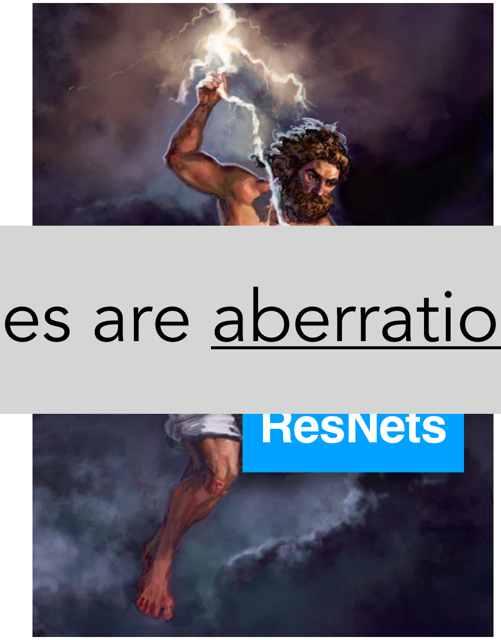
Why **robust training** works?

Why **randomized smoothing** works?

$$d \to \infty$$

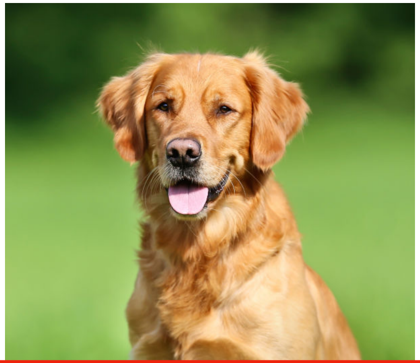
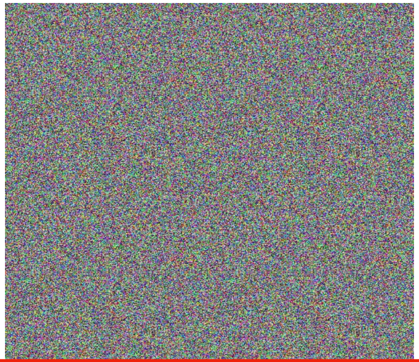
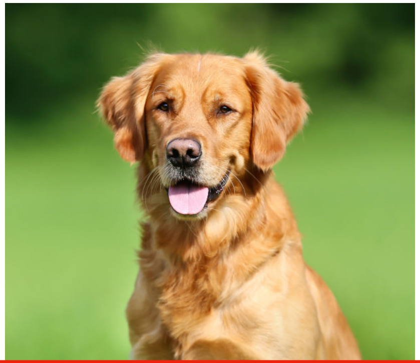


Why are our models brittle?



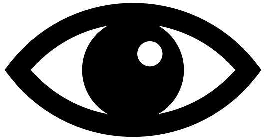
Unifying theme: Adversarial examples are <u>aberrations</u>

Why Are Adv. Perturbations Bad?

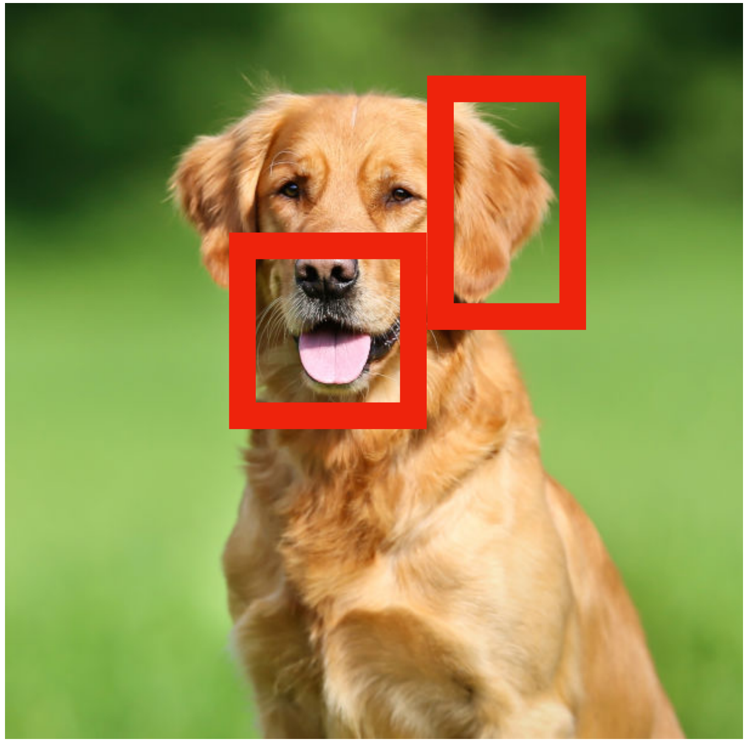
 +  = 

dog **meaningless perturbation** **cat**

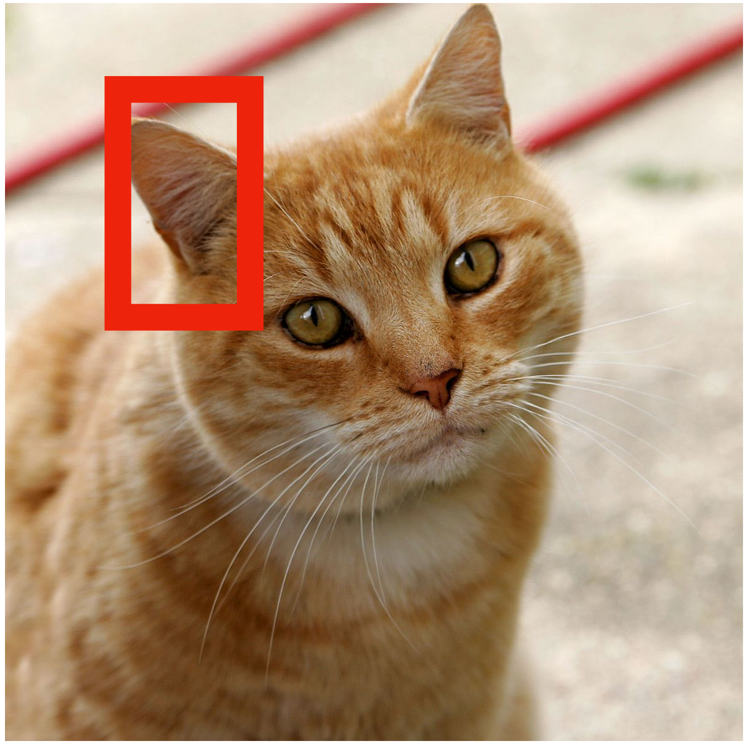
But: This is only a “human” perspective



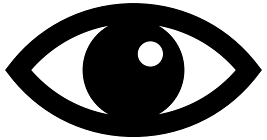
Human Perspective



dog



cat



ML Perspective

dog

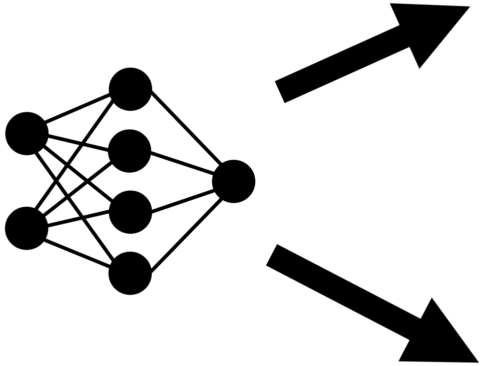


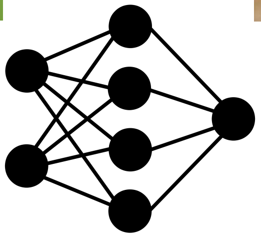
Image is meaningless

Classes are meaningless

Only goal:
Max (test) accuracy

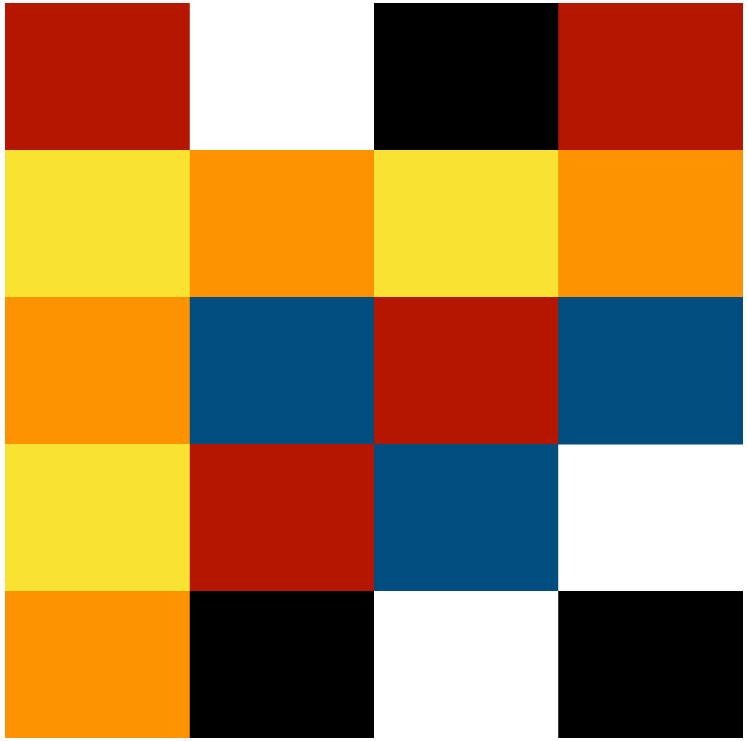
ML Perspective

dog

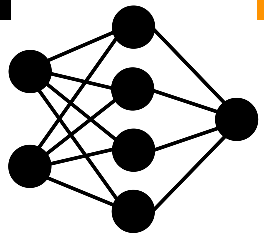


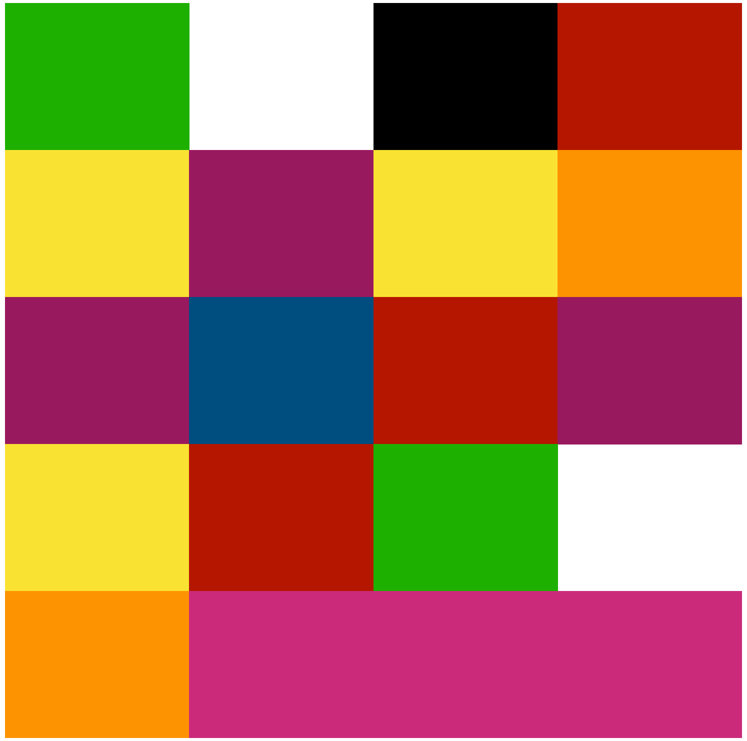
cat

ML Perspective



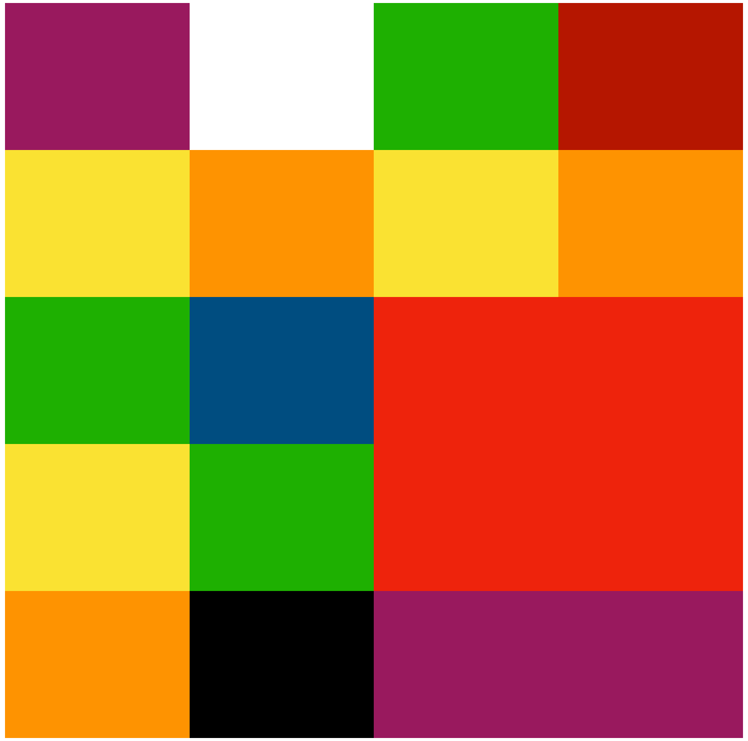
tap



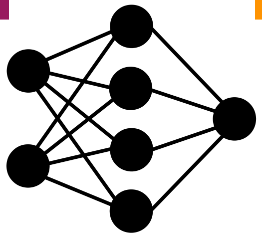


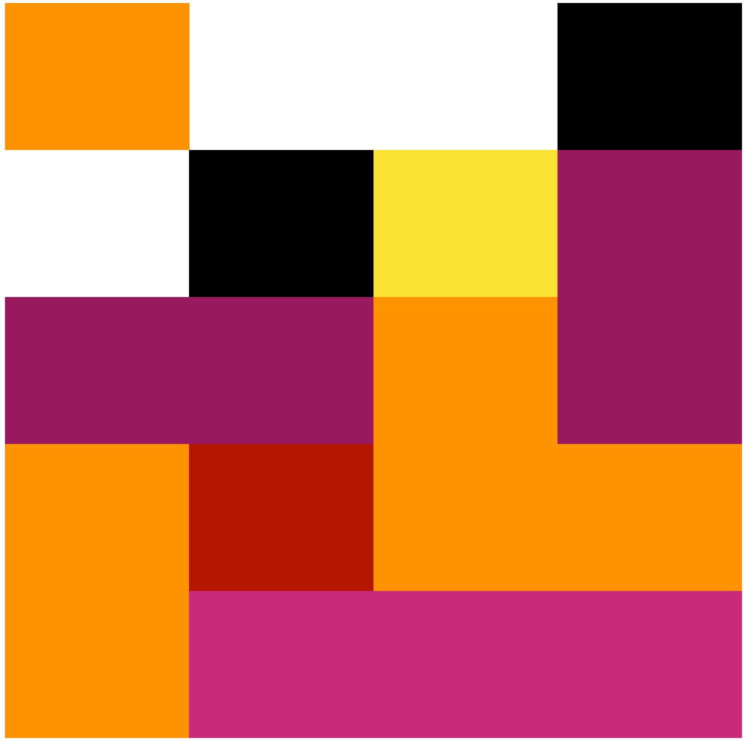
toc

ML Perspective



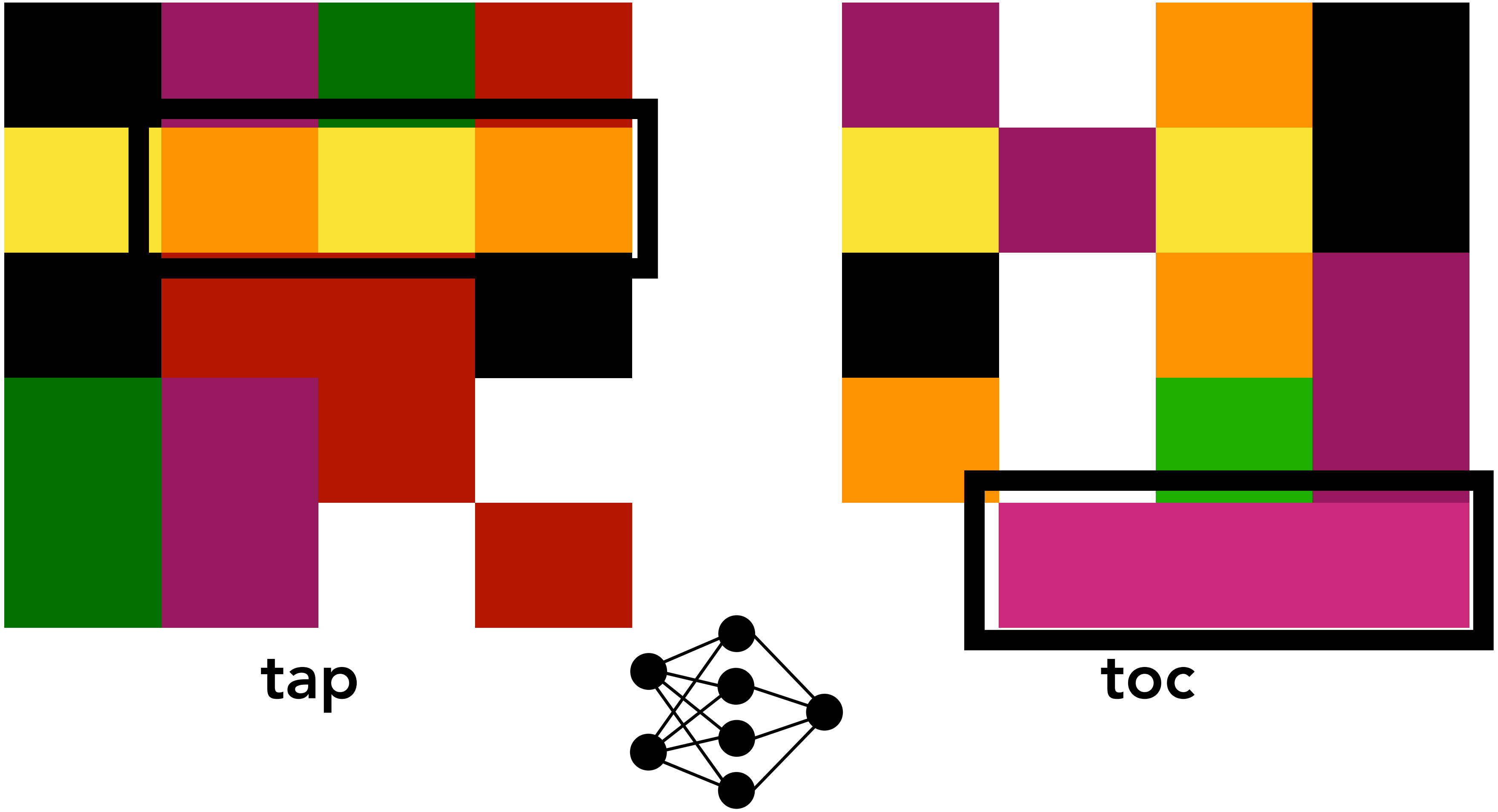
tap



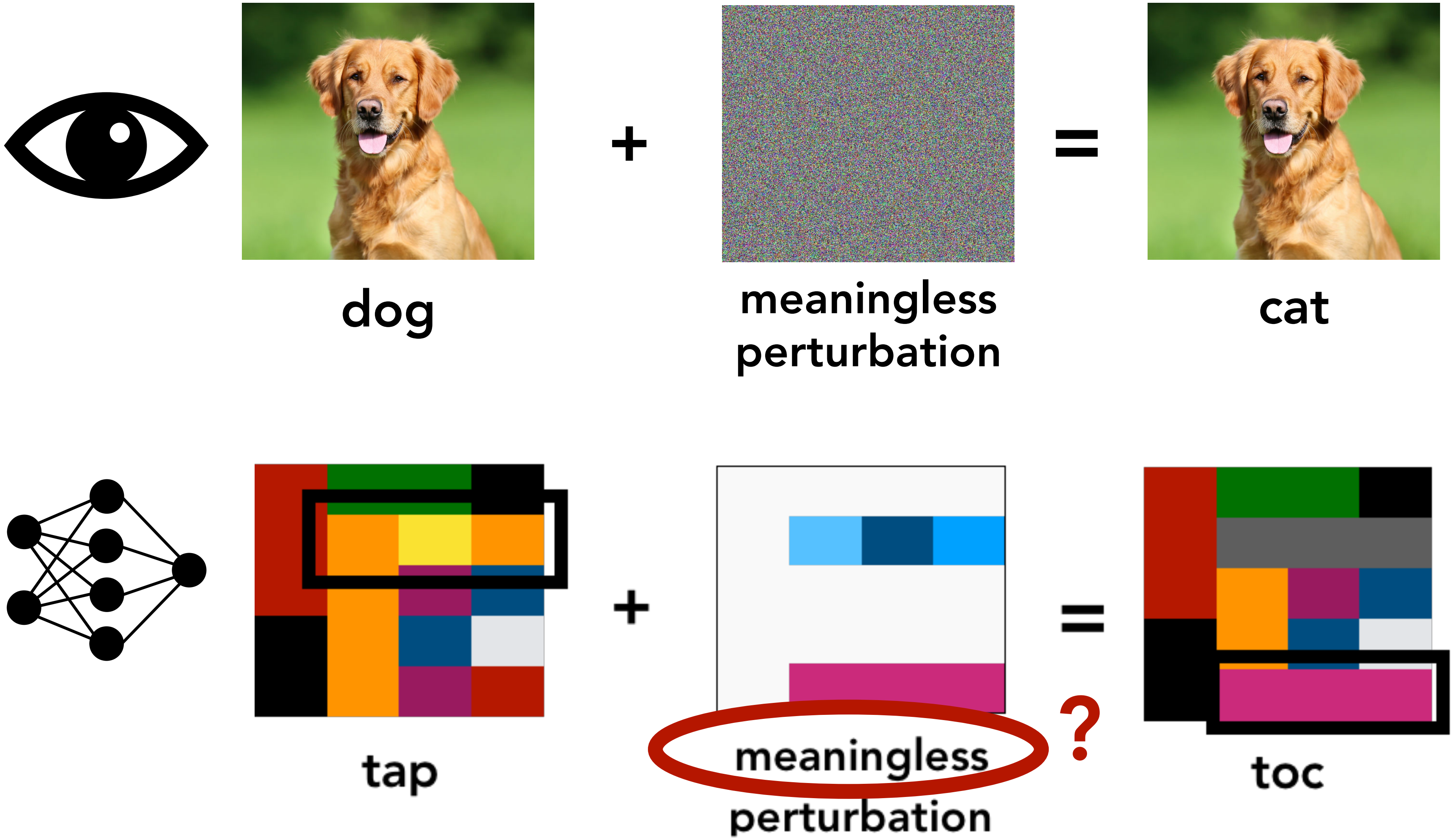


toc

ML Perspective



ML Perspective



Are adversarial perturbations just meaningless artifacts?

[Ilyas Santurkar Tsipras Engstrom Tran M '19]

A Simple Experiment

1. **Make adversarial example** towards the other class
2. **Relabel** the image as the target class
3. Train with **new** dataset but test on the <u>**original**</u> test set

A Simple Experiment

So: We train on a “totally mislabeled” dataset but expect performance on a “correct” dataset

What will happen?

A Simple Experiment

Result: We get a **nontrivial accuracy** on the **original** classification task

(For example, 78% on the CIFAR dog vs cat)

What's going on?

What if adversarial perturbations are **not** aberrations but **features**?

The Robust Features Model

Robust features
Correlated with label
even with adversary

Non-robust features
Correlated with label on average,
but can be flipped within, e.g., ℓ_2 ball

When maximizing (test) accuracy: All features are good

And: Non-robust features are often great!

That's why our models pick on them
(and **become vulnerable to adversarial perturbations**)

The Simple Experiment: A Second Look

Training set

New training set

All robust features are **misleading**

But: Non-robust features suffice for good generalization

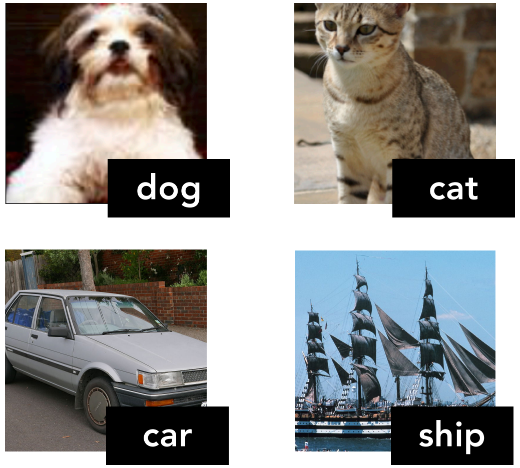
The Simple Experiment: A Second Look

New training set

cat

Train

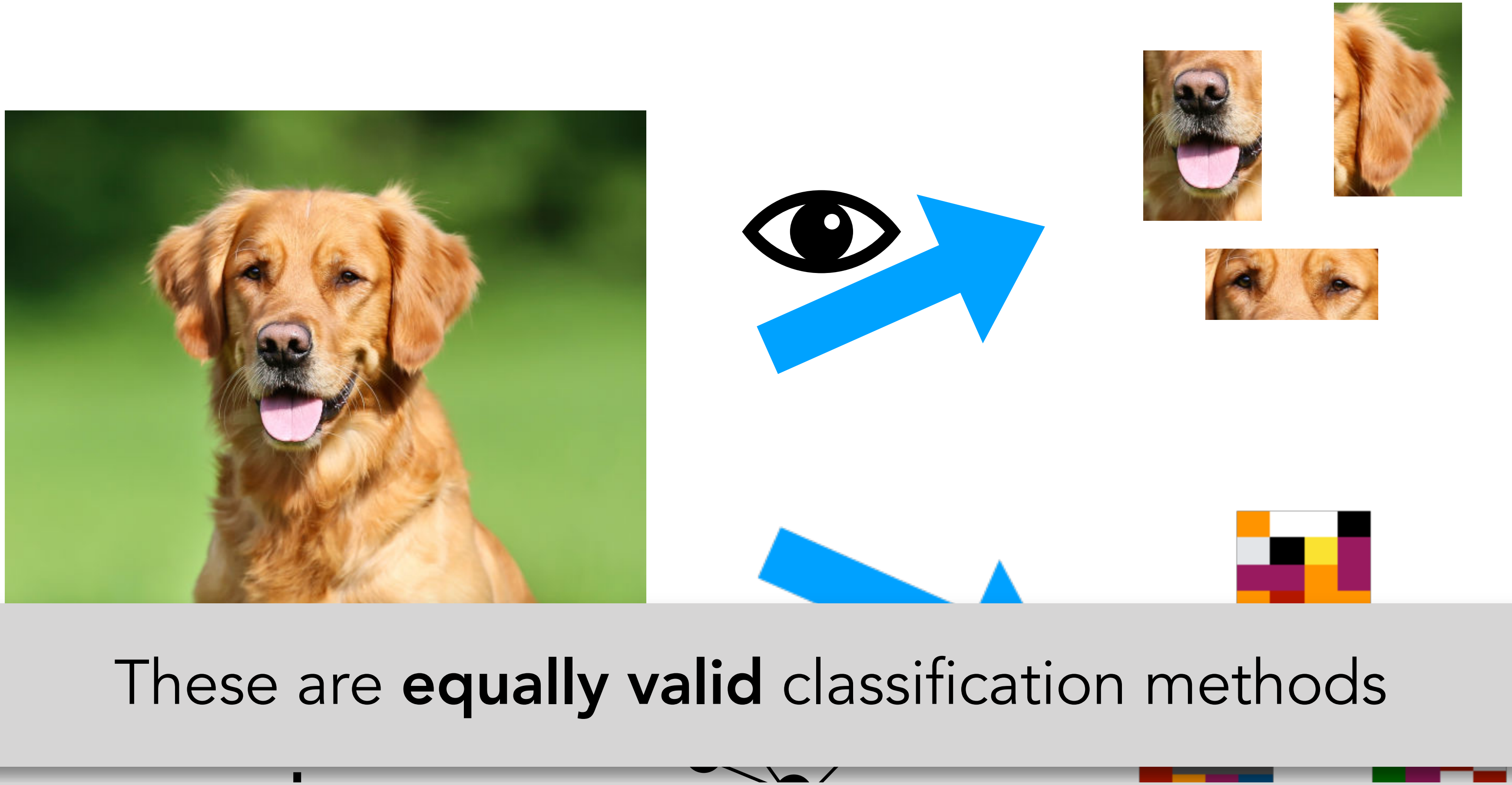
Test set



Robust features: dog
Non-robust features: cat

Good test accuracy on original test set

Human vs ML Model Priors



These are **equally valid** classification methods

No reason to expect our models to use the first one

Human vs ML Model Priors

Adversarial examples are a **human** phenomenon

No hope for interpretable models without intervention **at training time** (instead of post-hoc)

Need **additional restrictions (priors)** on what features models should use to make predictions

What now?

A (new) perspective on
adversarial robustness

(Provides insights into other questions too)

New capability: Robustification

Training set

frog

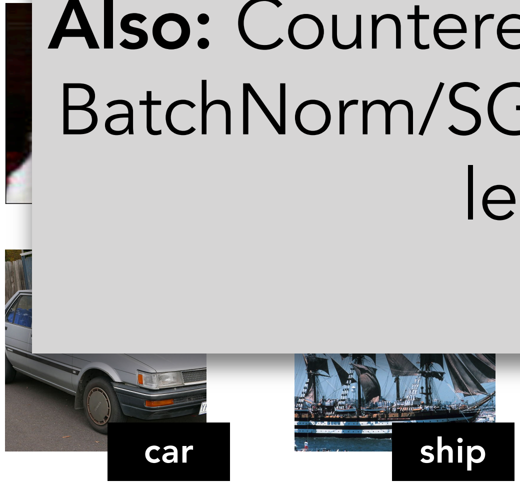
Restrict to features
of robust model

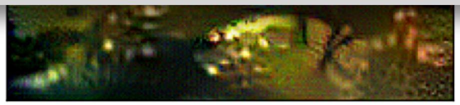
New training set

"robustified" frog

New capability: Robustification

Also: Counterexample to any statement that "Training with BatchNorm/SGD/ResNets/overparameterization/etc. <u>alone</u> leads to adversarial vulnerability"





"robustified" frog

We get both standard and **robust** accuracy

So: It really is about features

Some Direct Consequences

Transferability: Features = property of **datasets** (not models)

Effectiveness of Robust Training:
Makes features that are non-robust w.r.t. Δ **useless**

Effectiveness of Randomized Smoothing:
Overwhelms non-robust (w.r.t. Δ) features with noise

Robustness and Data Efficiency

Robust models can only leverage **robust** features

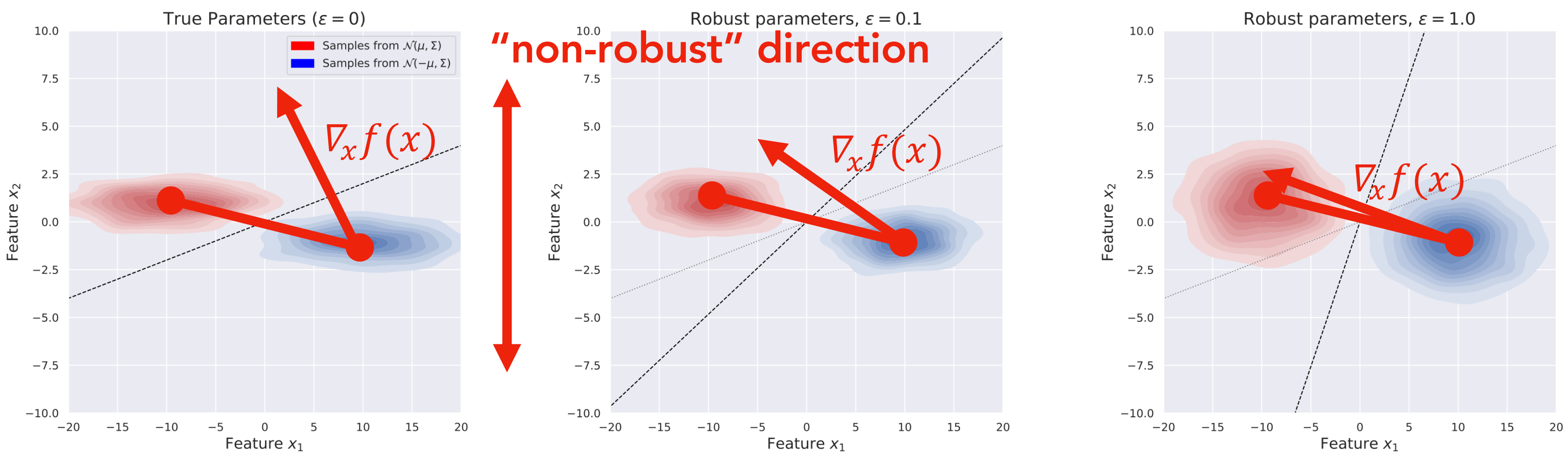
(Even though non-robust features **do** help with generalization)

$\longrightarrow$ Need **more data** to get a given (robust) accuracy
(vide **[Schmidt Santurkar Tsipras Talwar M '18]**)

$\longrightarrow$ Will get a **lower standard accuracy**
(vide **[Tsipras Santurkar Engstrom Turner M '18]**)

But: Is leveraging non-robust features good?

A Simple Theoretical Setting:
Robust Max Likelihood Gaussian Classification



Things to observe:

→ Non-robust features are needed to get better standard accuracy but lead to vulnerability

→ Gradient directions in robust models are more aligned with the"semantic"/human-preferred direction (will get back to this)

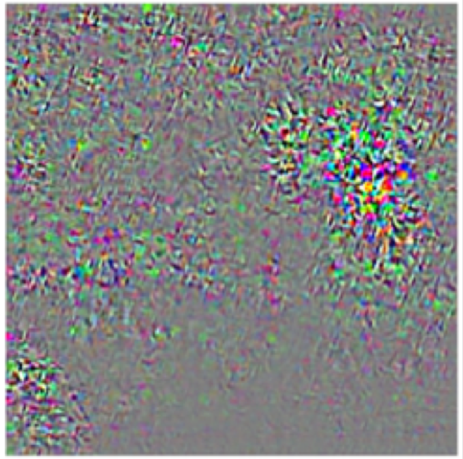
(Exact theorems in the paper)

What if we **prevent** models from learning **non-robust** features?

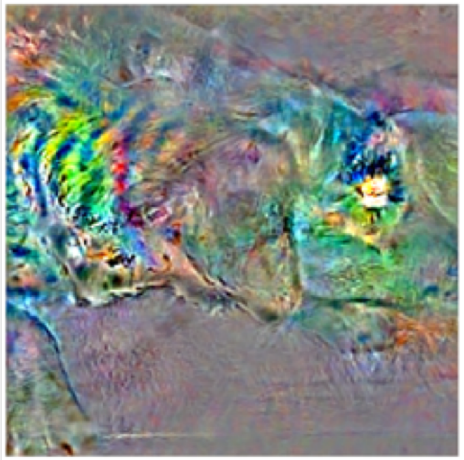
[Tsipras Santurkar Engstrom Turner M ’18]
[Engstrom Ilyas Santurkar Tsipras Tran M ’19]

Robustness → Perception Alignment

Input



Gradient of
standard model

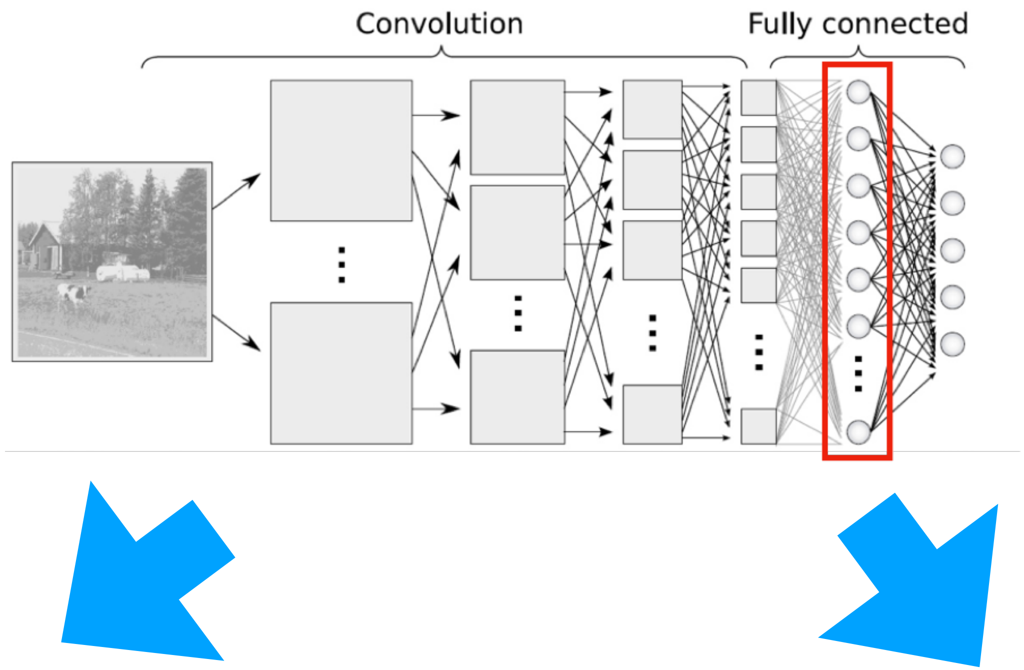


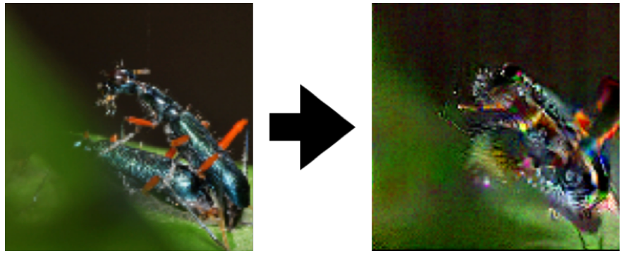
Gradient of
adv. robust model

Models become more (human) perception aligned

→ Robustness acts as a **prior** for "meaningful" features

Robustness → Better Representations





Standard Representation

Robust Representation

Robustness → Better Representations

Robust representations enable a wide range of feature manipulations/visualizations in a **simple** way

Feature manipulations/visualization are not new

[Mahendran Vedaldi '15][Simonyan Vedaldi Zisserman '14][Øygard '15]
[Nguyen Yosinski Clune '15][Yosinski Clune Nguyen Fuchs Lipson '15]
[Mordvintsev Olah Tyka '15][Nguyen Dosovitskiy Yosinski Brox Clune '16]
[Radford Metz Chintala '16][Larsen Sønderby Larochelle Winther '16][Tyka '16]

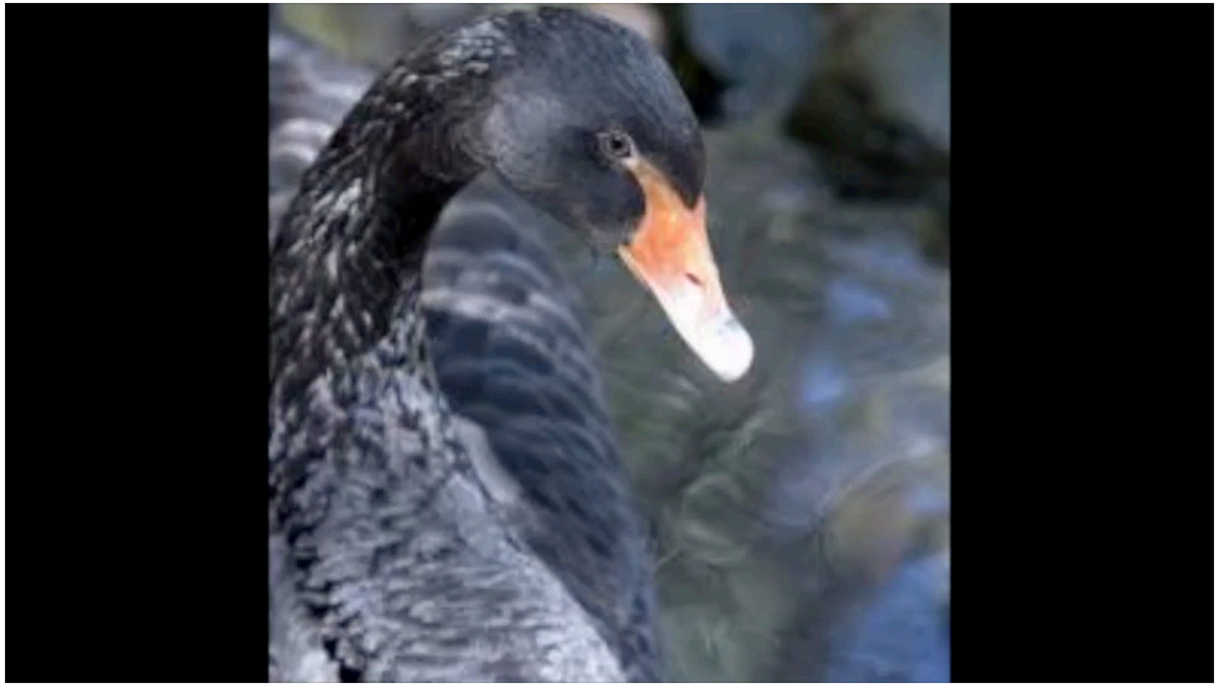
But here:

[Brock et al '18] + [Isola '18]

→ Everything boils down to simple optimization primitives

→ No priors, no regularization, no post-processing
(and thus we are fully faithful to the model)

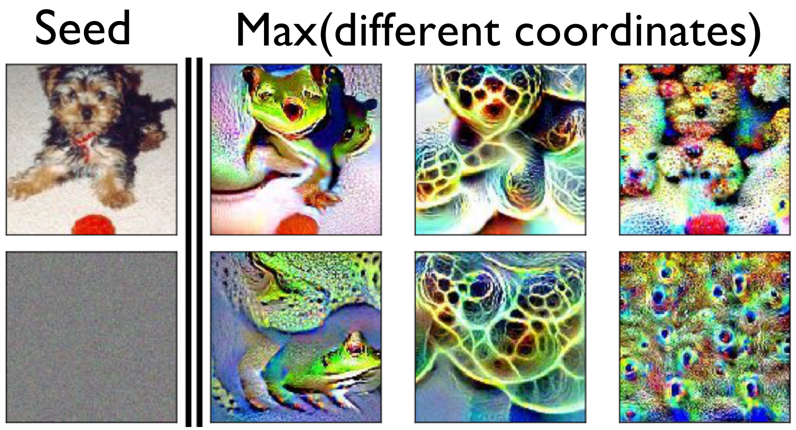
Robustness → Better Representations

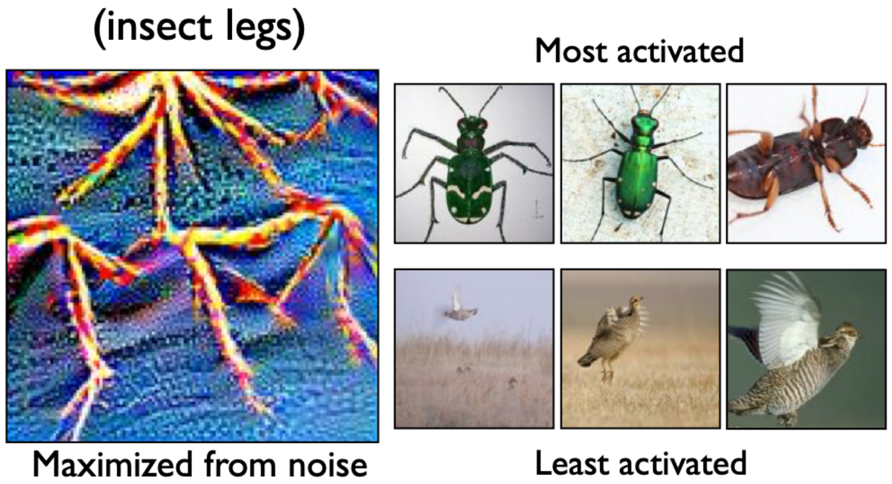


Interpolation between **any** two inputs

(Can do it for **any** two inputs)

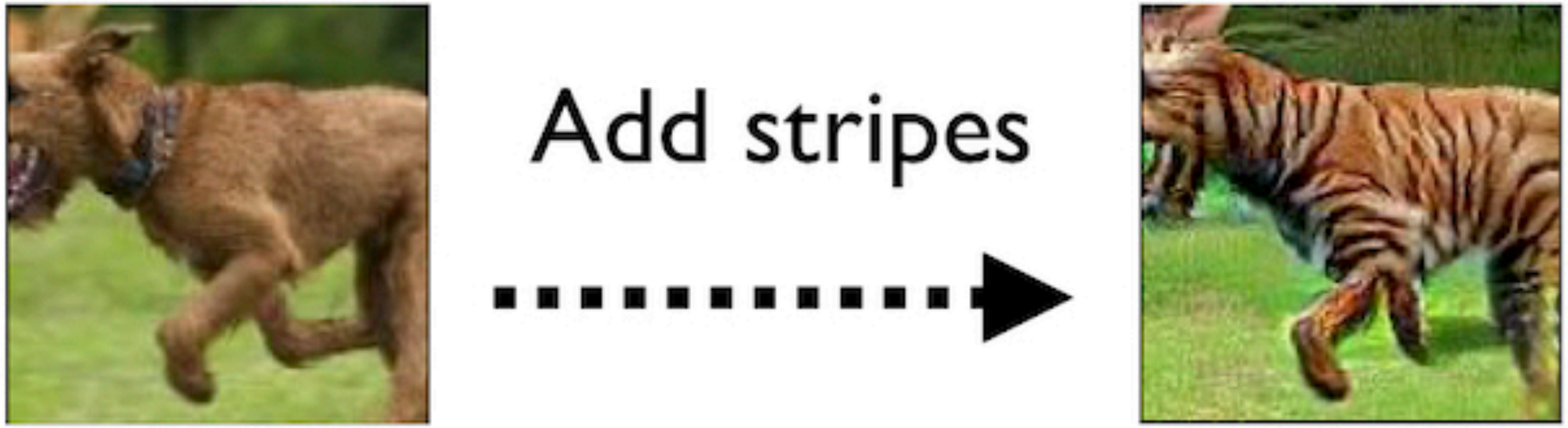
Robustness → Better Representations





Direct feature visualization

Robustness → Better Representations



Direct feature manipulation

Robustness → Better Representations

Original image

label: "insect"; prediction: "dog"

Feature-level sensitivity analysis

What else can we do?

[Santurkar Tsipras Tran Ilyas Engstrom M ‘19]

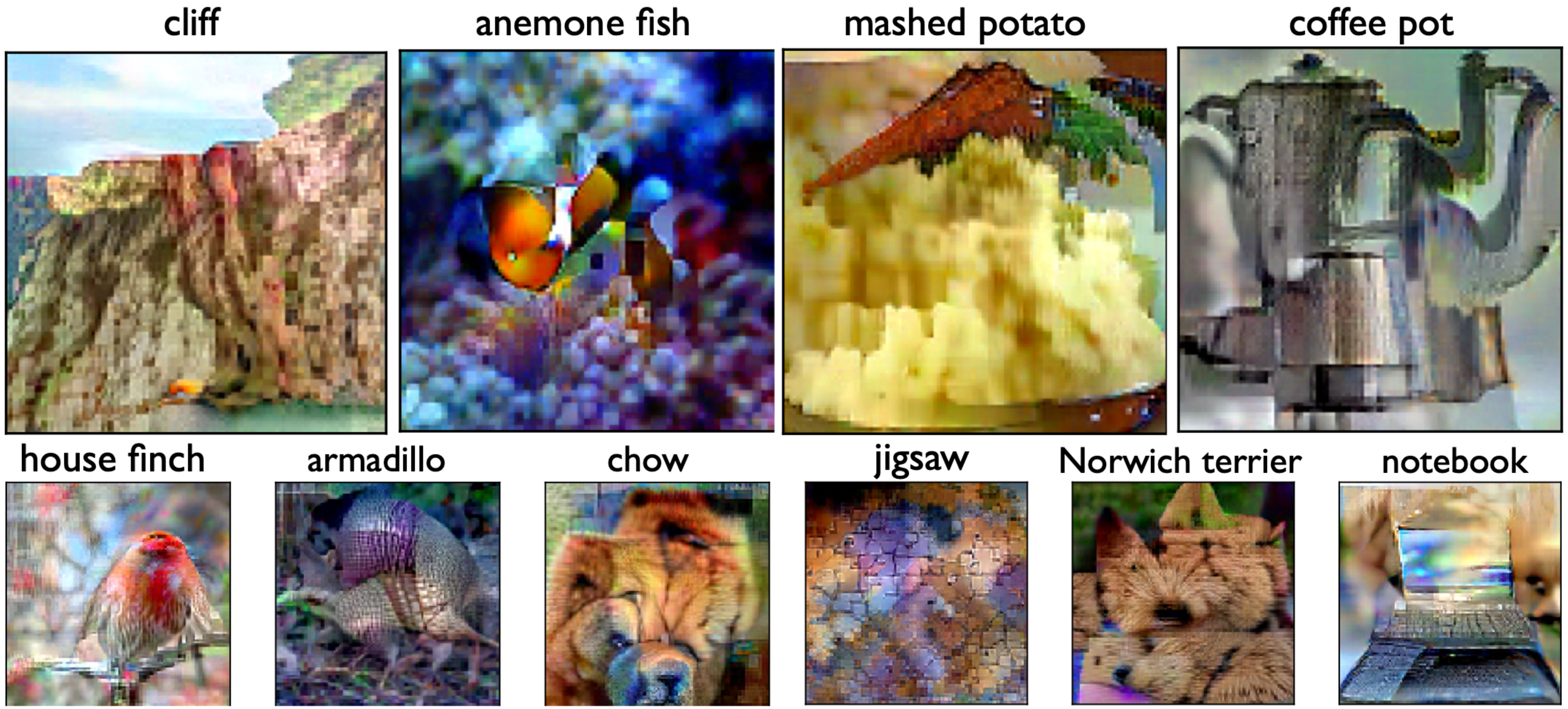
Robustness → CV Applications

A **single robust classifier** suffices to perform a wide range of computer vision task

In fact: The simplest possible approach is enough

→ **Classifier + grad descent** is all one needs

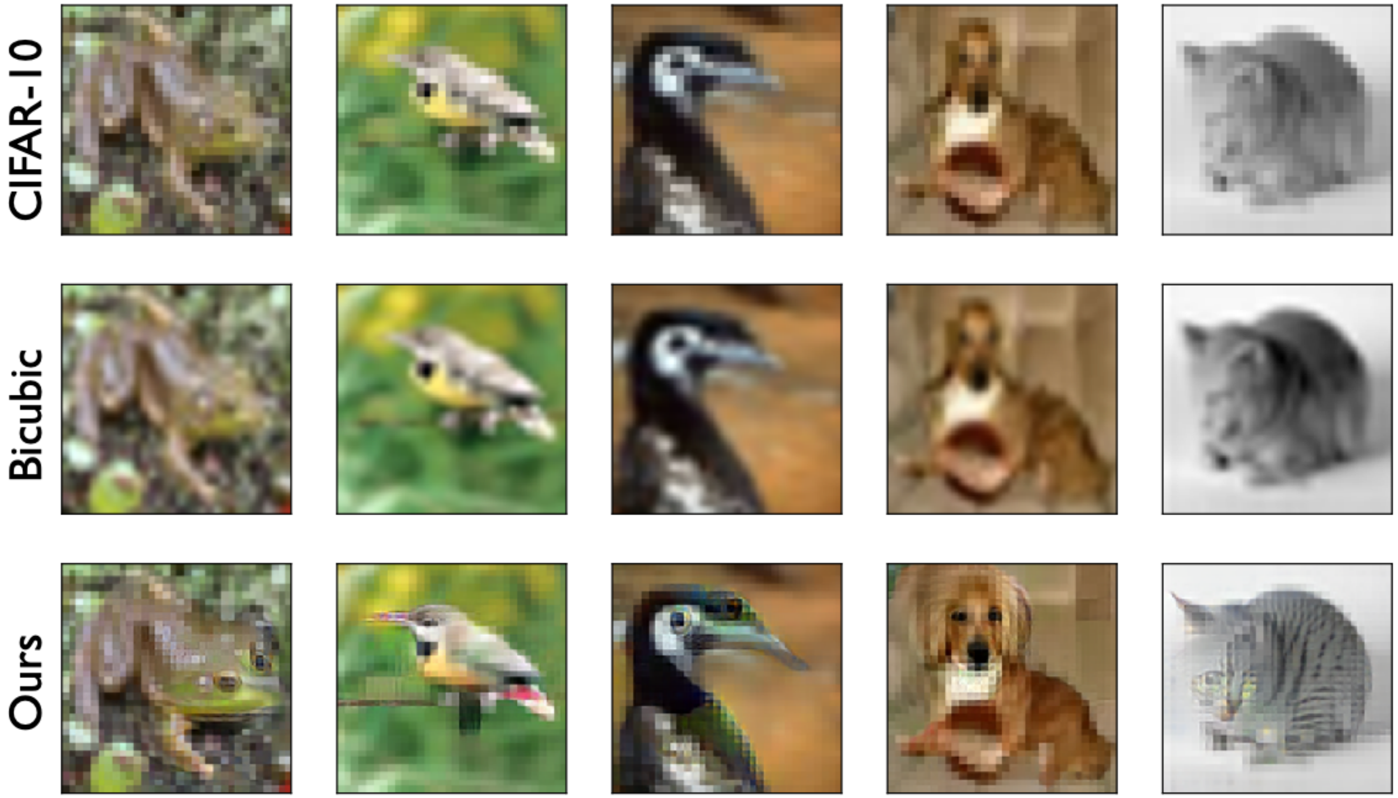
Robustness → CV Applications



(<u>Random</u> samples, 1K training images, <u>no</u> tuning)

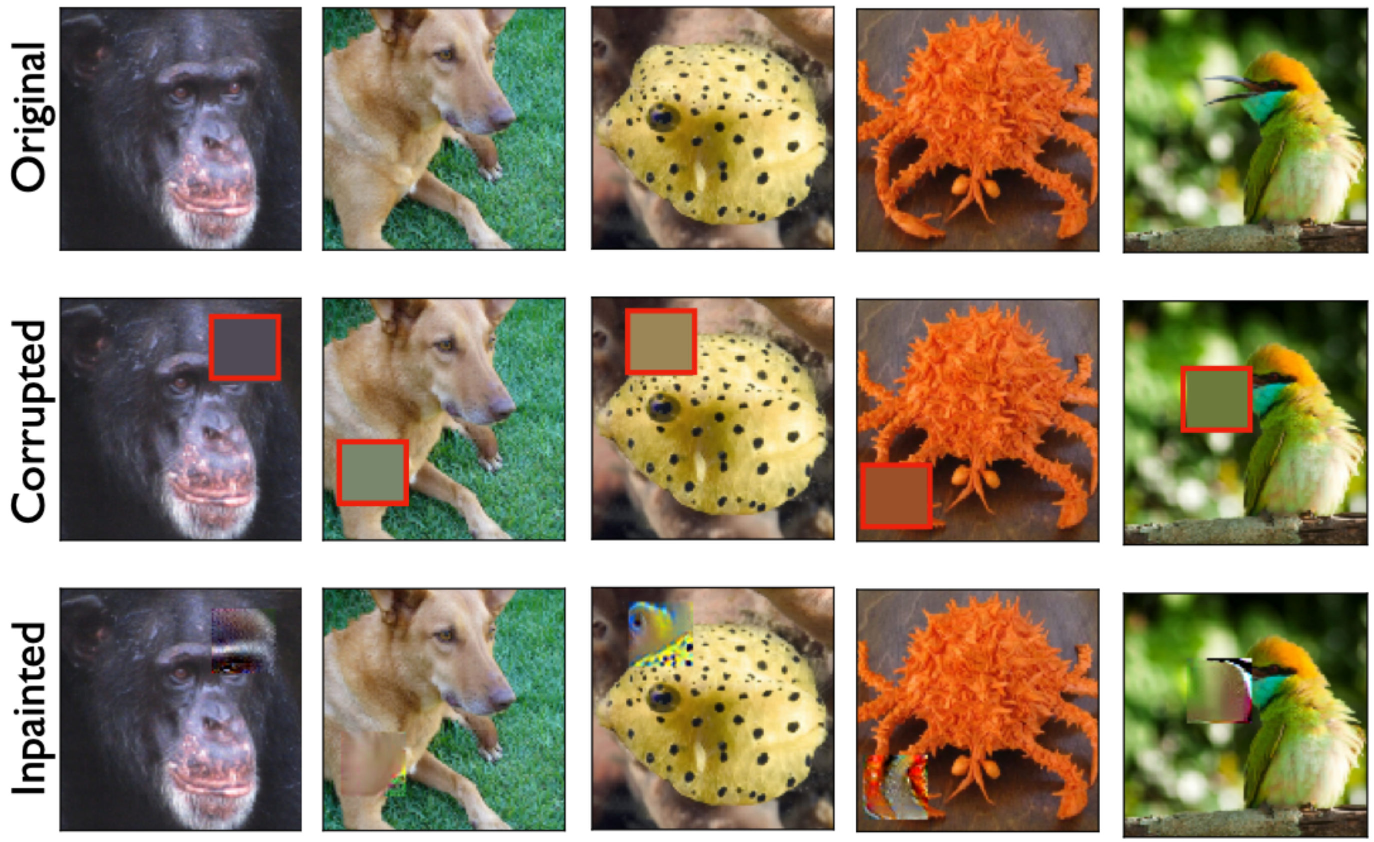
Generative models (that work **better** on **large** datasets)

Robustness → CV Applications



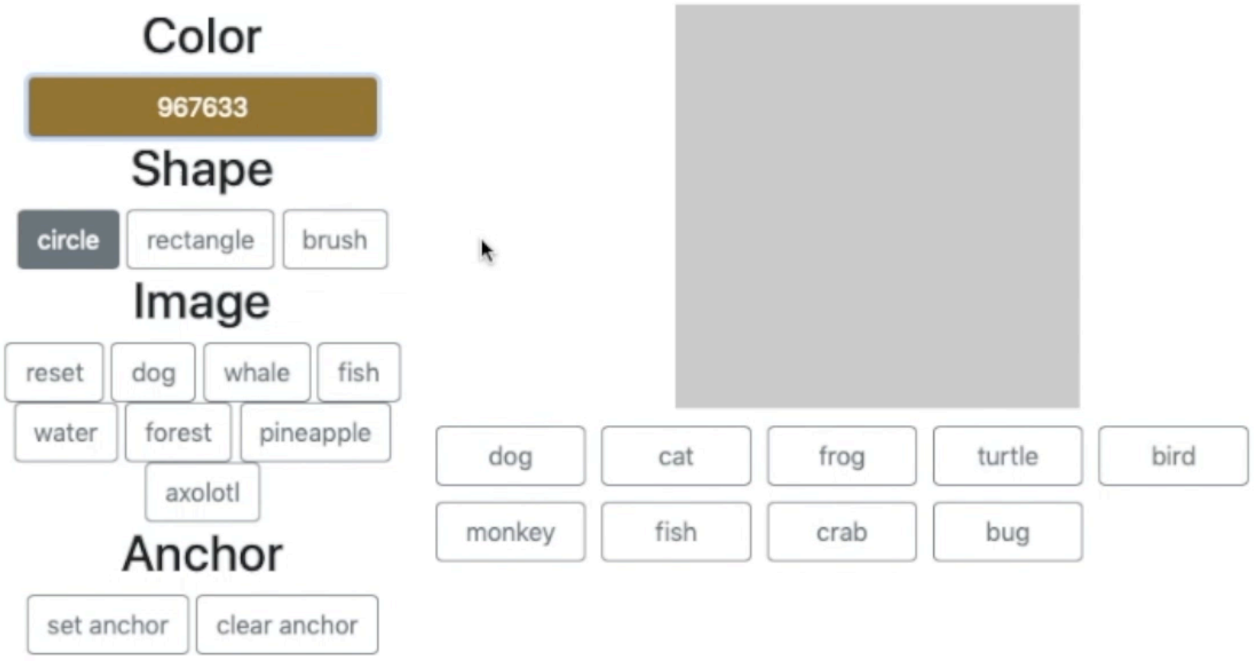
Super-Resolution

Robustness → CV Applications



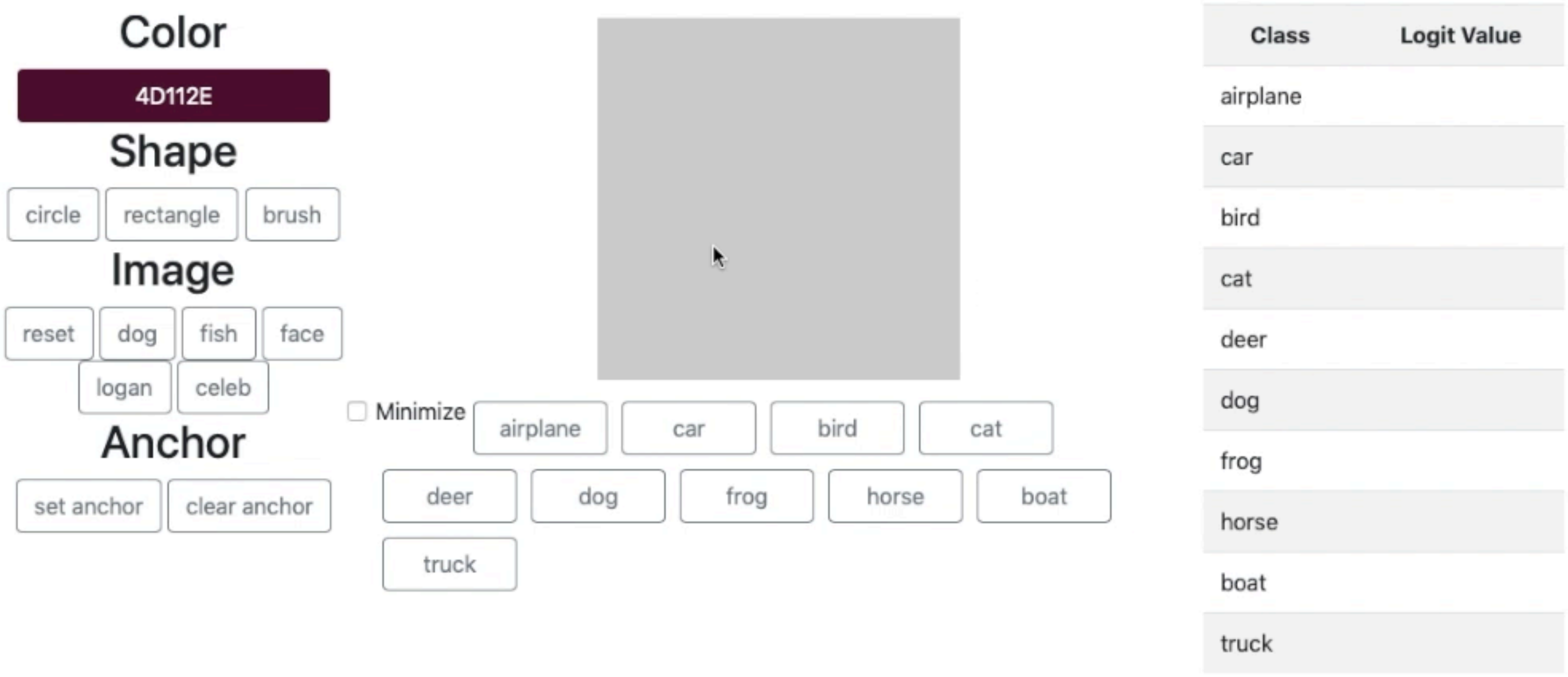
In-Painting

Robustness → CV Applications



Interactive **image class** manipulation

Robustness → CV Applications



Enables exploration of data space

See: http://bit.ly/robustness_demo

Takeaways

Adversarial examples arise from
non-robust features in the data

→ These features **do** help in generalization (a lot!)

→ **Robust training/Randomized smoothing** prevents the model from depending on them (hence they make models be robust)

→ Explains many aspects of robustness (e.g., transferability)

→ **Enables a new capability:** Robustification

→ Interpretability needs to be addressed **at training time**

Robust models yield more human aligned representations

→ Enables a broad range of vision applications (in a simple way)

But: Adv. robustness is not only about robustness to an adversary → it's about **how our models learn**

→ What is the "right" notion of generalization?
Is it really about getting max accuracy possible?

→ How to measure distribution shift?
Shouldn't it be more about representations?

→ How much do we value human alignment/interpretability?

Adversarial robustness =
Framework for making our models better

Here: "Adversary" corresponds to a "human critic"