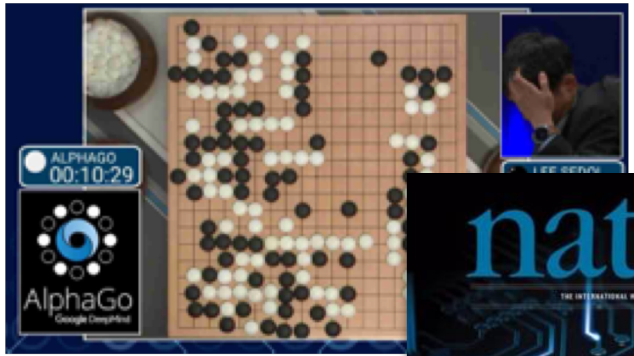
Machine Learning: A Security Perspective

Aleksander Mądry

madry-lab.ml

Machine Learning: The Success Story



Machine Learning: The Success Story

IS "DEEP LEARNING" A REVOLUTION IN ARTIFICIAL INTELLIGENCE?

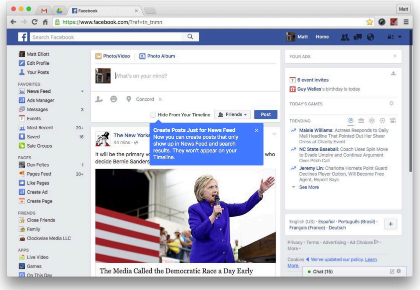
Trump Signs Executive Order Promoting Artificial Intelligence

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.

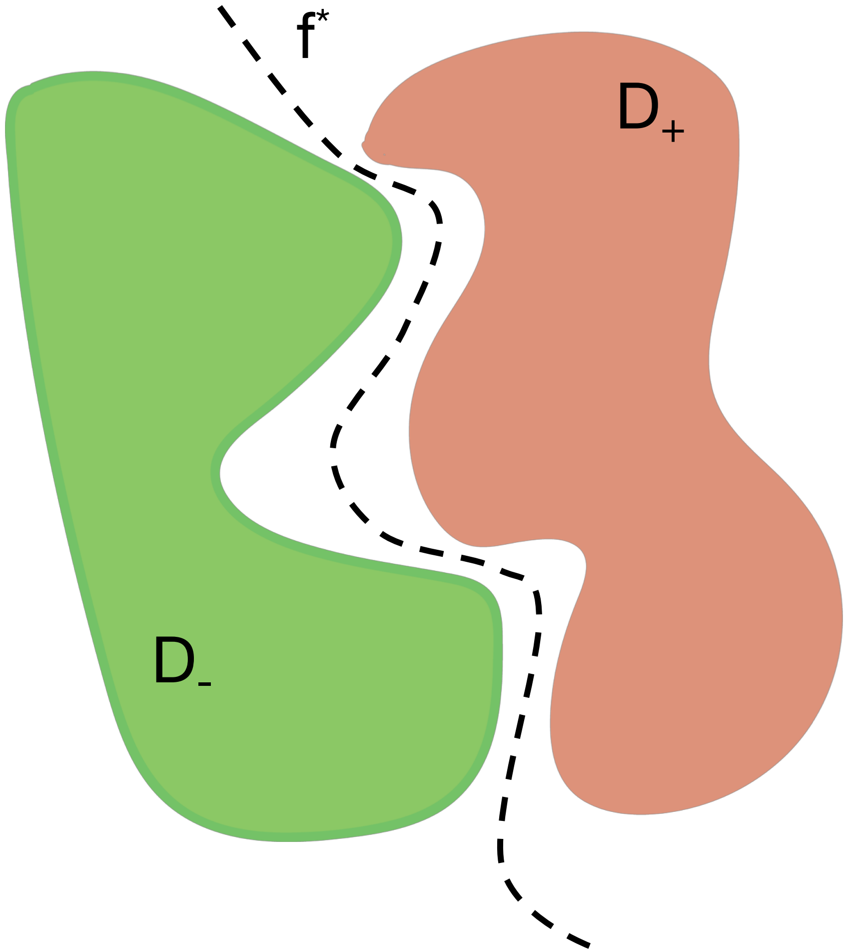


Is ML **truly** ready for
real-world deployment?

Can We Truly Rely on ML?

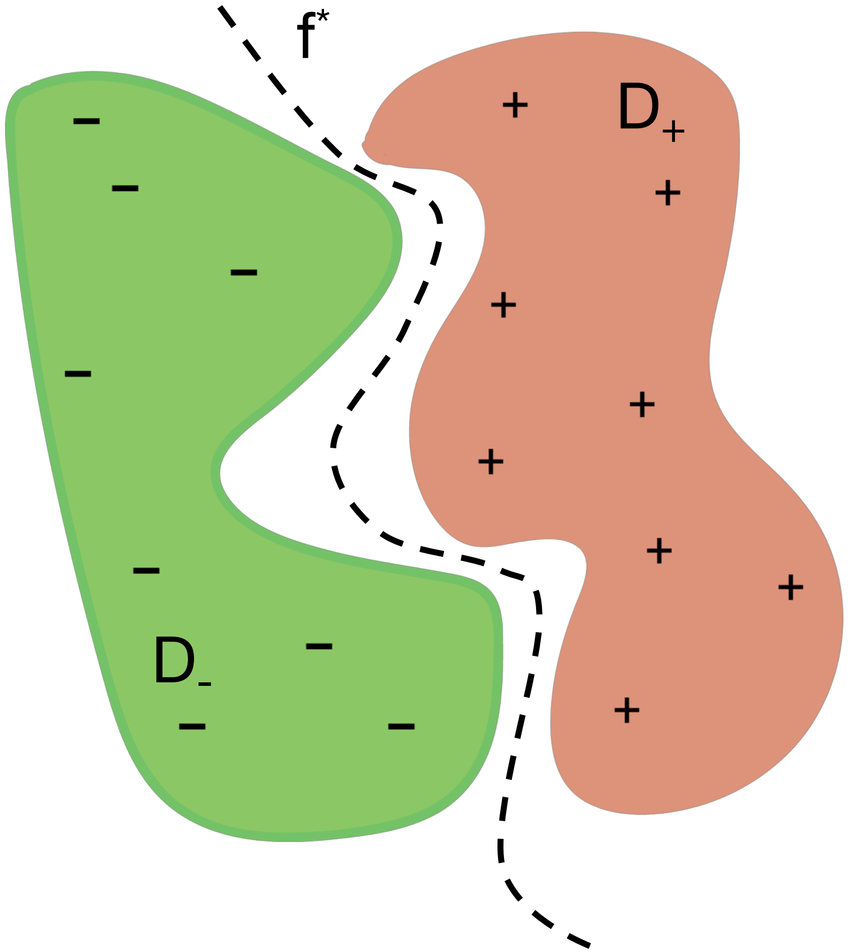
(Supervised) Machine Learning: A Quick Primer

Supervised Machine Learning



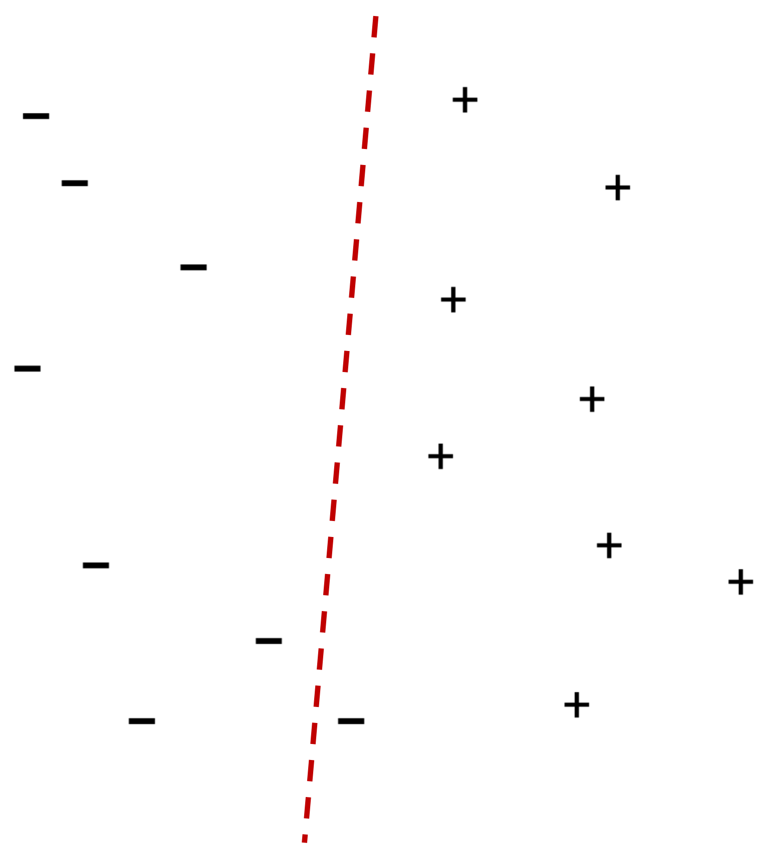
f^*= concept to learn

Supervised Machine Learning



f^*= concept to learn

Supervised Machine Learning



f^*= concept to learn

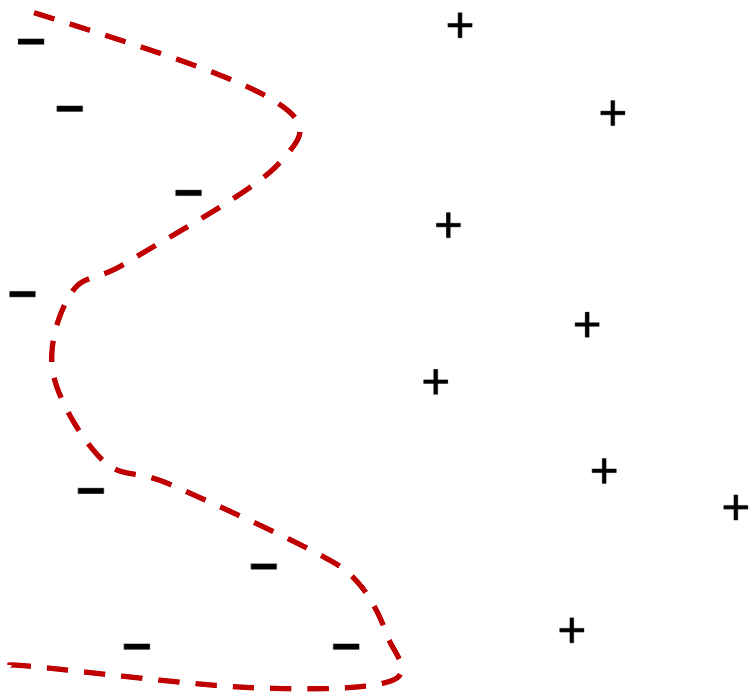
Training: Find parameters $\boldsymbol{\theta}^*$ that make our classifier $\mathbf{f(\theta^*)}$ fit/"explain" the training data (and thus approx. $\mathbf{f^*}$)

Here: f(θ) = a family of classifiers parametrized by **θ**

Choice of the family f(·) is crucial

Too simple → underfitting

Supervised Machine Learning



f^*= concept to learn

Training: Find parameters $\boldsymbol{\theta}^*$ that make our classifier $\mathbf{f}(\boldsymbol{\theta}^*)$ fit/"explain" the training data (and thus approx. $\mathbf{f}^*$)

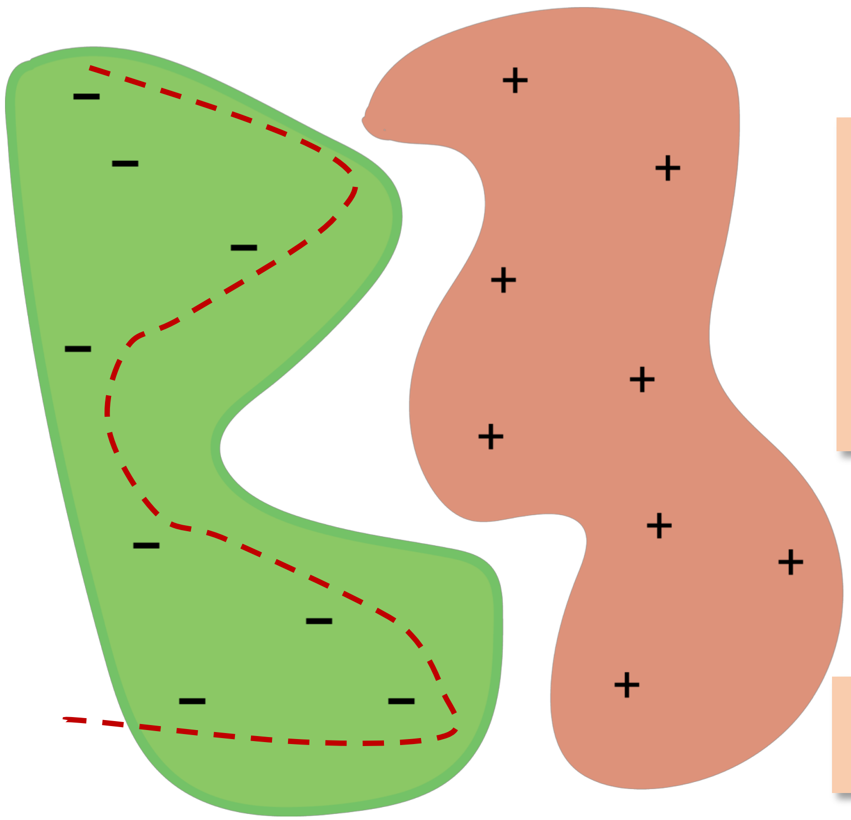
Here: $\mathbf{f}(\boldsymbol{\theta})$ = a family of classifiers parametrized by $\boldsymbol{\theta}$

Choice of the family $f(\cdot)$ is crucial

Too flexible → overfitting

Supervised Machine Learning

f^*= concept to learn



Training: Find parameters $\boldsymbol{\theta}^*$ that make our classifier $\mathbf{f}(\boldsymbol{\theta}^*)$ fit/"explain" the training data (and thus approx. $\mathbf{f}^*$)

Here: $\mathbf{f}(\boldsymbol{\theta})$ = a family of classifiers parametrized by $\boldsymbol{\theta}$

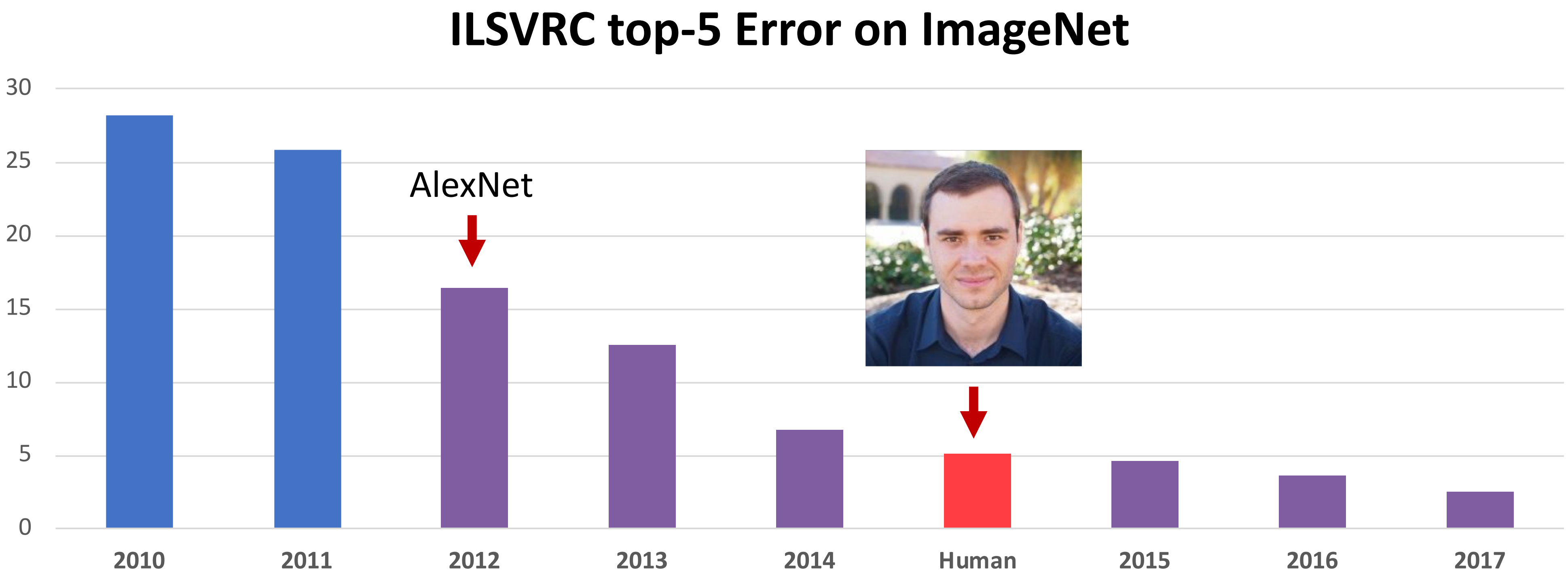
Choice of the family f(·) is crucial

Too flexible → overfitting

ML developed a rich theory to guide us here (and this was its **only** goal)

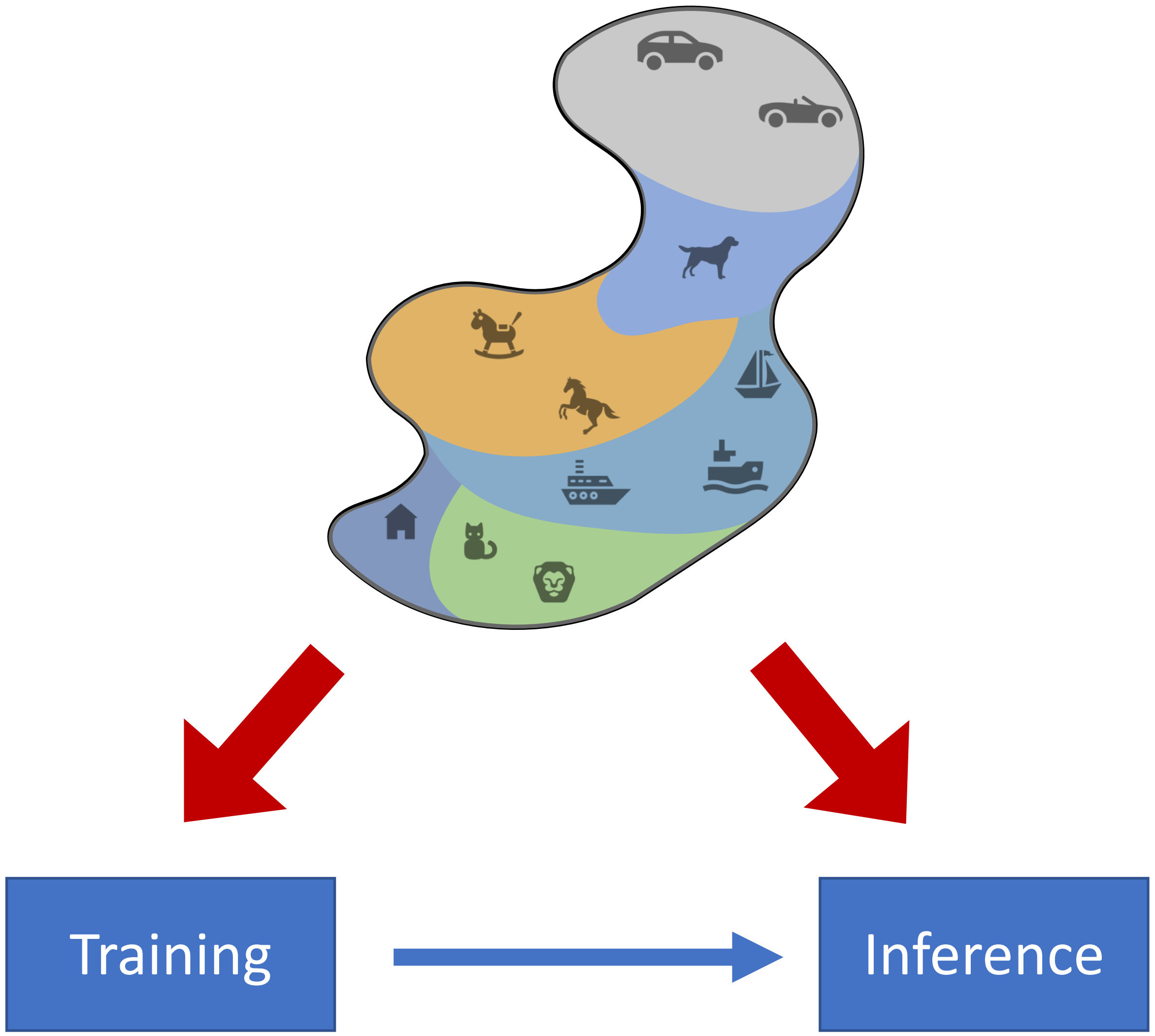
Robust, Reliable and Secure ML: The Challenges

ImageNet: An ML Home Run



But what do these results *really* mean?

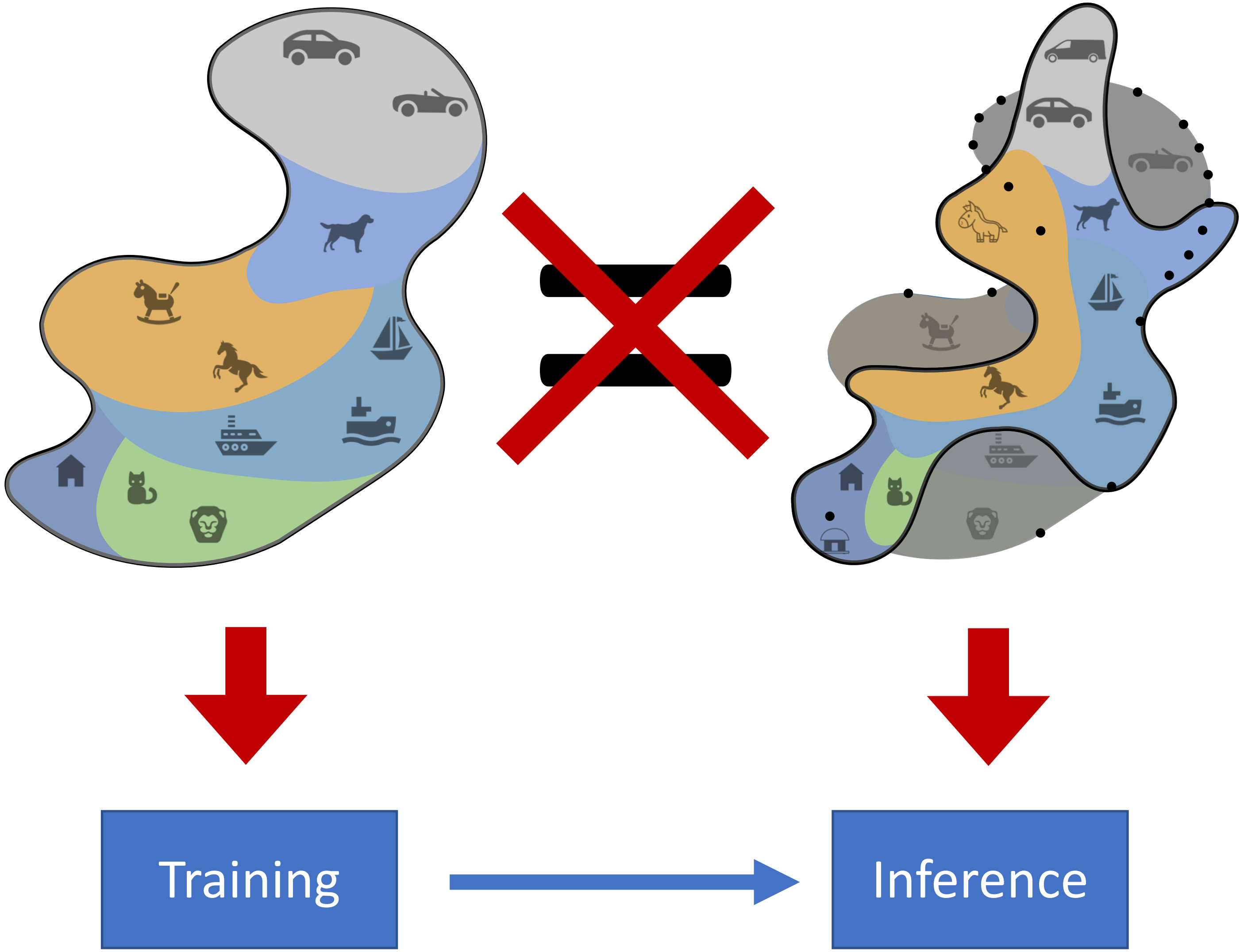
A Limitation of the (Supervised) ML Framework



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



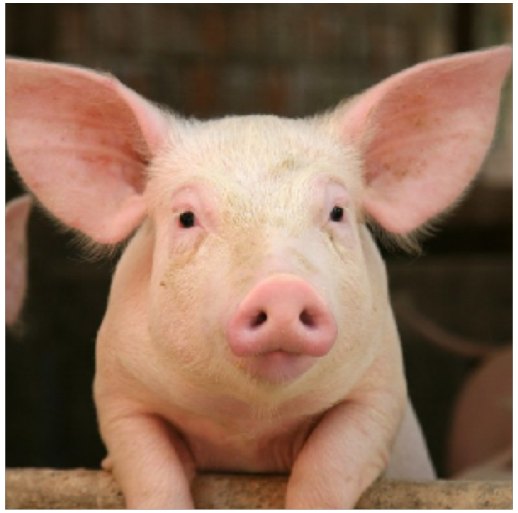
Measure of performance:
Fraction of mistakes during testing

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

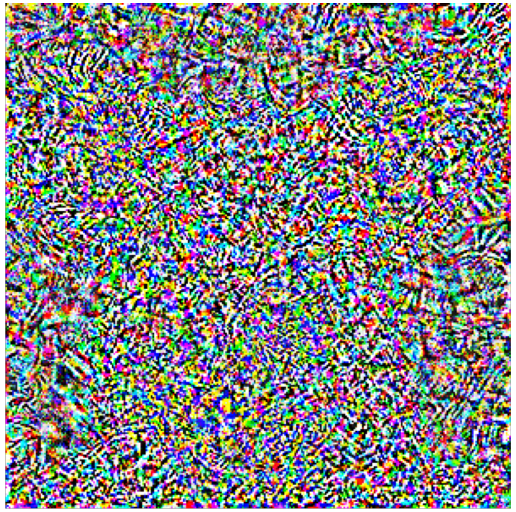
What can go wrong?

ML Predictions Are (Mostly) Accurate but Brittle

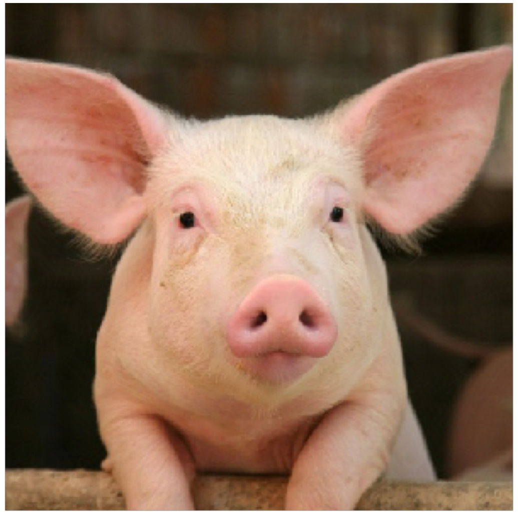
“pig” (91%) **noise (NOT random)** **“airliner” (99%)**



+ 0.005 x



=



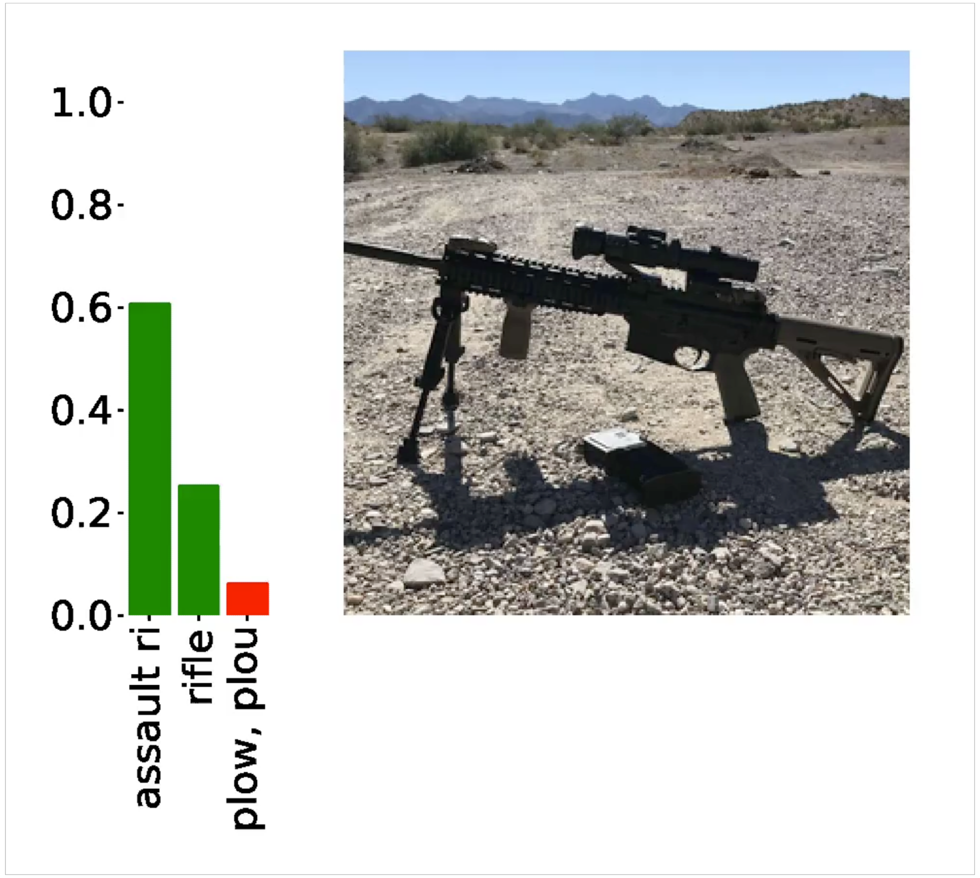
[Szegedy Zaremba Sutskever Bruna Erhan Goodfellow Fergus 2013]
[Biggio Corona Maiorca Nelson Srndic Laskov Giacinto Roli 2013]

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

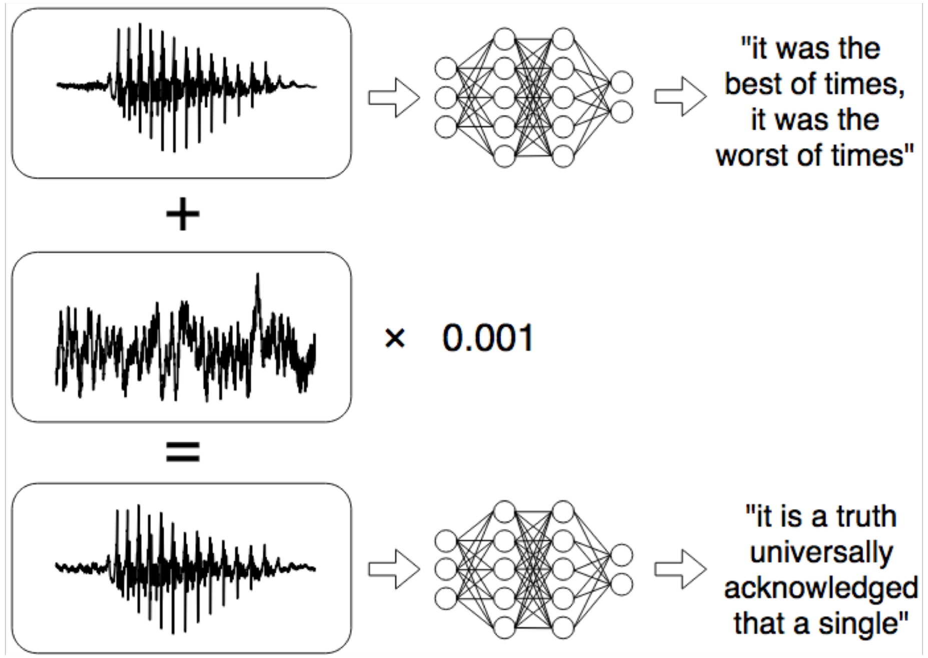
So: Brittleness of ML is a thing

Should we be worried?

Why Is This Brittleness of ML a Problem?

→ Security

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

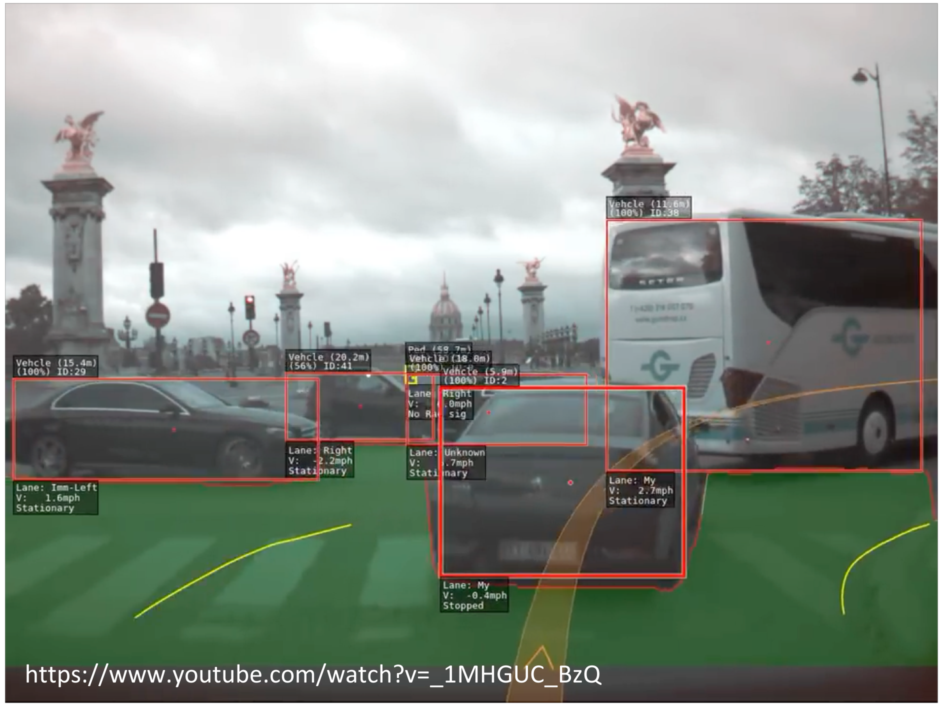


[Sharif Bhagavatula Bauer Reiter 2016]:
Glasses that fool face recognition

Why Is This Brittleness of ML a Problem?

→ Security

→ Safety



https://www.youtube.com/watch?v=_1MHGUC_BzQ

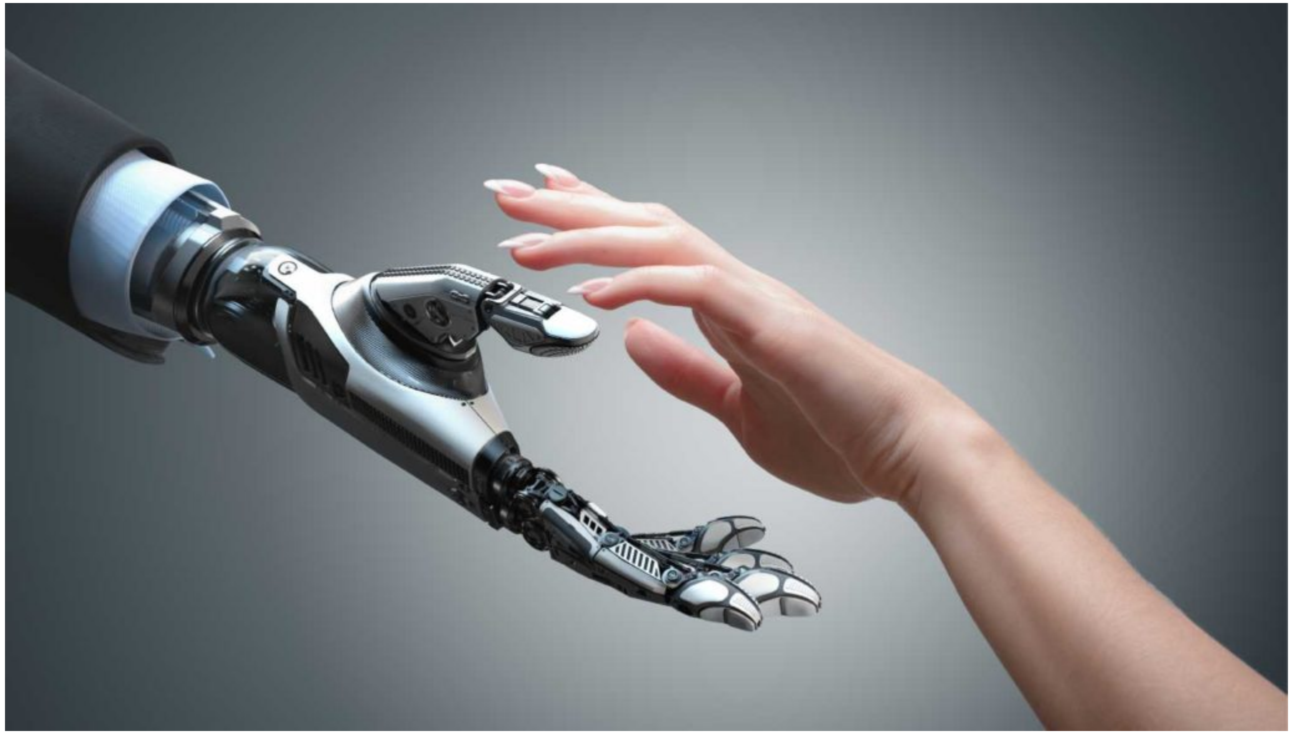
https://www.youtube.com/watch?v=TIUU1xNqI8w

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

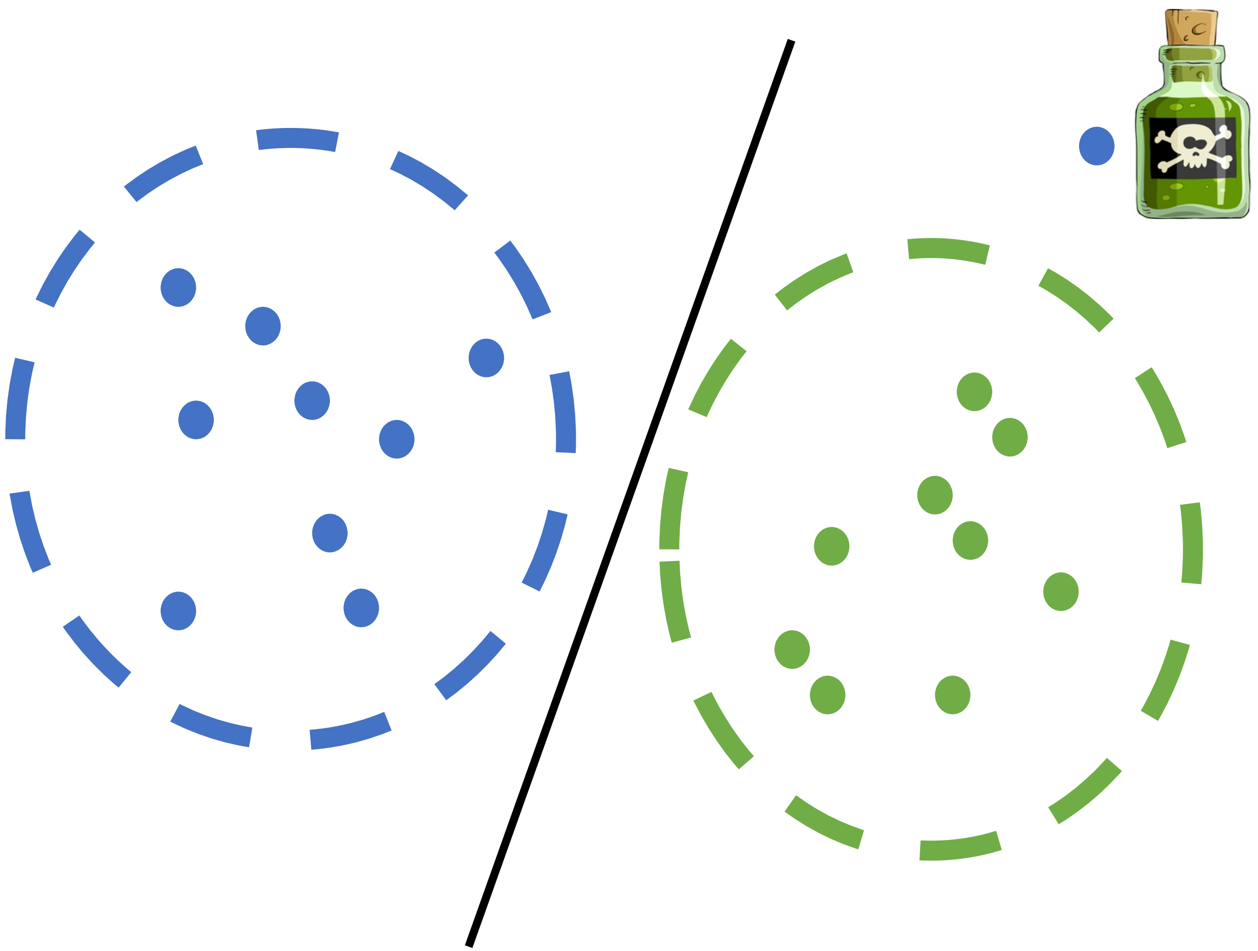
(Deep) ML is ”data hungry”

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

What can go wrong?

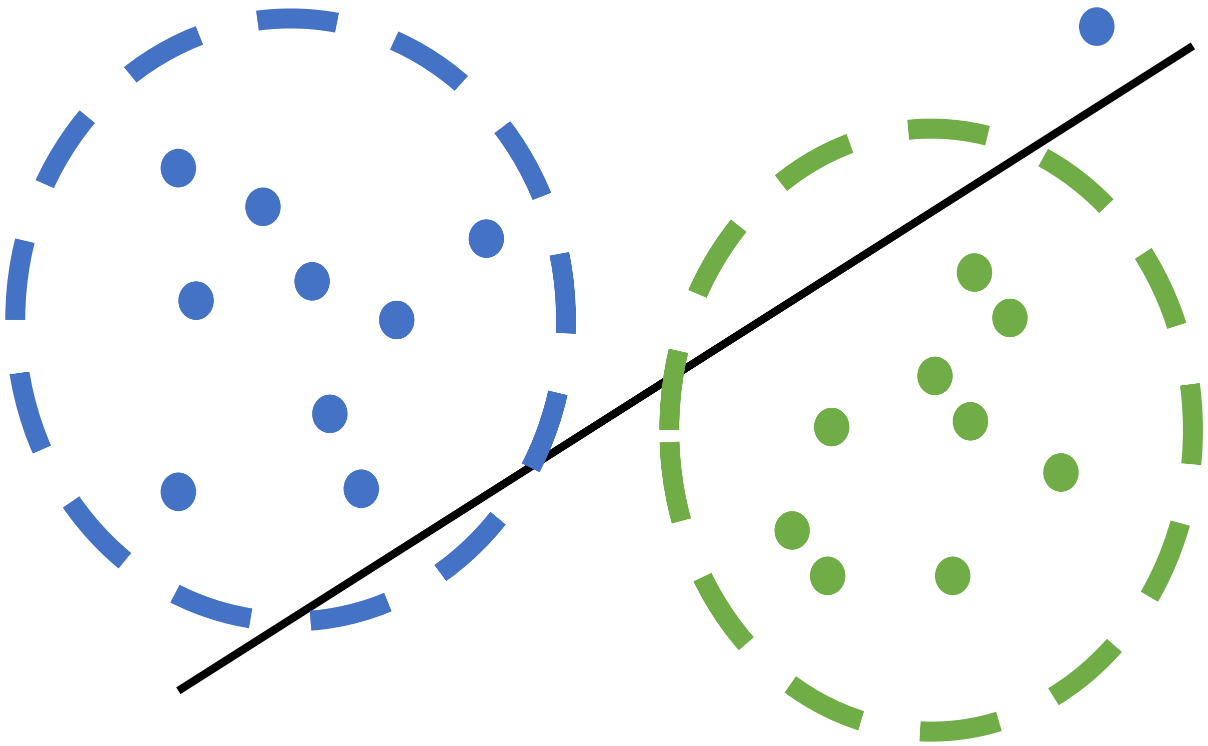
Data Poisoning

Goal: Maintain training accuracy but hamper generalization



Data Poisoning

Goal: Maintain training accuracy but hamper generalization



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

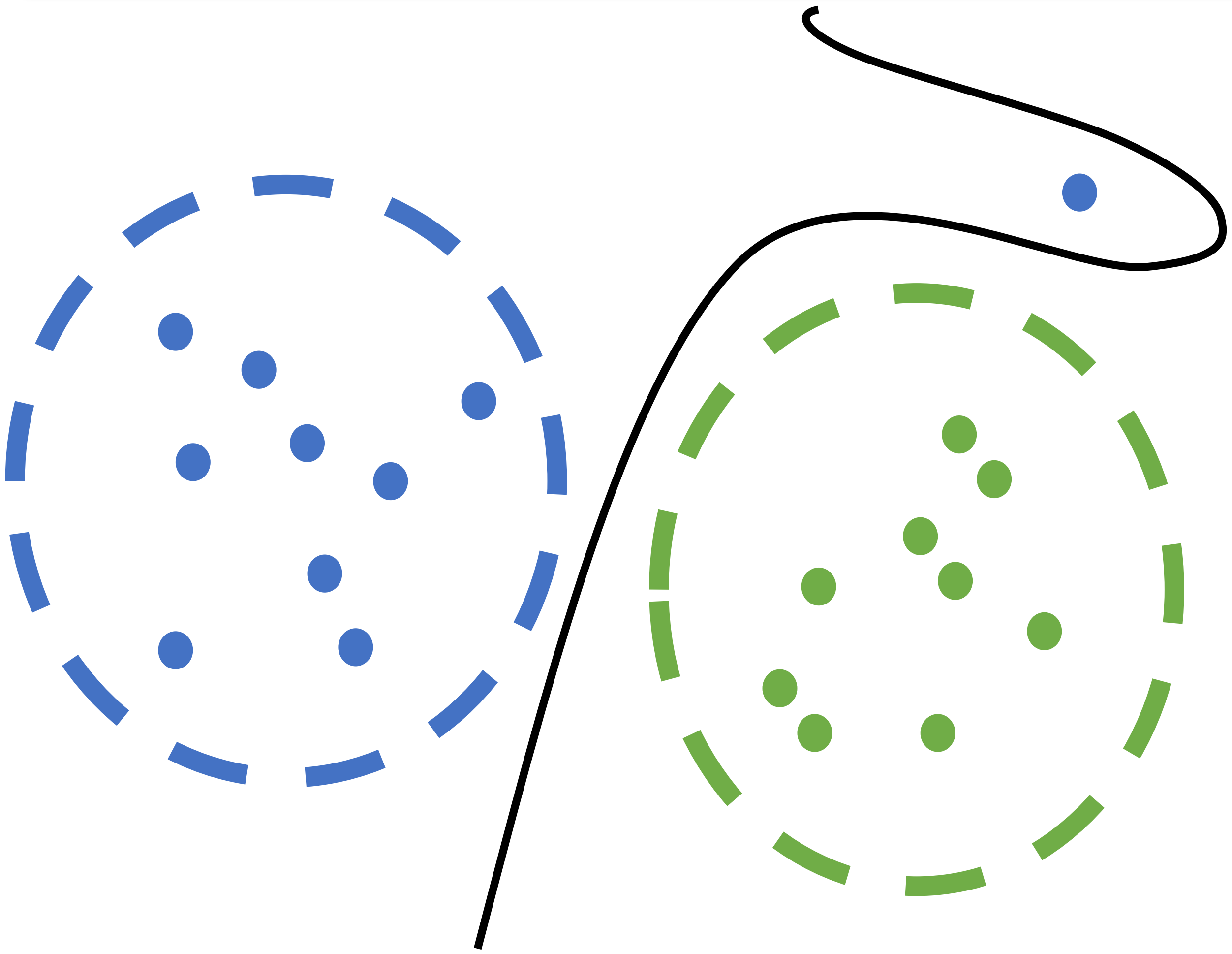
→ **But:** seems less so in deep learning

→ **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

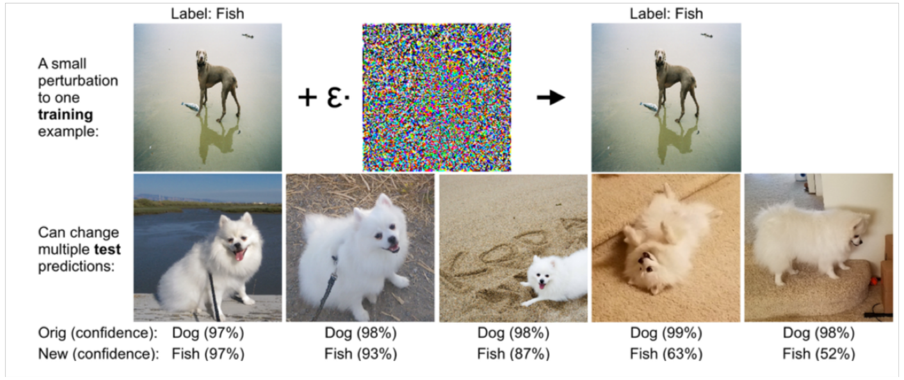
→ **Reason:** Memorization?

Is that it?

Data Poisoning

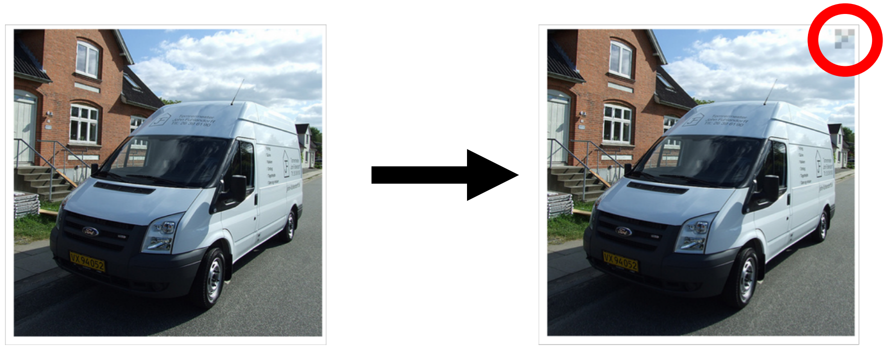
classification of **specific** inputs

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



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

But: This gets (much) worse



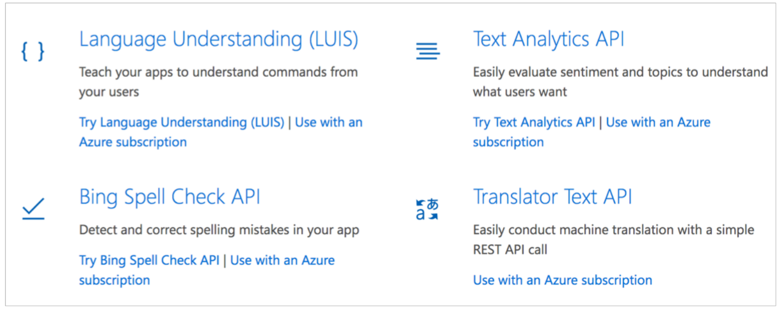
"van" **"dog"**

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

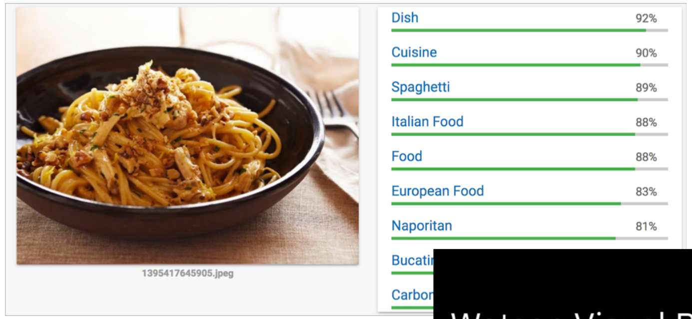
Some defense mechanisms exist but not there (yet?) **[Tran Li M 2018]**

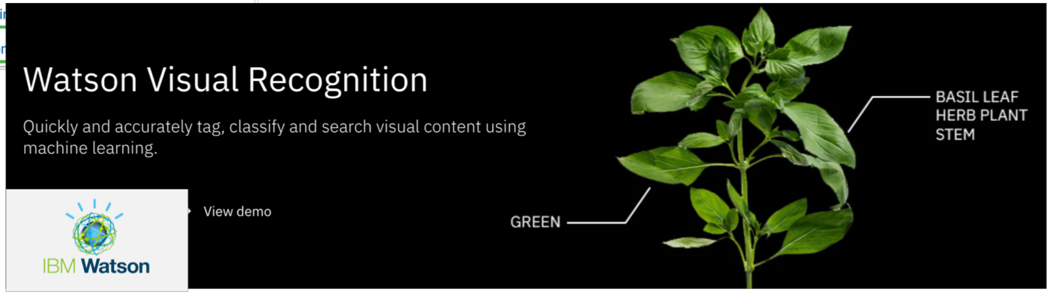
Is That It?

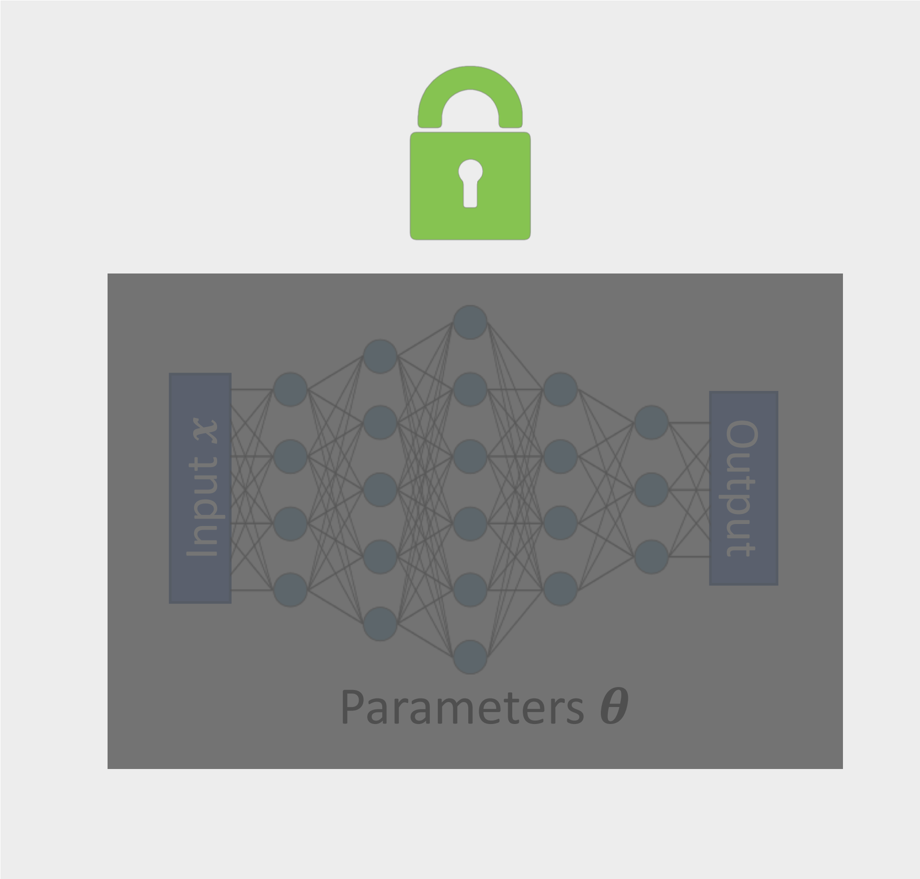
Microsoft Azure (Language Services)



Google Cloud Vision API





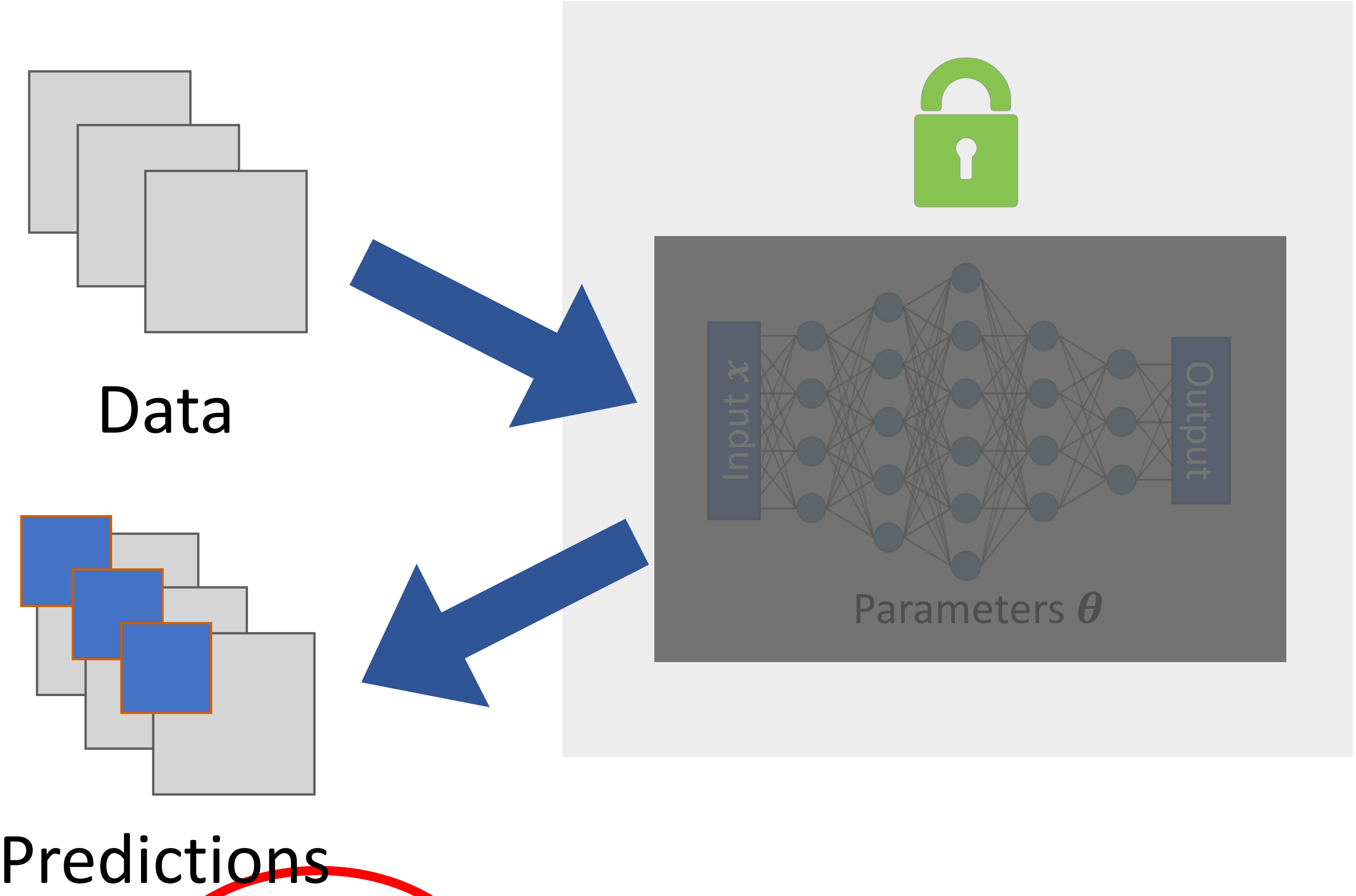


Training → Inference → Deployment

Is That It?

Does limited access give security?

In short: No



Training → Inference → Deployment

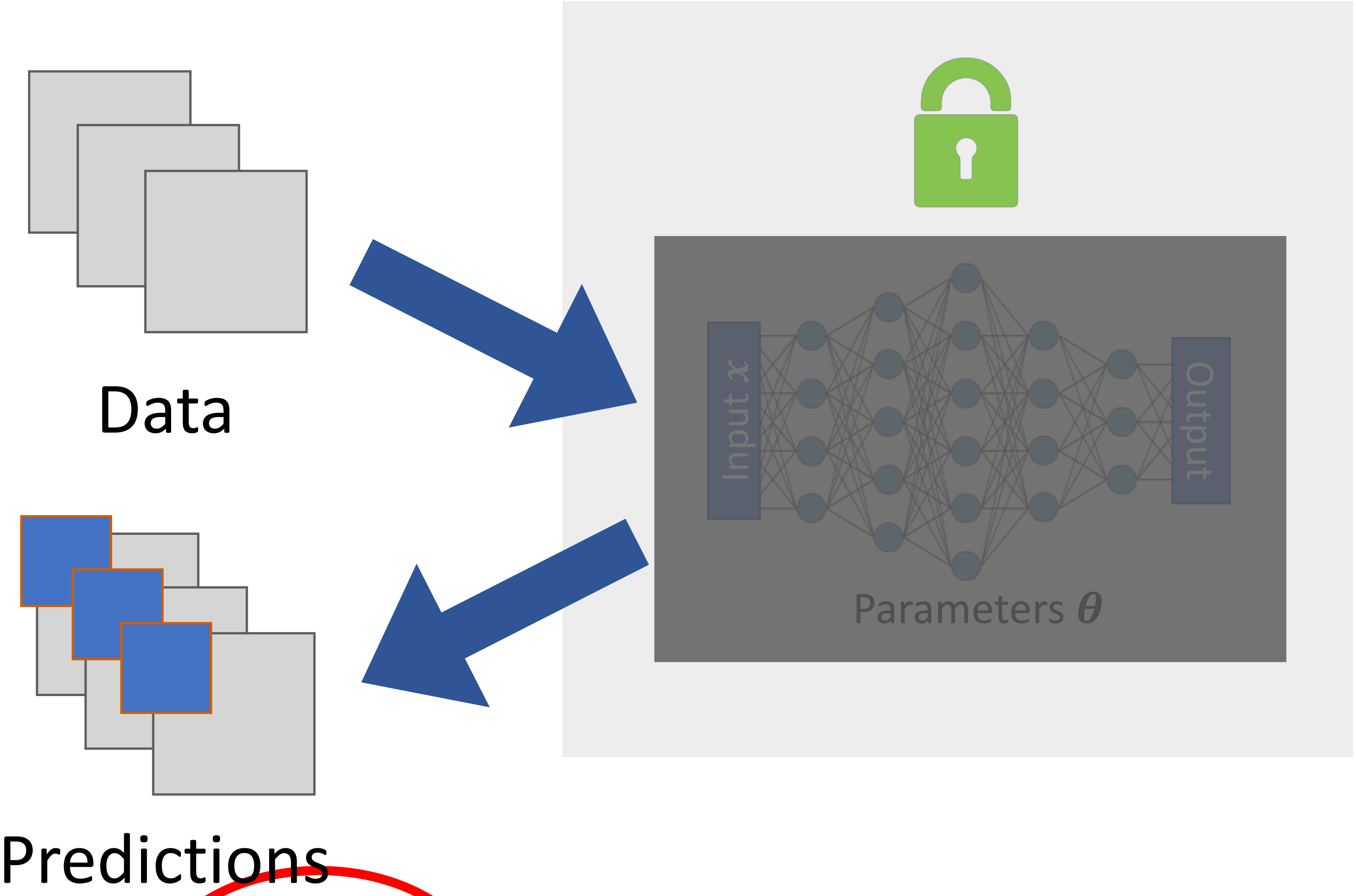
Black box attacks

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?)

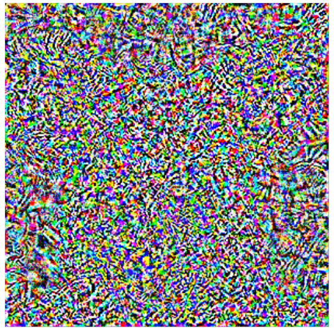
No: But we need to re-think how we do ML

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

Towards Adversarially Robust Models

“pig” (91%)

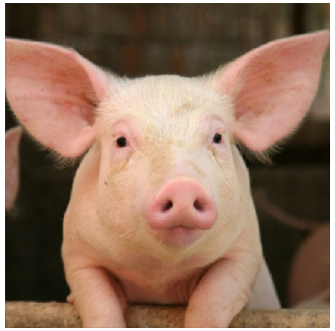
+ 0.005 x



=

“pig”

~~“airliner”~~ (99%)



Where Do Adversarial Examples Come From?

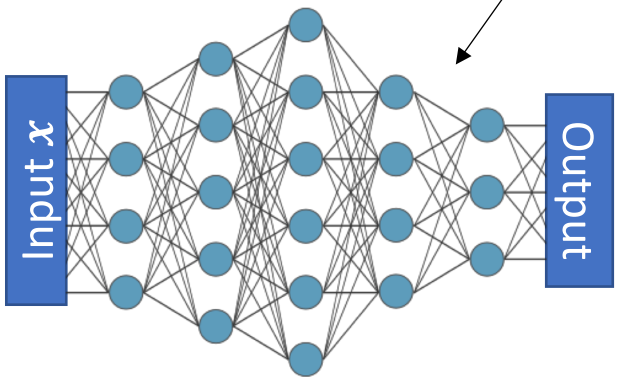
To get an adv. example

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

Model Parameters   Input   Correct Label

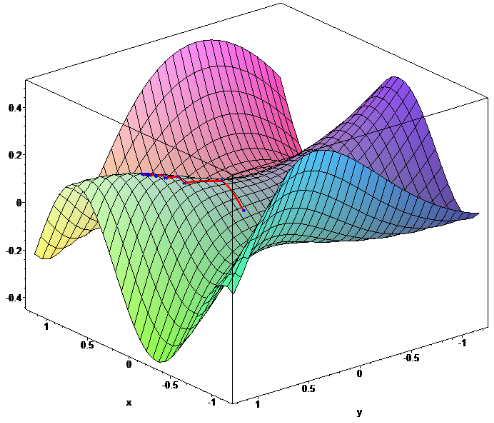
$$min_{\theta}\ loss(\theta, x, y)$$

Differentiable



Parameters $\boldsymbol{\theta}$

Can use gradient descent method to find good θ

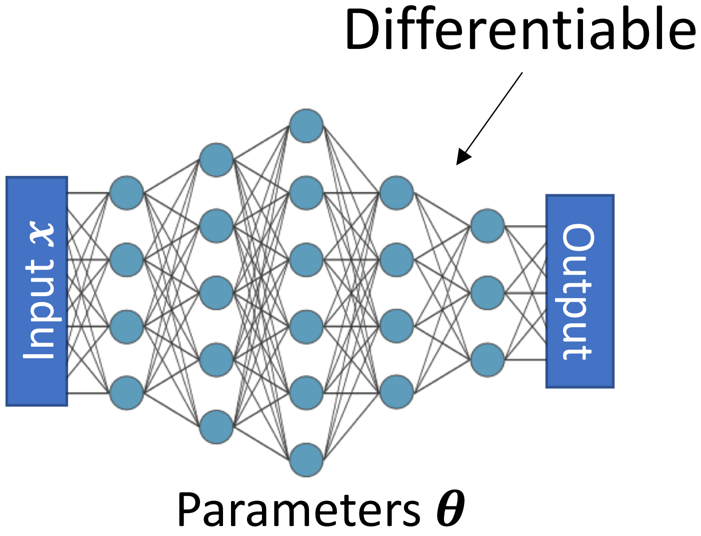


Where Do Adversarial Examples Come From?

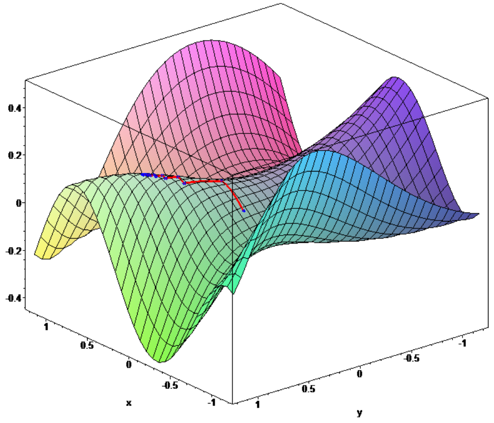
To get an adv. example

~~Goal of training:~~

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



Can use gradient descent method to find good θ

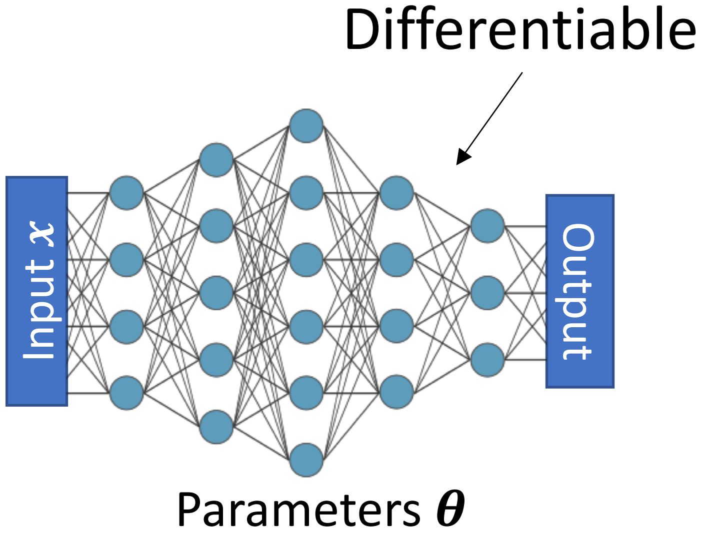


Where Do Adversarial Examples Come From?

To get an adv. example

~~Goal of training~~:

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



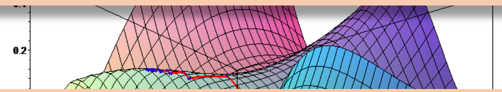
Which δ are allowed?

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

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

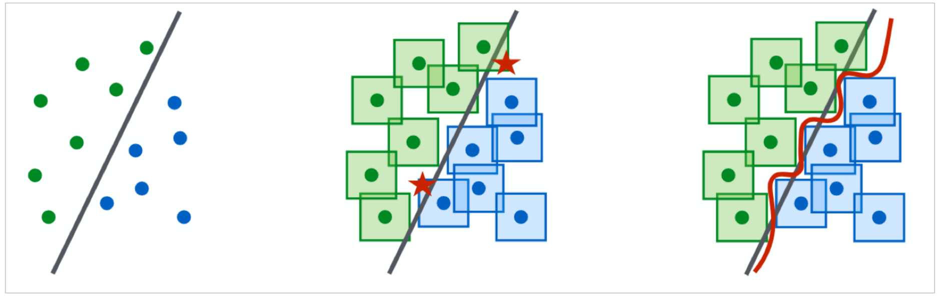
To improve the model: Train on **perturbed** inputs
(aka as "adversarial training" **[Goodfellow Shlens Szegedy '15]**)

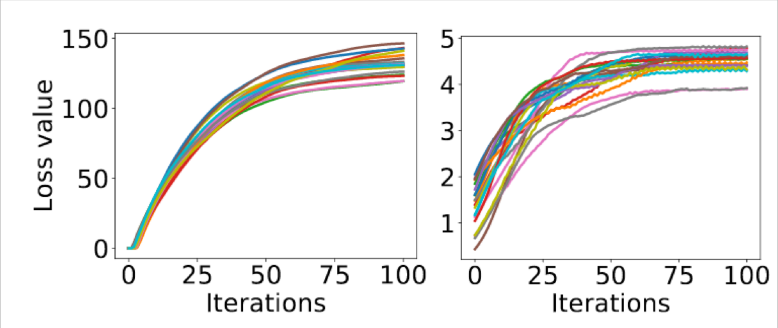
Does this work?   **Yes!** (In practice)
But certain care is required

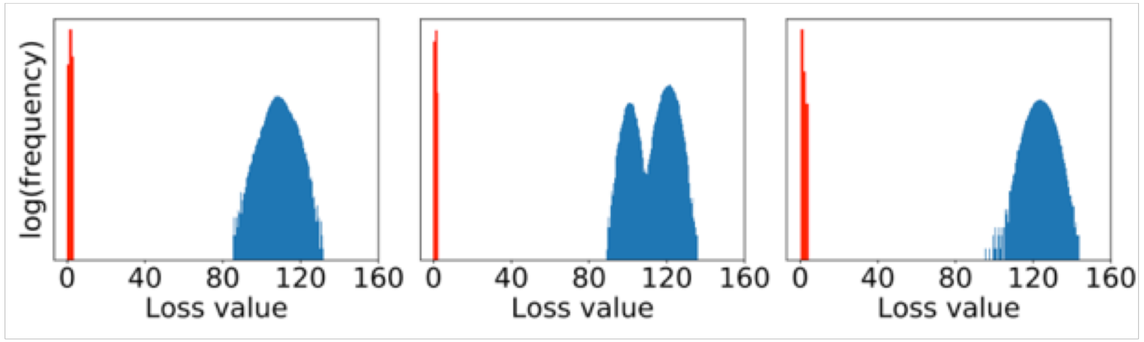
Key Components

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

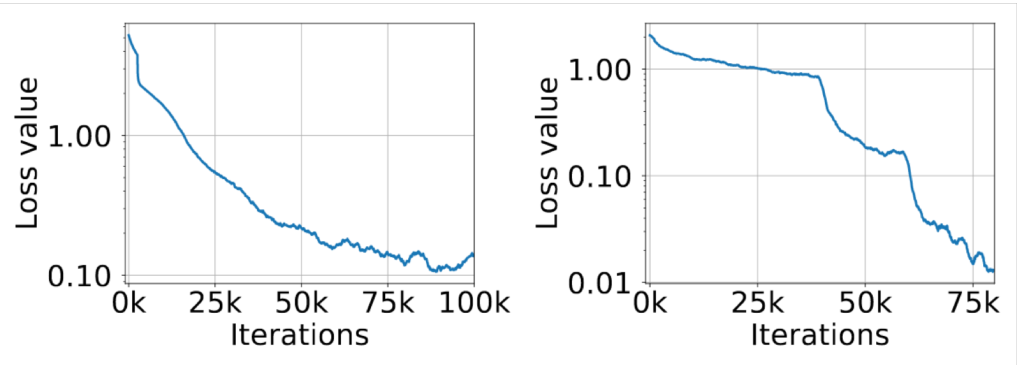
→ Sufficient model capacity



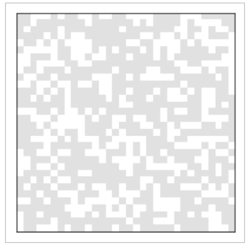
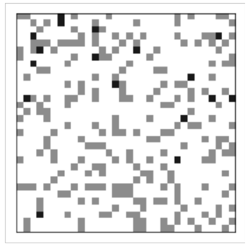
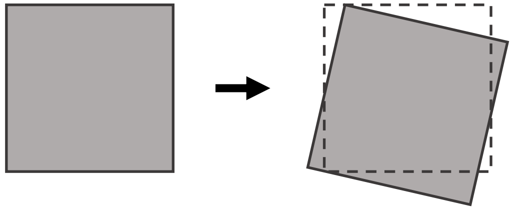




Result: Robustness increases steadily



(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?

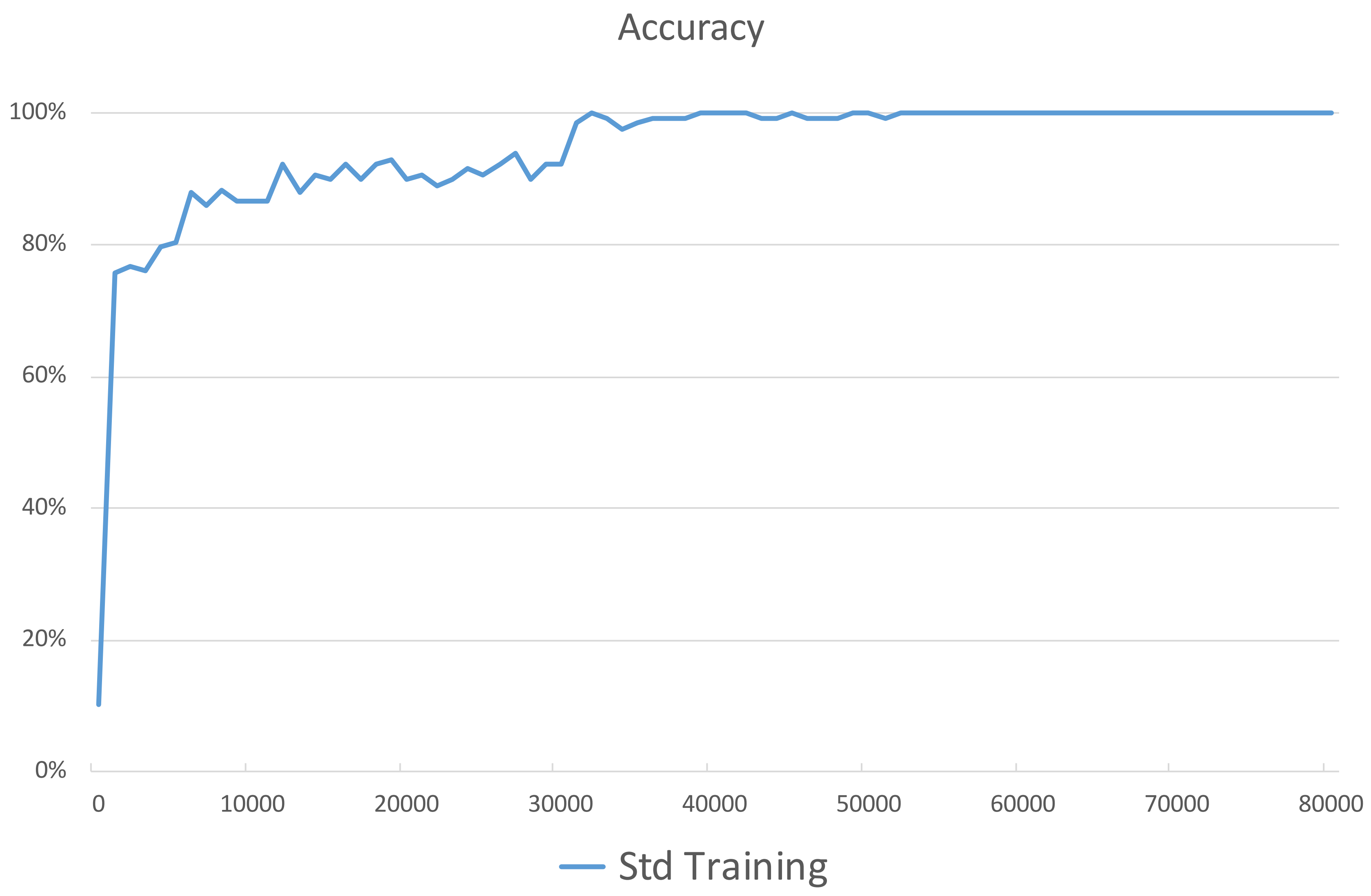
$$\mathbb{E}_{(x,y)\sim D}\left[loss(\theta, x, y)\right]$$

vs

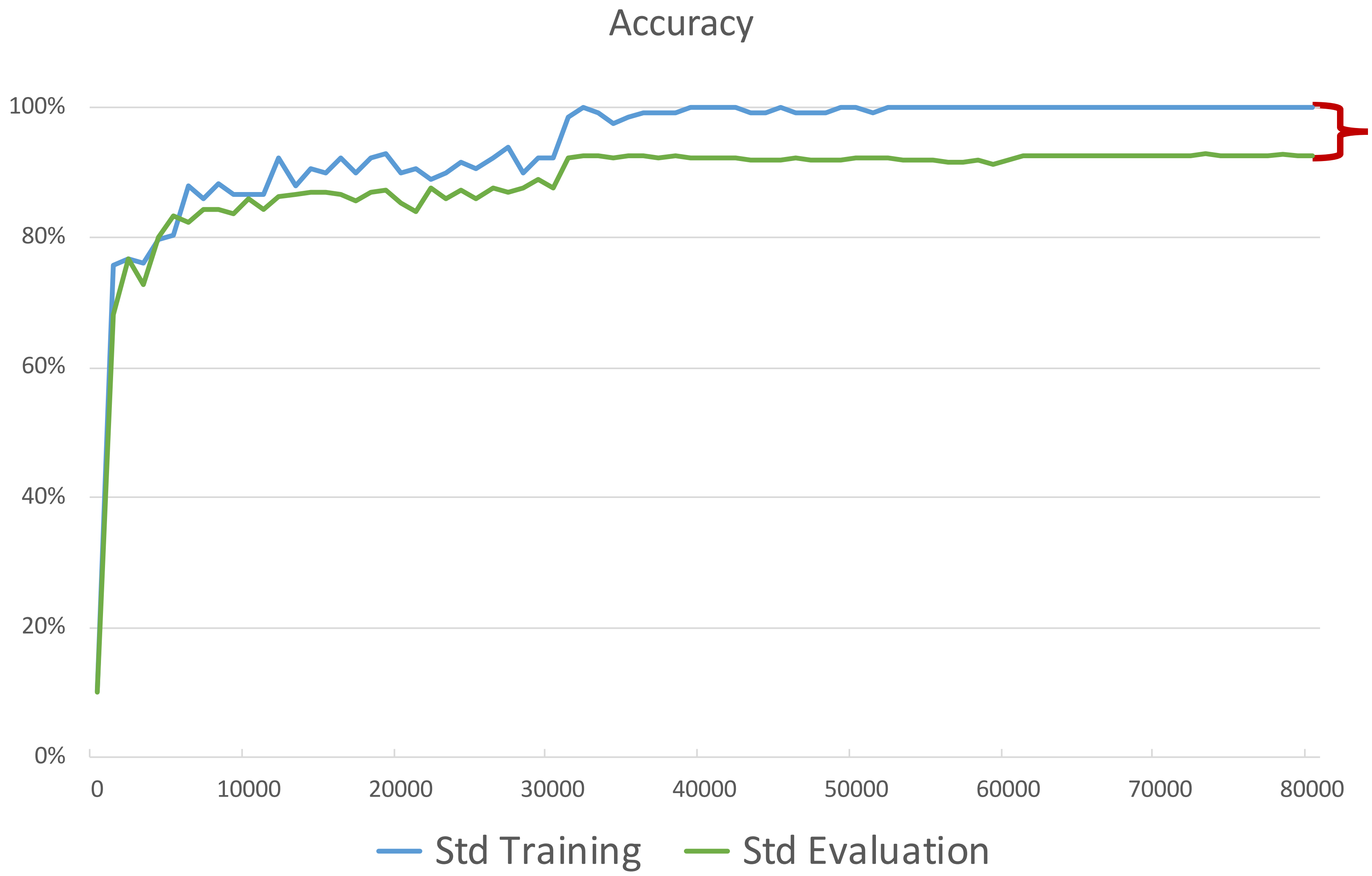
$$\mathbb{E}_{(x,y)\sim D}\left[\max_{\boldsymbol{\delta}\in\boldsymbol{\Delta}} loss(\theta, x + \boldsymbol{\delta}, y)\right]$$

(This goes **beyond** deep learning)

Do Robust Deep Networks Overfit?

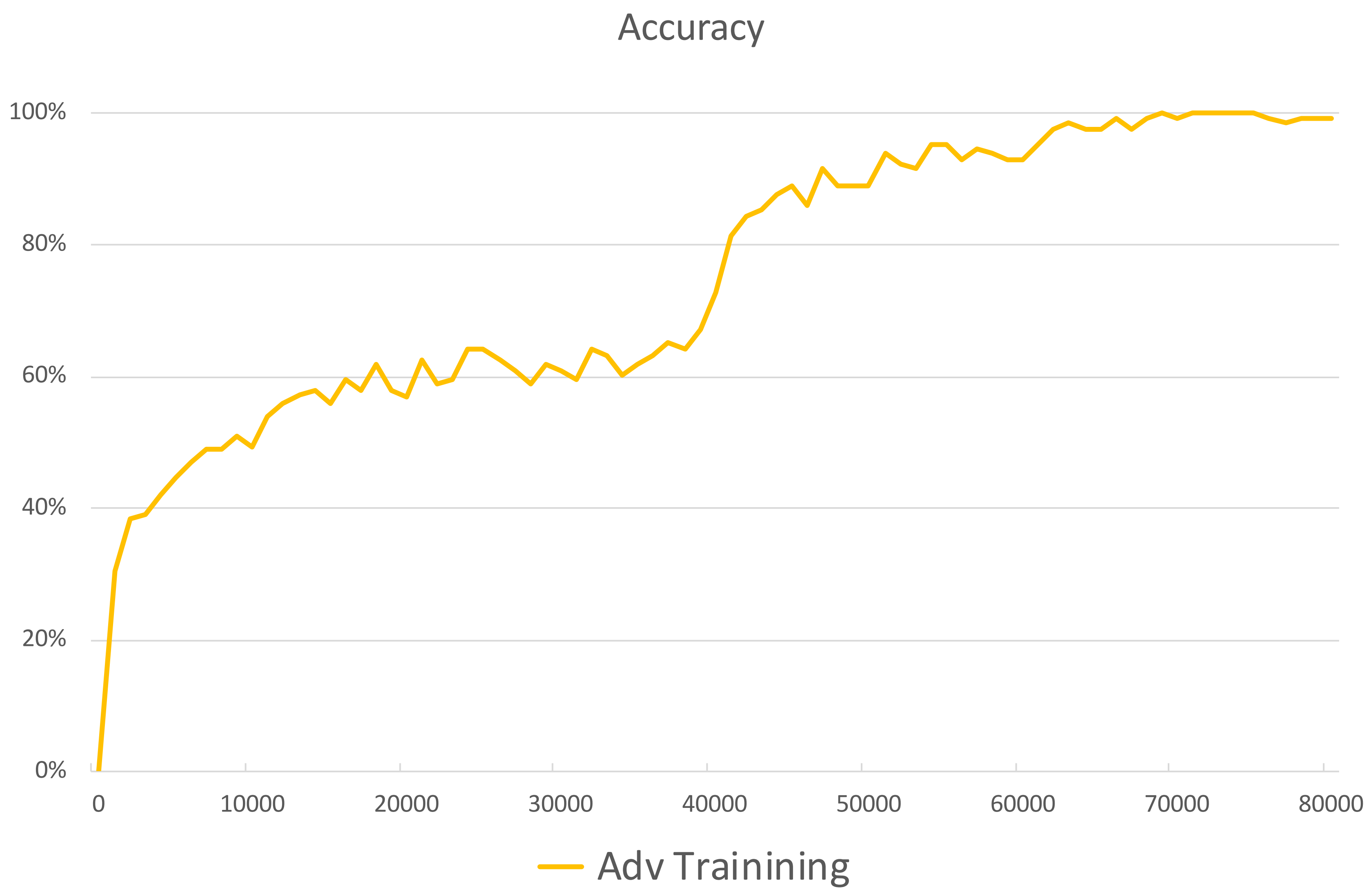


Do Robust Deep Networks Overfit?

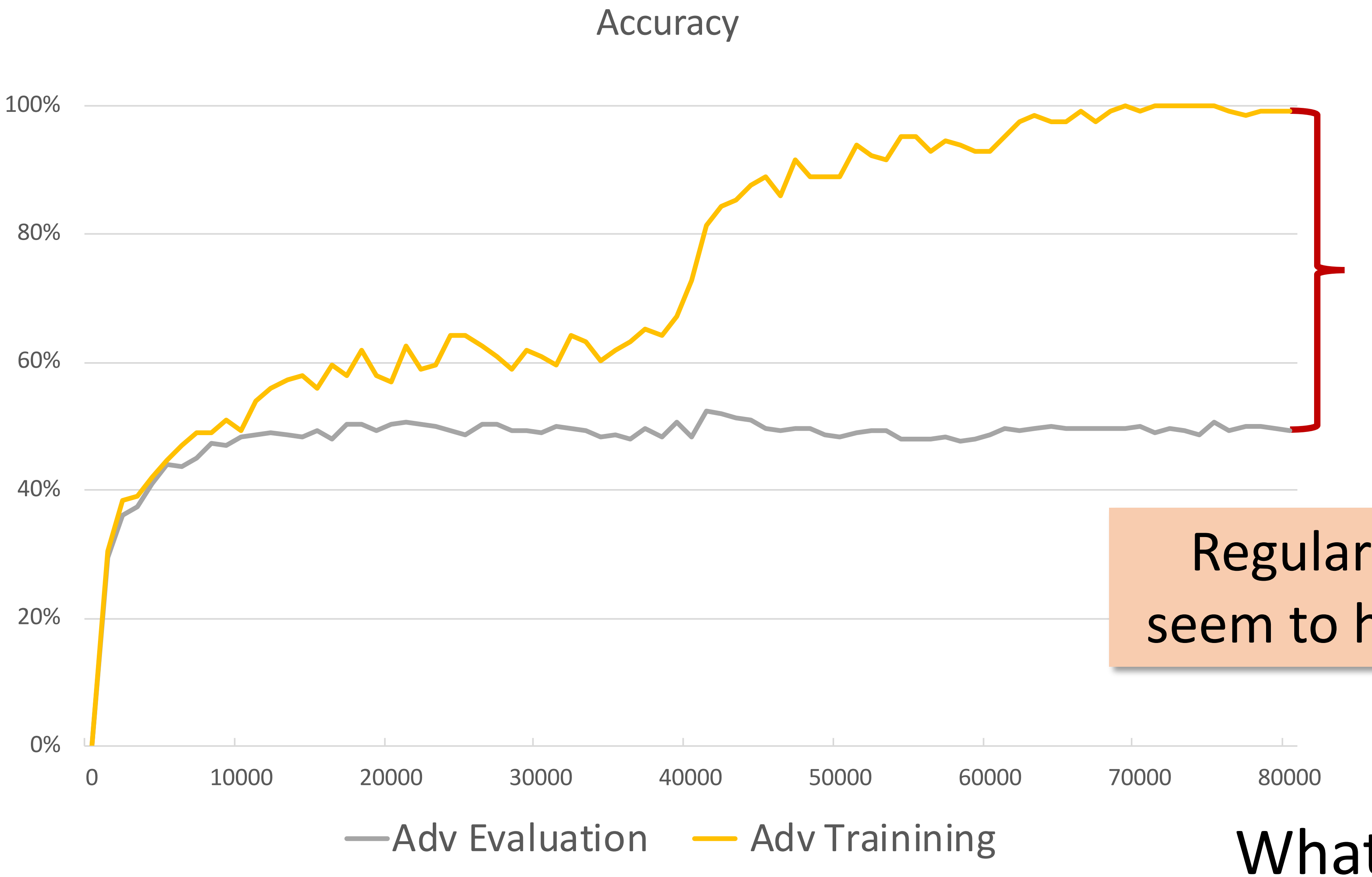


(small)
generalization gap

Do Robust Deep Networks Overfit?



Do Robust Deep Networks Overfit?



(large)
generalization gap

Regularization does not seem to help (much) either

What’s going on?

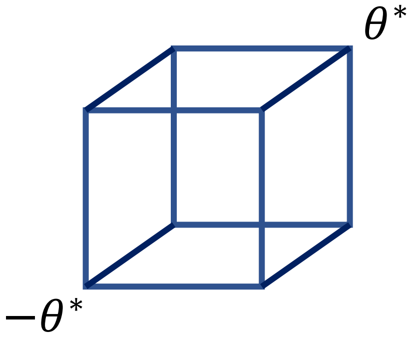
Adv. Robust Generalization Needs More Data

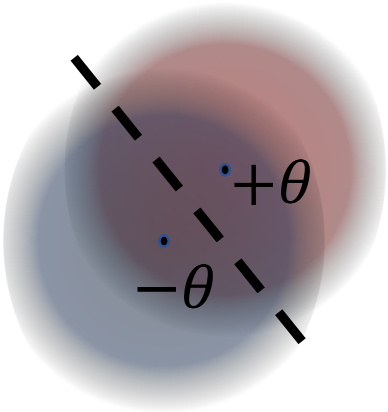
Theorem [Schmidt Santurkar Tsipras Talwar M 2018]:
Sample complexity of adv. robust generalization can be **significantly larger** than that of "standard" generalization

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

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

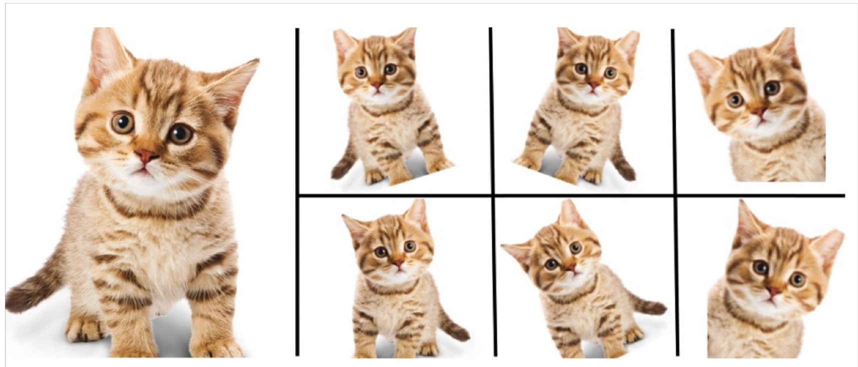
→ **But**: Need $\mathbf{\Omega}\left(\sqrt{\mathbf{d}}\right)$ samples for better-than-chance **robust** classifier





Does Being Robust Help “Standard” Generalization?

Data augmentation: An effective technique to improve “standard” generalization

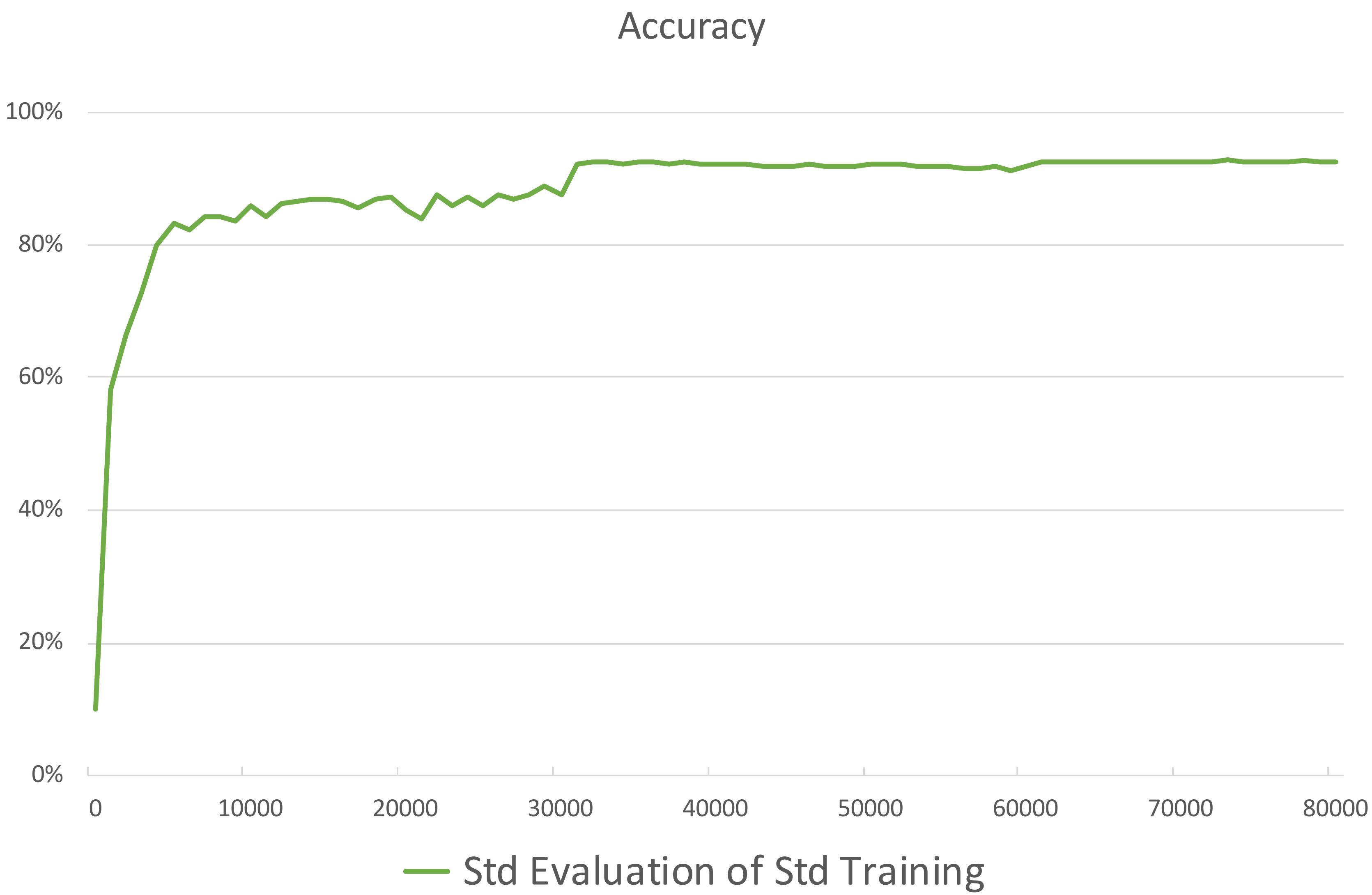


Adversarial training
=
An “ultimate” version of data augmentation?

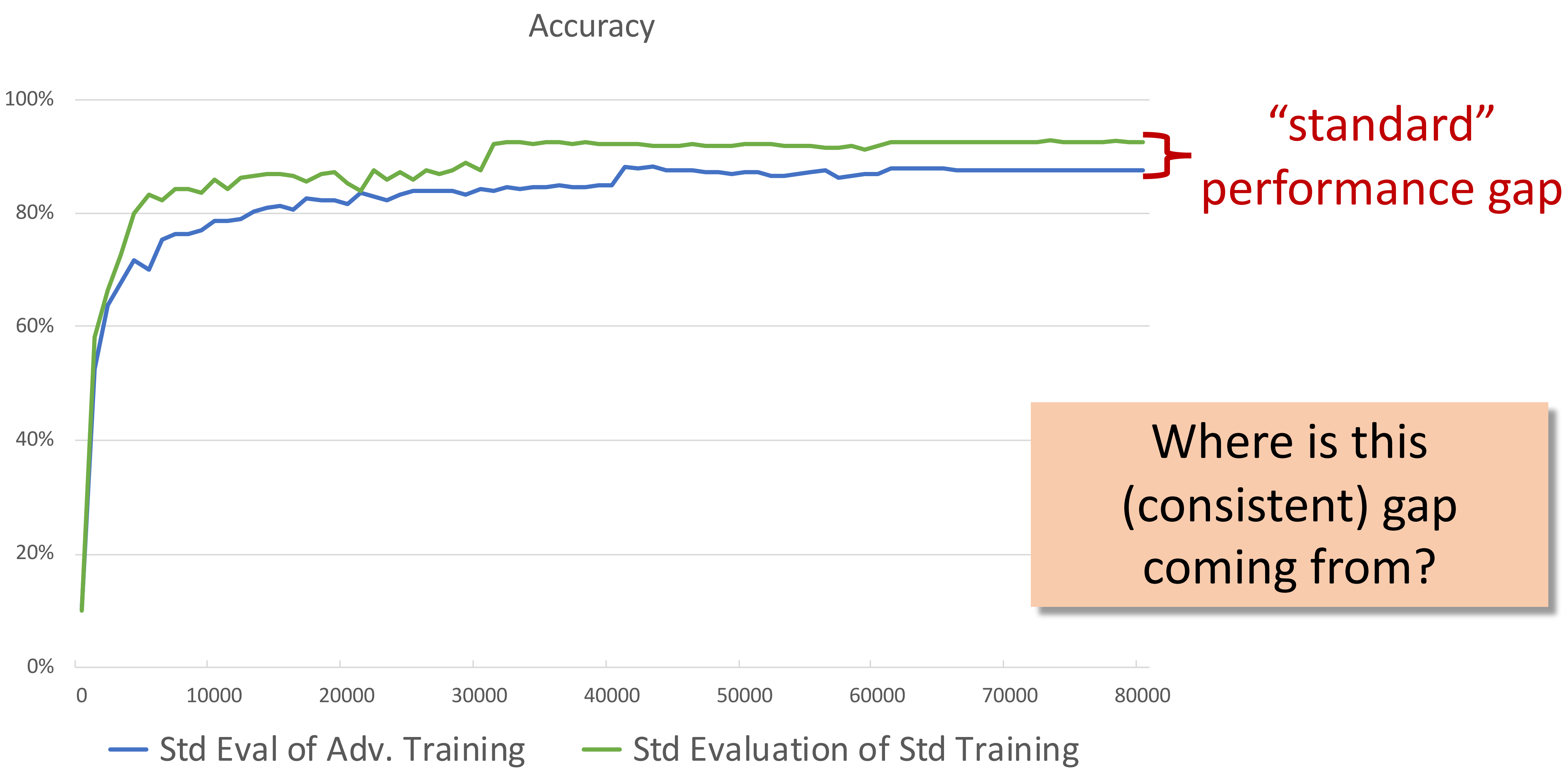
(since we train on the ”most confusing” version of the training set)

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

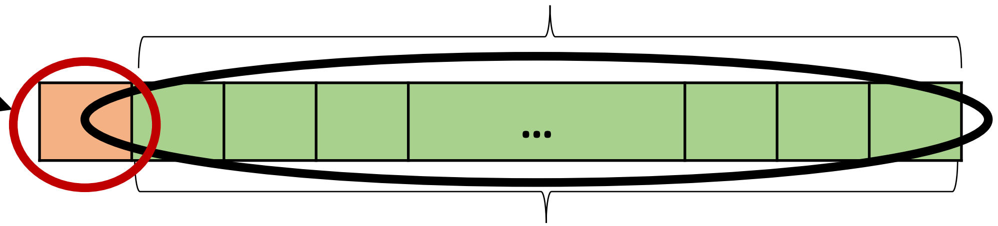
Basic intuition:

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

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

aggregates to a very accurate **(but non-robust!)** "meta-feature"

Strong (but not perfect) correlation

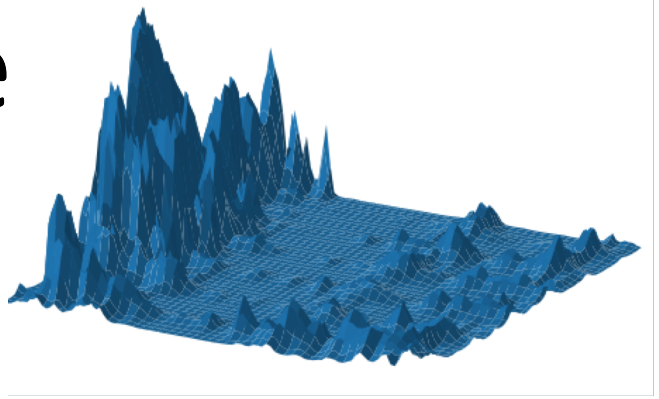


Weak correlation

Standard training: use all of features, maximize accuracy

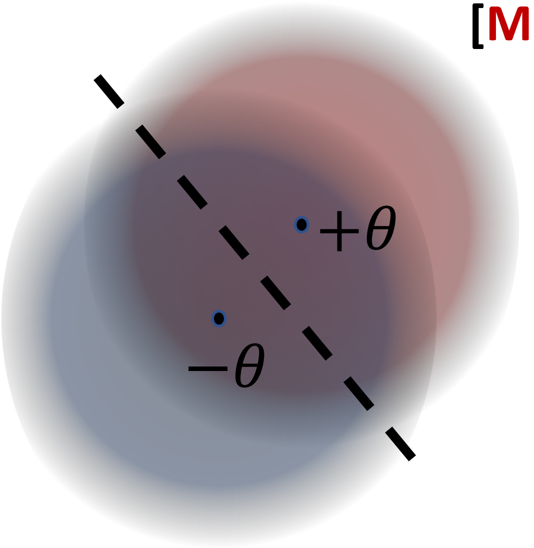
Adversarial training: use only single robust feature (**at the expense of accuracy**)

Adversarial Robustness is Not Free



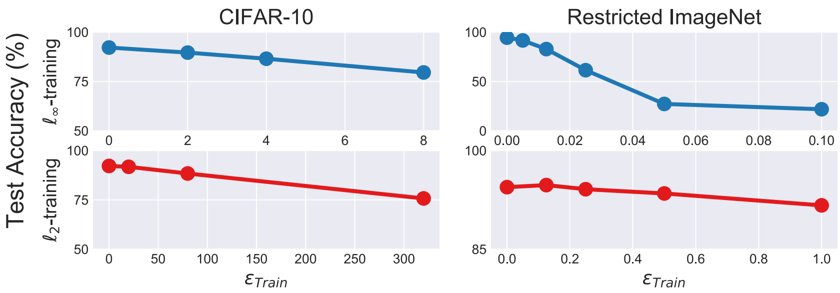
→ Optimization during training more difficult and models need to be larger

[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

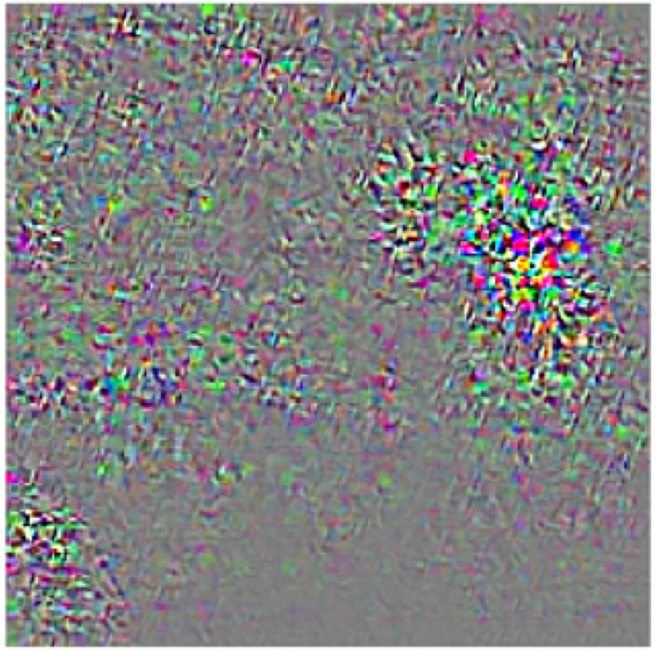
[Tsipras Santurkar Engstrom Turner M 2018] (Also see: **[Bubeck Price Razenshteyn 2018]**)

But There Are (Unexpected?) Benefits Too

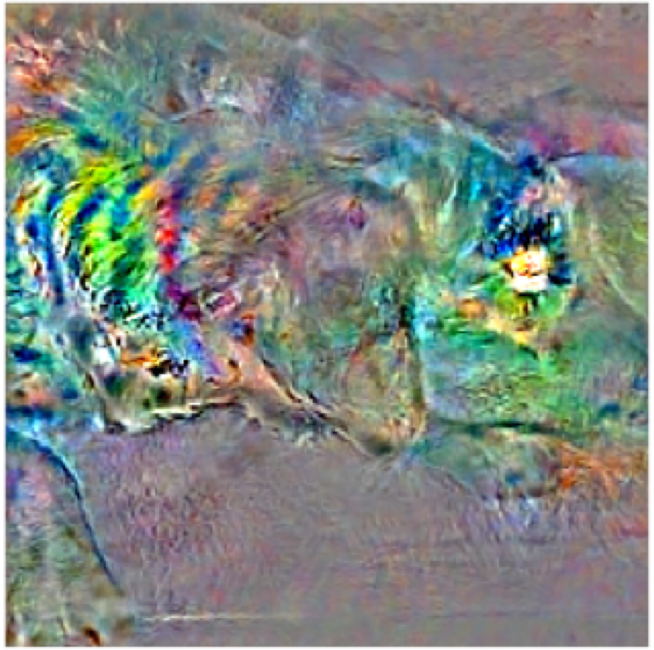
[Tsipras Santurkar Engstrom Turner M 2018]

Models become more **semantically meaningful**

Input



Gradient of standard model



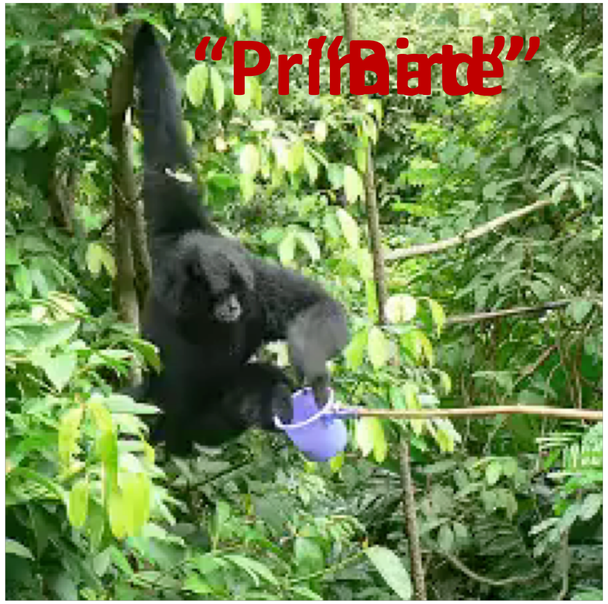
Gradient of adv. robust model

But There Are (Unexpected?) Benefits Too

[Tsipras Santurkar Engstrom Turner M 2018]

Models become more semantically meaningful

Standard model

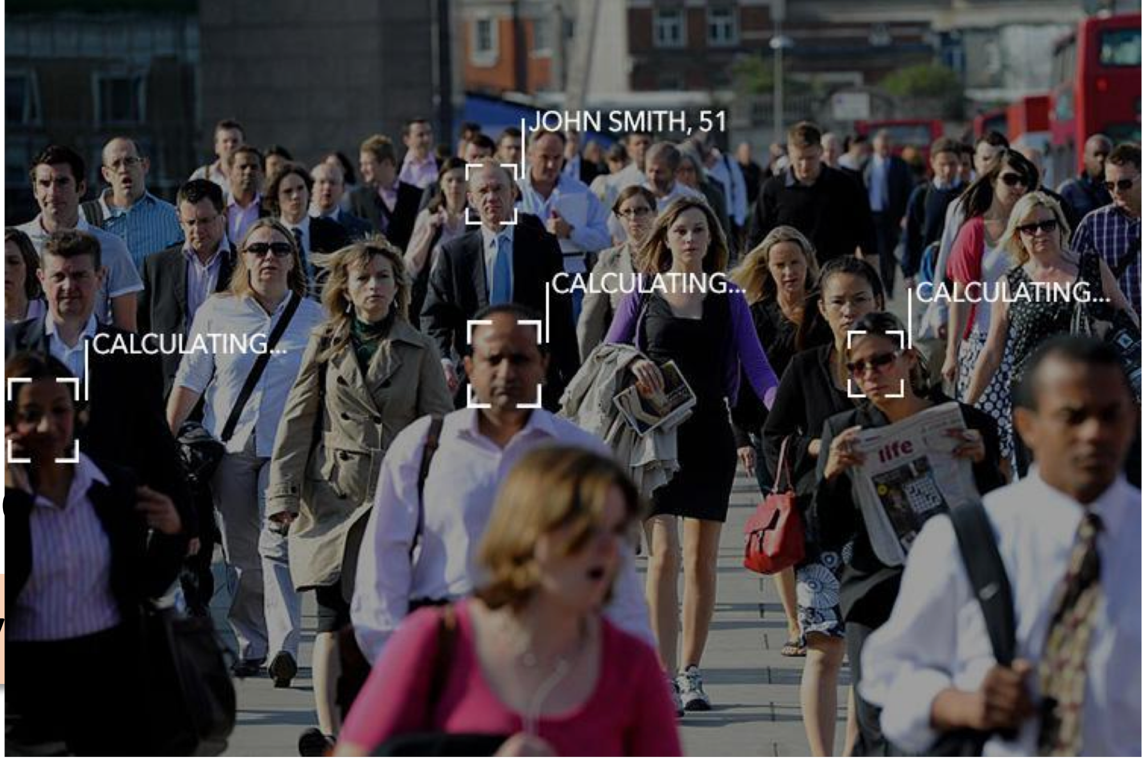


Adv. robust model

Conclusions

→ ML can play a big role in many domains
(and this is exciting!)

→ **But:** It is still Wild West out there
(we struck gold but there is lots of fool's g



Next frontier: Building ML y

We need to:

→ Attain a principled understanding of core techniques and tools

→ Rethink the whole pipeline from a robustness/safety/security perspective

Broader question:
Is ML human-ready?

Want to learn more? See **gradient-science.org** and **adversarial-ml-tutorial.org**

@aleks_madry

madry-lab.ml