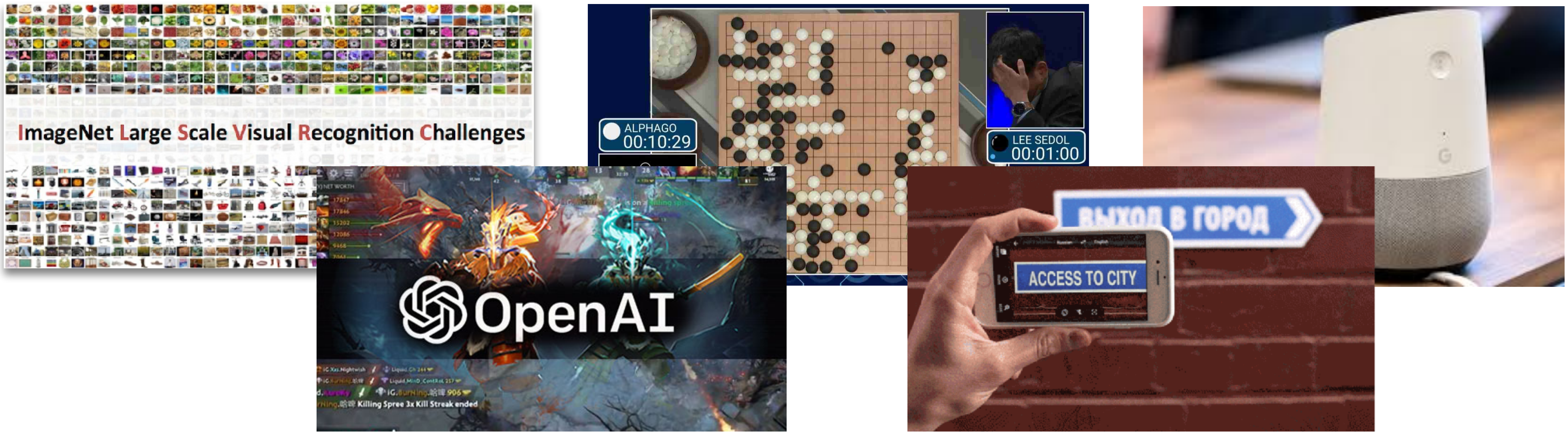
Adversarial Robustness: Theory, Practice, and Beyond

Aleksander Mądry

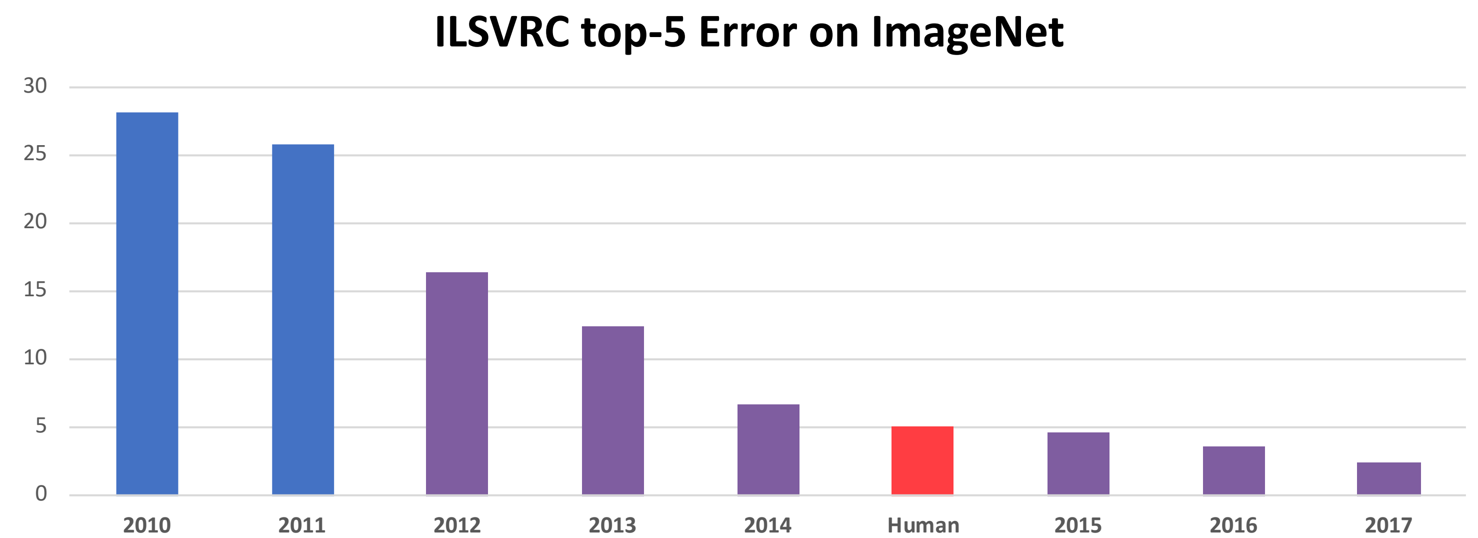
@aleks_madry

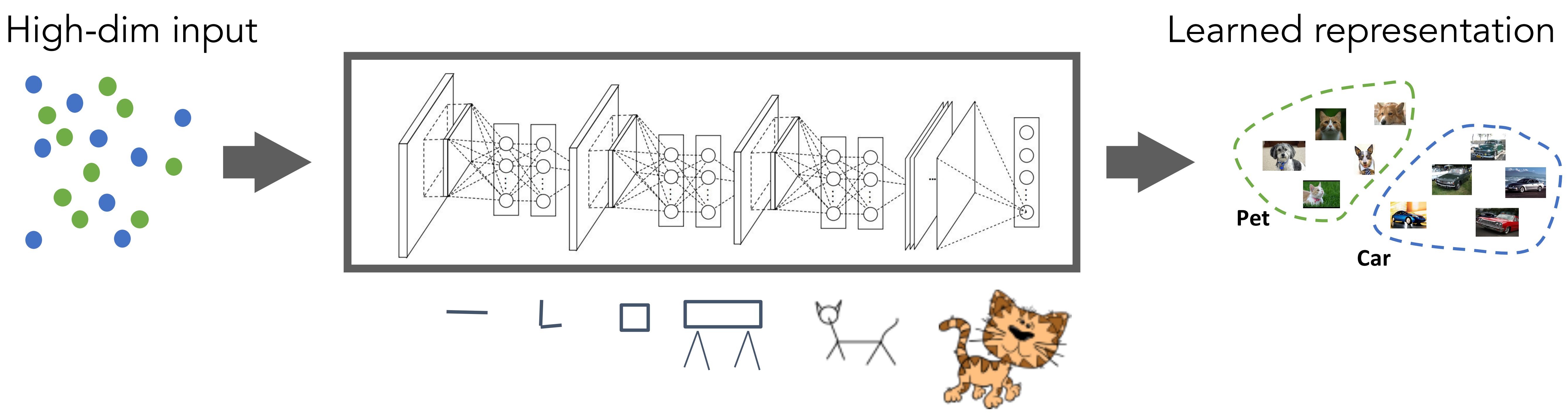
gradientscience.org

Why do we love deep learning?



Why do we love deep learning?

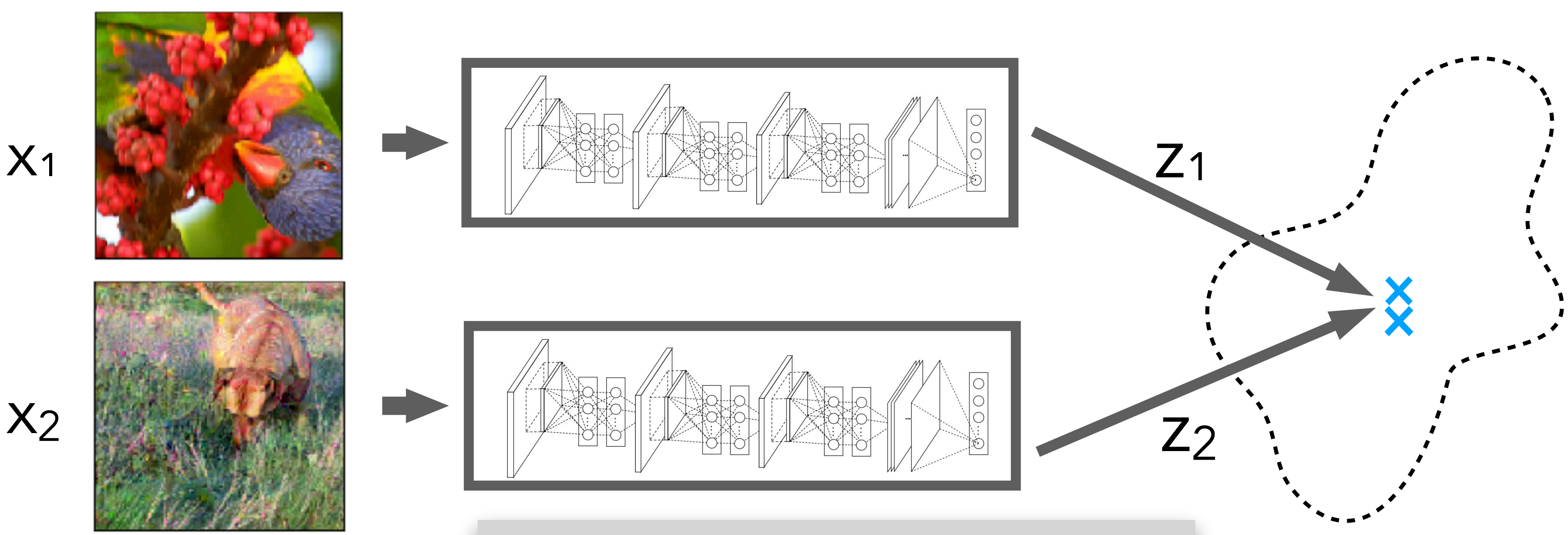




But…

Correct label: insect

Predicted label: dog



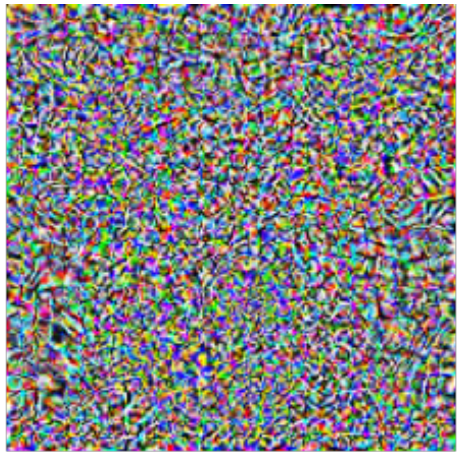
$\mathbf{x_1} \neq \mathbf{x_2}$ but $\mathbf{z_1} \approx \mathbf{z_2}$

What’s going on?

Key Problem: Adversarial Perturbations

[Szegedy et al 2013] [Biggio et al 2013]

+ 0.005 x



=

"pig" (91%) **noise (NOT random)** **"airliner" (99%)**

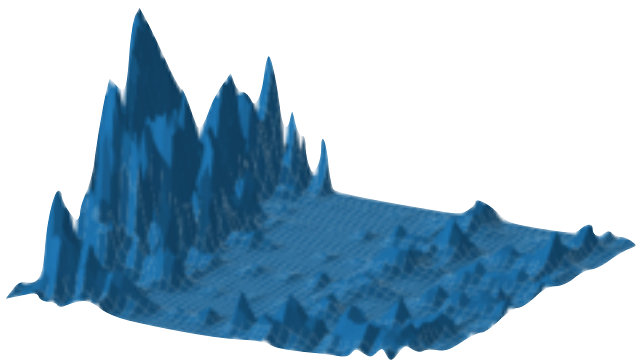
Emerging goal: (Adversarially) robust generalization

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D}[\max_{\delta\in\Delta} \ell(\theta; x+\delta, y)]$$

Desired invariance

→ We are (finally) starting to succeed here

ML via Adversarial Robustness Lens



- Training is harder and models need to be more complex

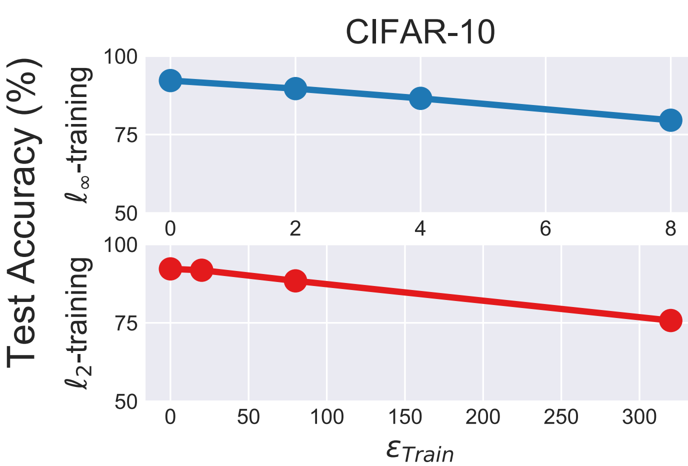
  [M Makelov Schmidt Tsipras Vladu 2018]

- Models may <u>have</u> to be less accurate

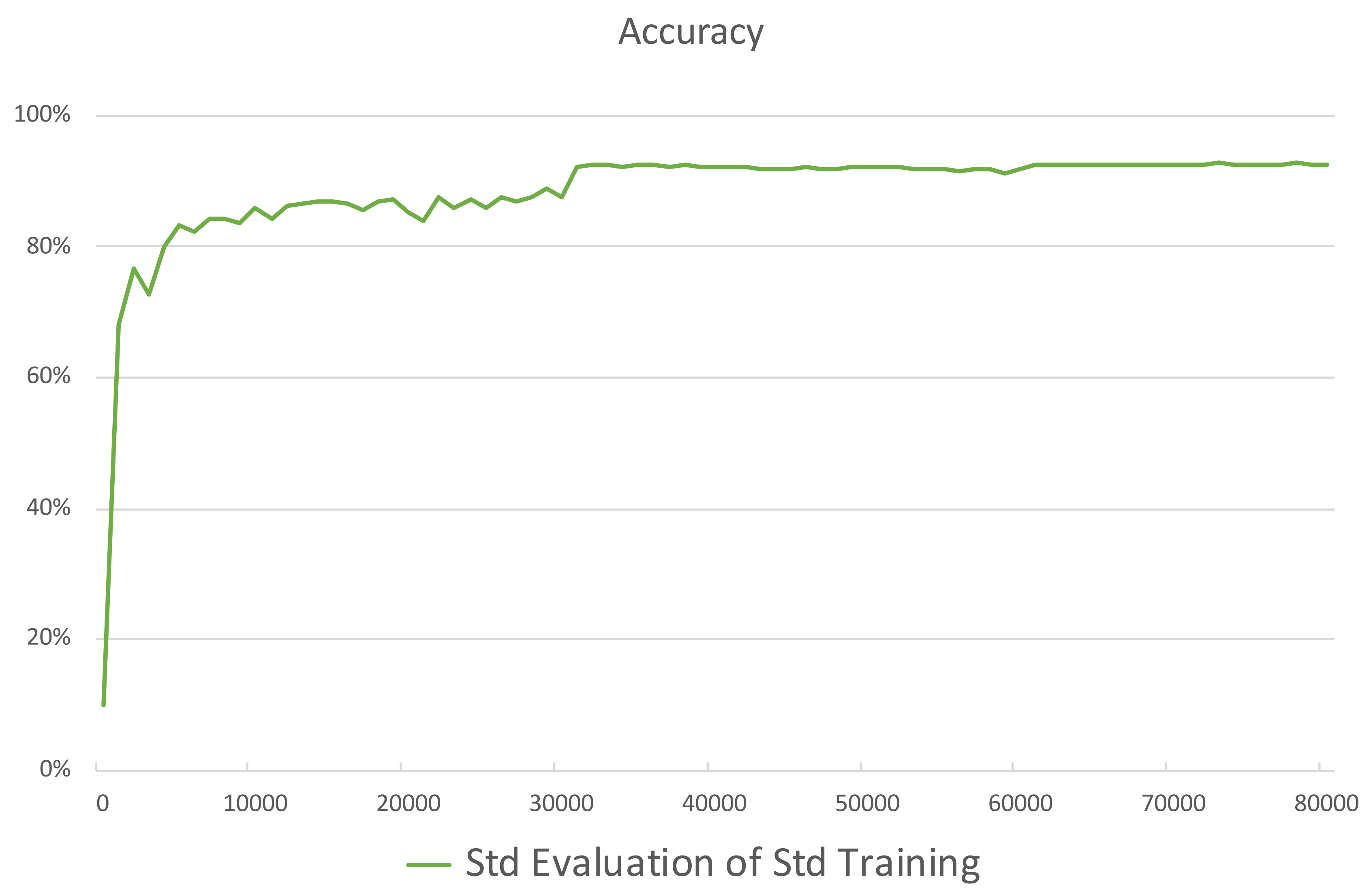
  [Tsipras Santurkar Engstrom Turner M 2018]

  [Bubeck Price Razenshteyn 2018]

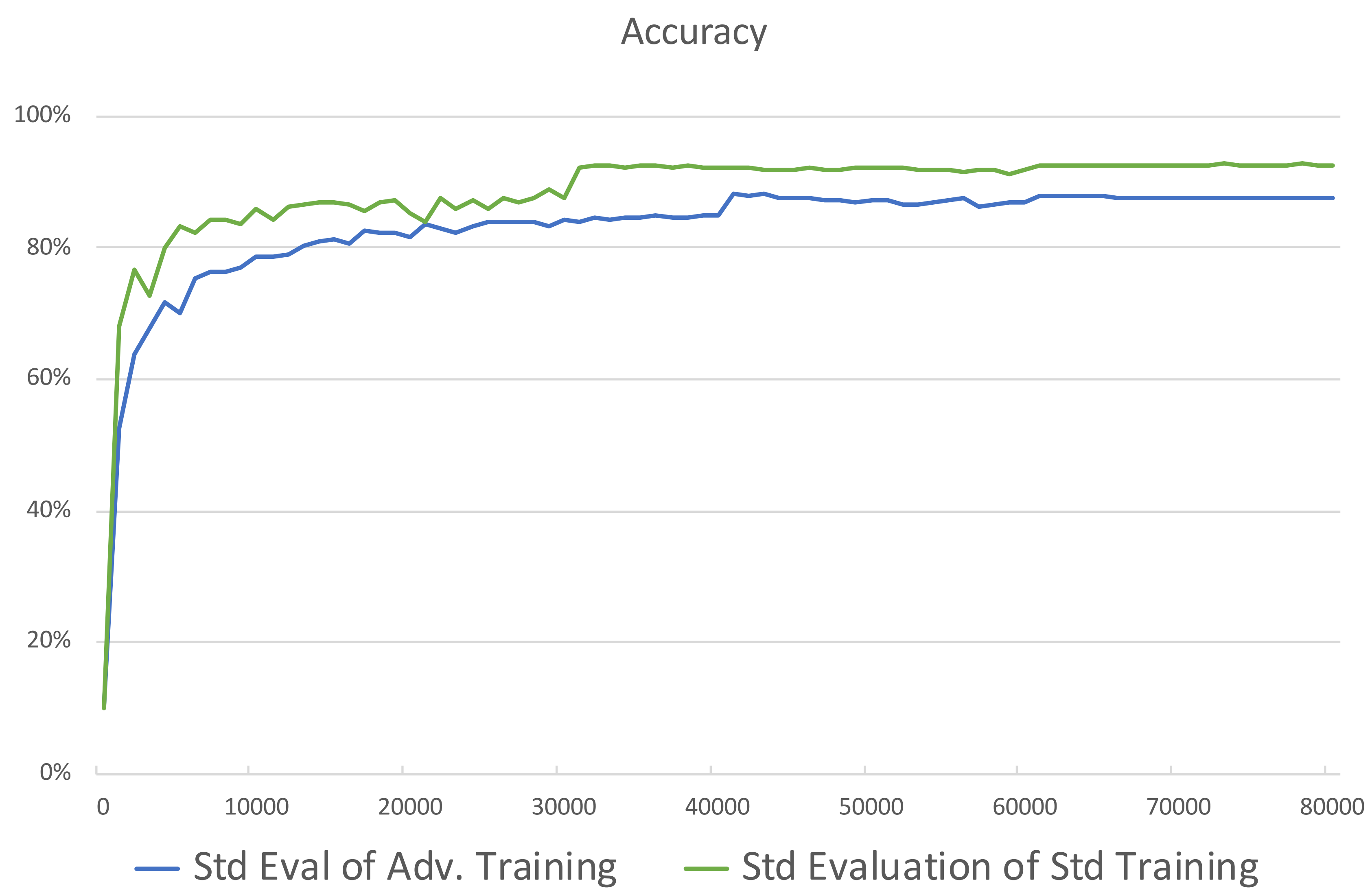
  [Degwekar Nakkiran Vaikunatanathan 2018]



Standard Generalization of Robust Models



Standard Generalization of Robust Models

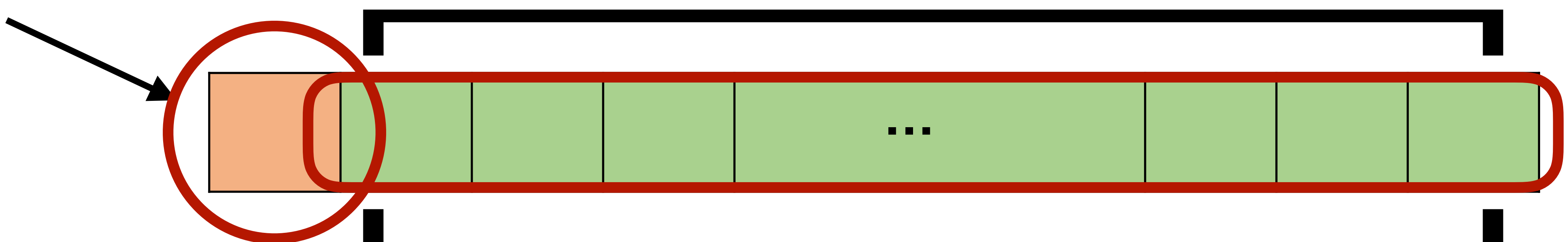


Standard Generalization of Robust Models

Theorem **[Tsipras Santurkar Engstrom Turner M 2018]:**
There exist distributions such that:
best ℓ_∞-robust accuracy **<< best** standard accuracy

Strong (but far from perfect) correlation

Many **independent** weak correlations

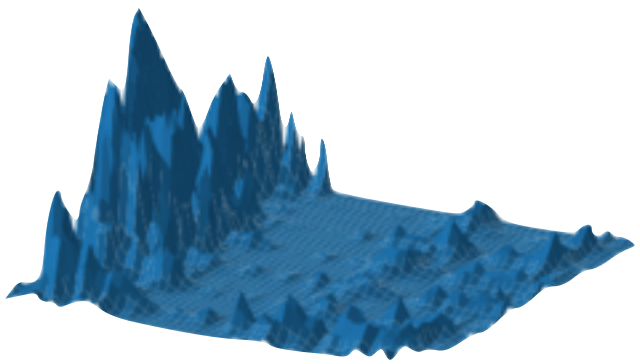


Aggregates to a **near-perfect** (but **non**-robust**)** "meta-feature"

→ **To maximize standard accuracy:** Rely on the meta-feature

→ **To be robust:** Need to focus on the single (imperfect) feature

ML via Adversarial Robustness Lens



- Training is harder and models need to be more complex

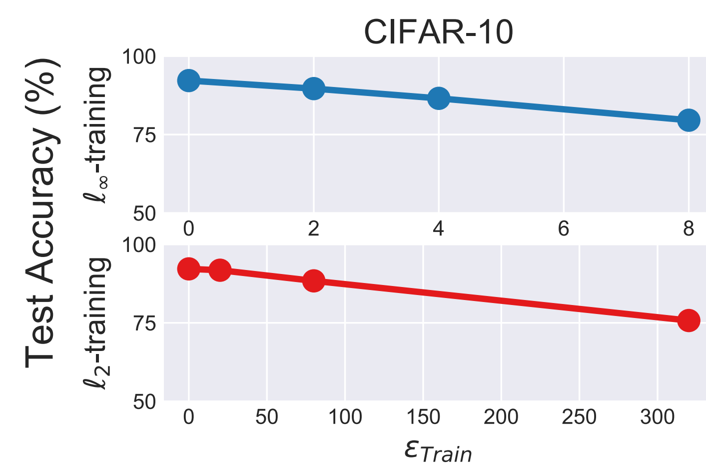
  [M Makelov Schmidt Tsipras Vladu 2018]

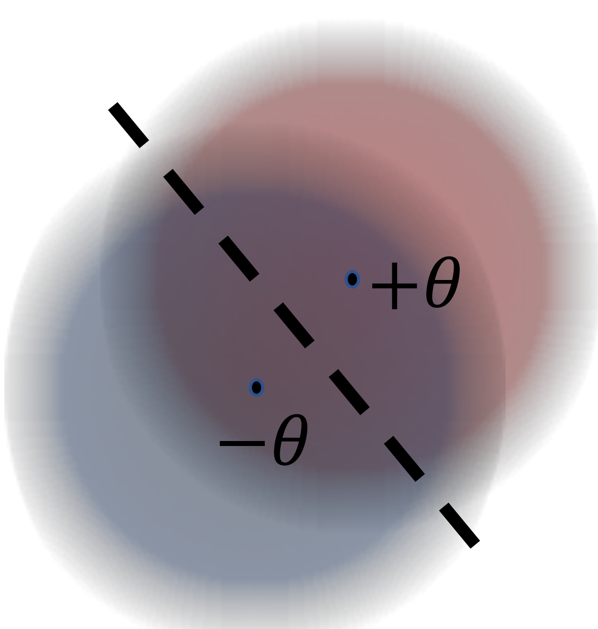
- Models may <u>have</u> to be less accurate

  [Tsipras Santurkar Engstrom Turner M 2018]

  [Bubeck Price Razenshteyn 2018]

  [Degwekar Nakkiran Vaikunatanathan 2018]





- We might need more training data

  [Schmidt Santurkar Tsipras Talwar M 2018]

Sample-Complexity of Robust Generalization

Theorem [Schmidt Santurkar Tsipras Talwar M 2018]:
There exist distributions for which we need **significantly** more samples to get a robust classifier

Specifically: There exists a **d**-dimensional distribution $\mathscr{D}$ such that:

→ A **single** sample from $\mathscr{D}$ enables us to get a classifier **C** s.t.

$$Pr_{(x,y)\in\mathscr{D}}[C(x) = y] > 0.99$$

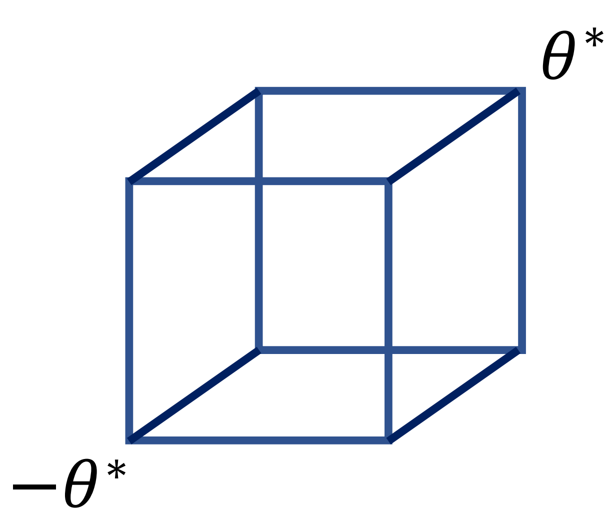
→ **But:** Without seeing $\Omega(\sqrt{d})$ samples from $\mathscr{D}$, we **cannot** find **C** s.t.

$$Pr_{(x,y)\in\mathscr{D}}[C(x+\delta) = y,\ \textbf{for all } \delta \in \Delta] > \frac{1}{2} + O(d^{-1}),$$

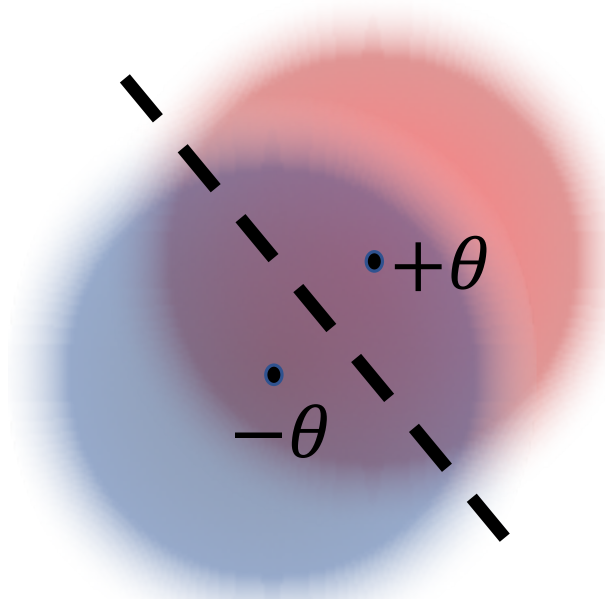
Sample-Complexity of Robust Generalization

Theorem **[Schmidt Santurkar Tsipras Talwar M 2018]:**
There exist distributions for which we need **significantly** more samples to get a robust classifier

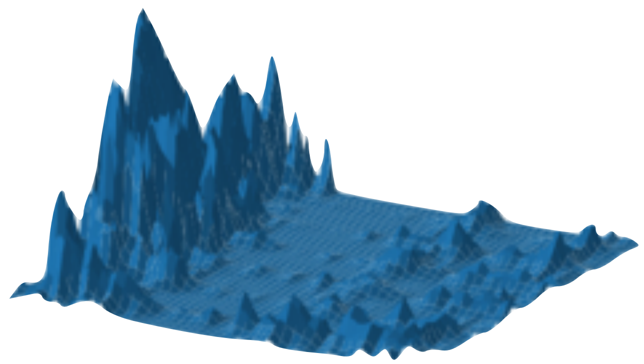
For <u>linear</u> classifiers:
Use a "noisy" hypercube vertex sampling



For <u>general</u> classifiers:
Use overlapping Gaussians



ML via Adversarial Robustness Lens



- Training is harder and models need to be more complex

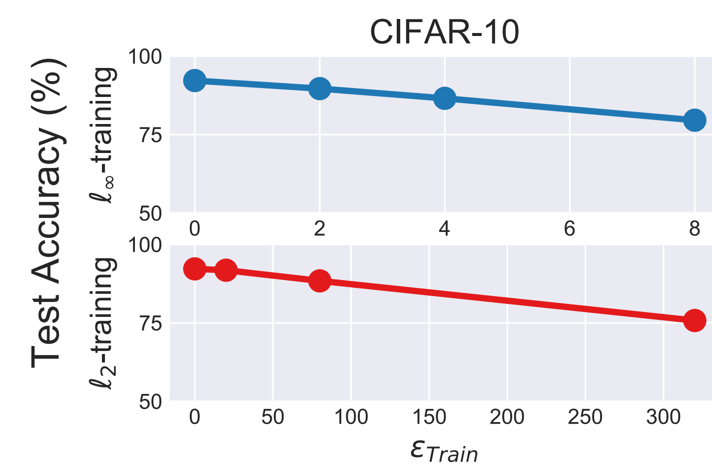
  [M Makelov Schmidt Tsipras Vladu 2018]

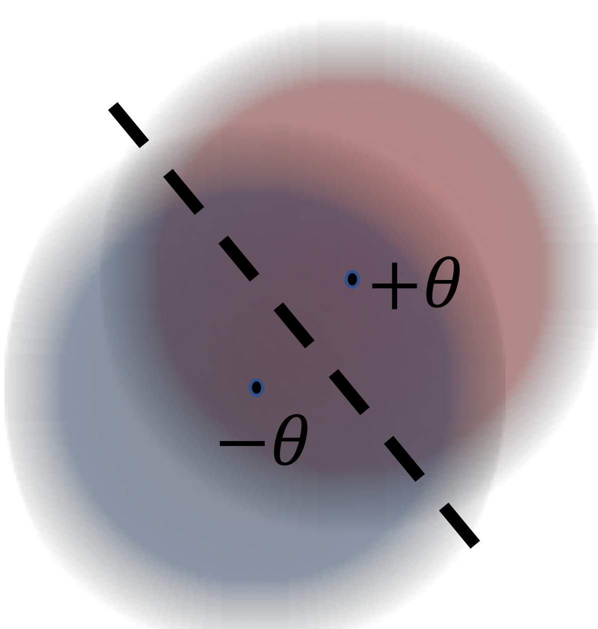
- Models may <u>have</u> to be less accurate

  [Tsipras Santurkar Engstrom Turner M 2018]

  [Bubeck Price Razenshteyn 2018]

  [Degwekar Nakkiran Vaikunatanathan 2018]





- We might need more training data

  [Schmidt Santurkar Tsipras Talwar M 2018]

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

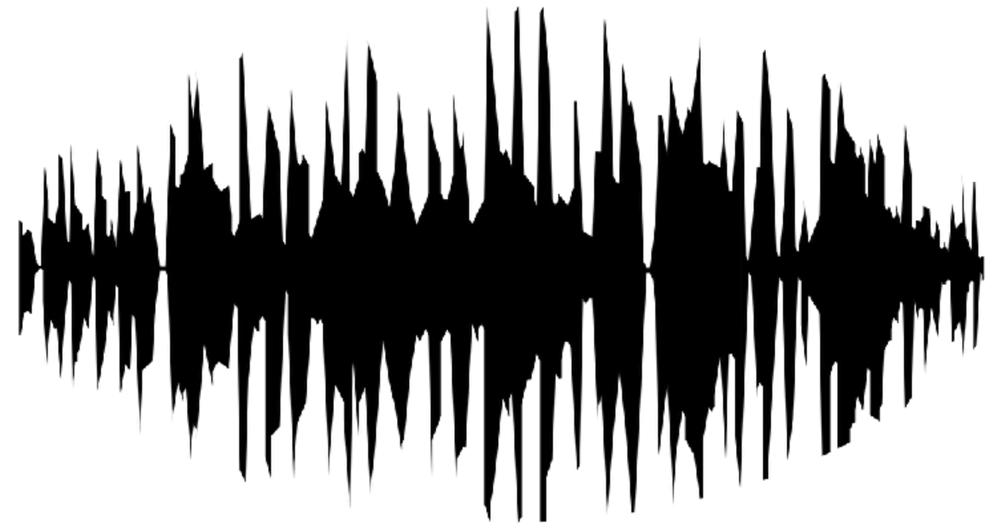
Why adversarial perturbations **exist**
(and **are so widespread**)?

Why these perturbations tend to **transfer**?

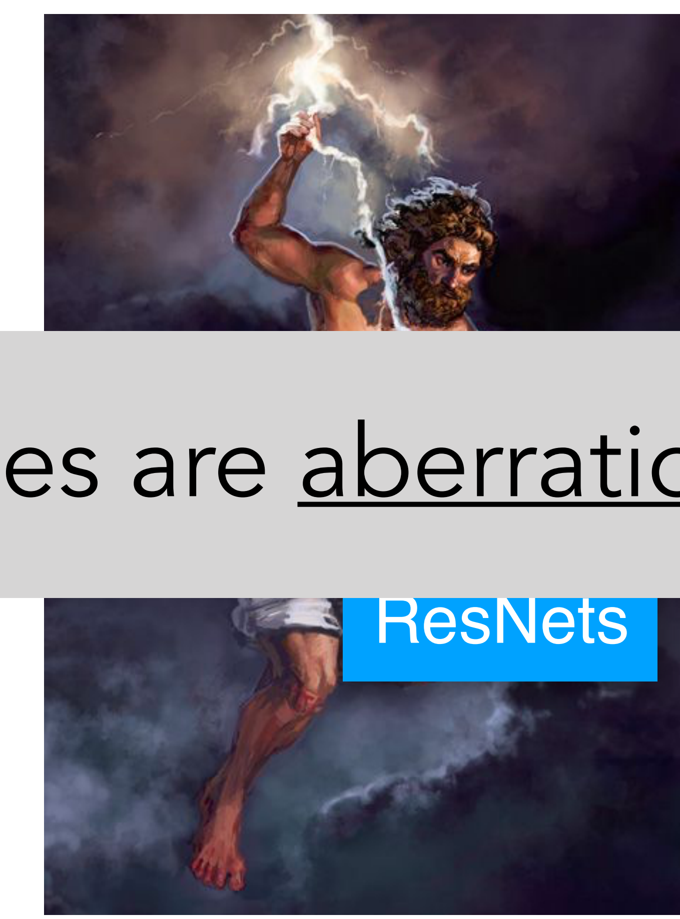
Why **robust training** works?

Why **randomized smoothing** works?

$d \to \infty$

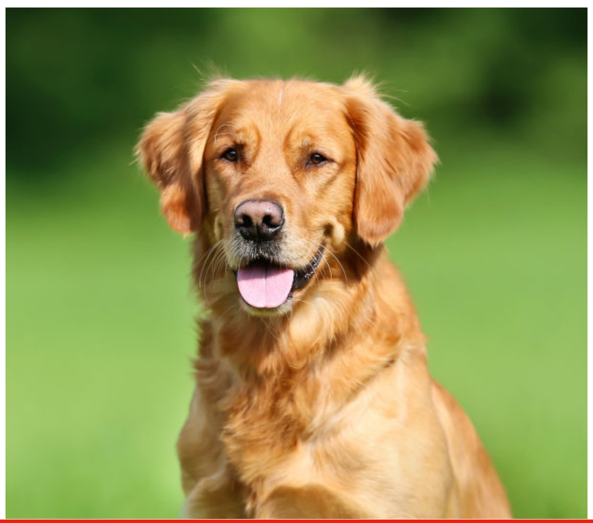
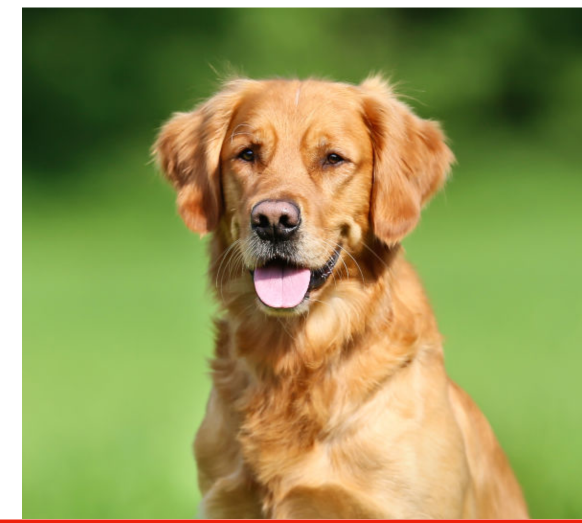


Why are our models brittle?



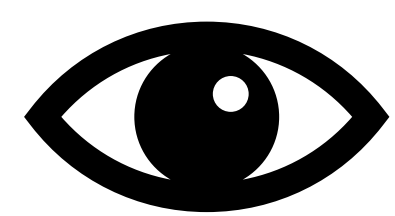
Unifying theme: Adversarial examples are <u>aberrations</u>

Why Are Adv. Perturbations Bad?

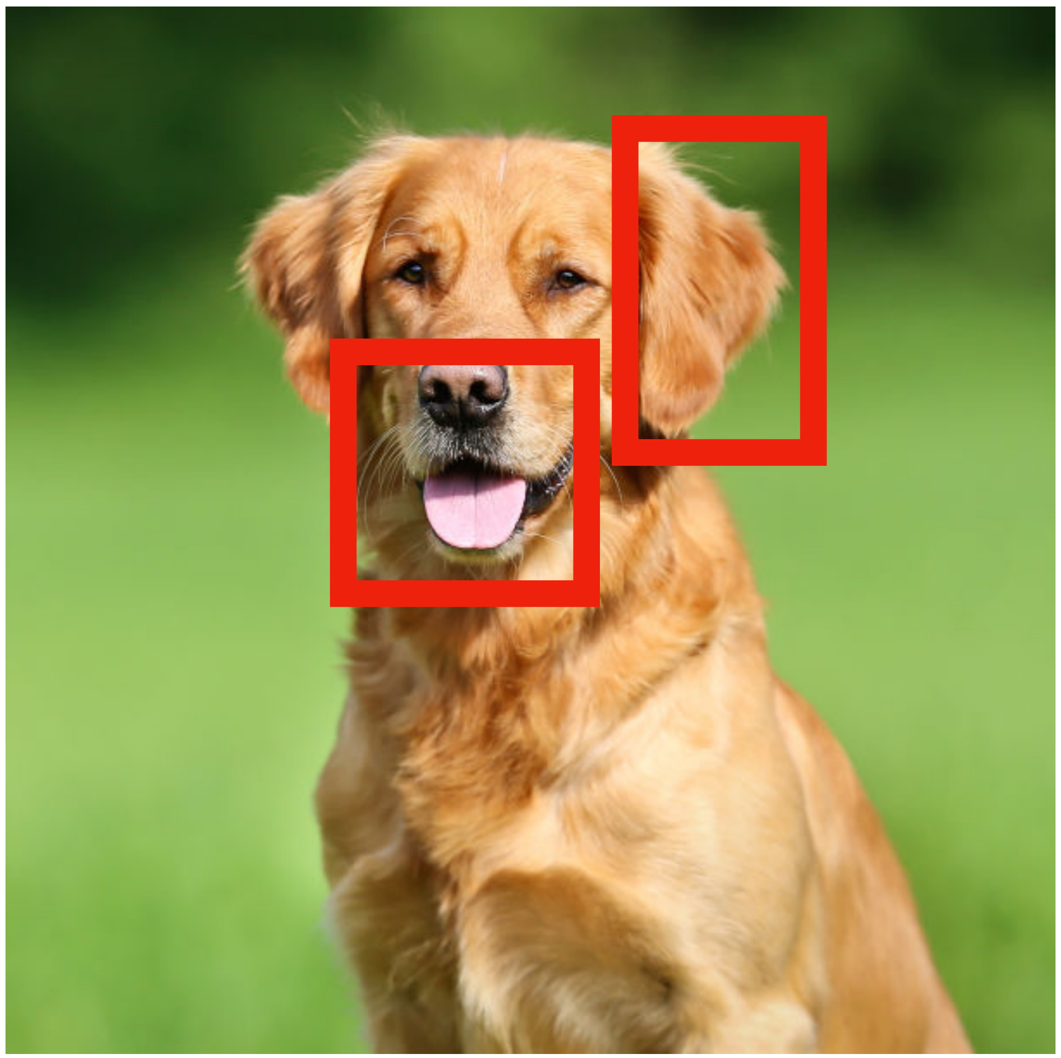
 +  = 

dog	meaningless perturbation	cat

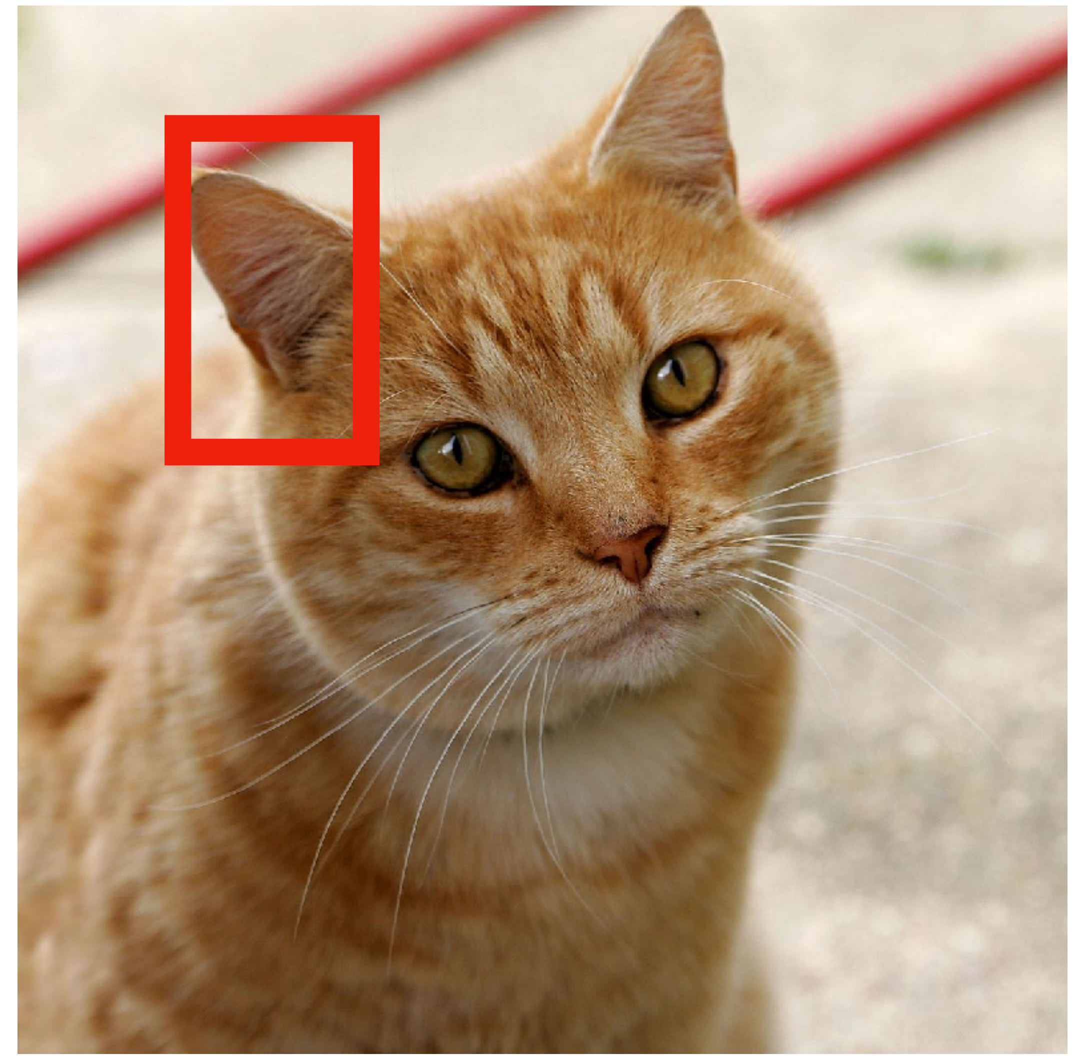
But: This is only a "human" perspective



Human Perspective



dog



cat

ML Perspective

dog

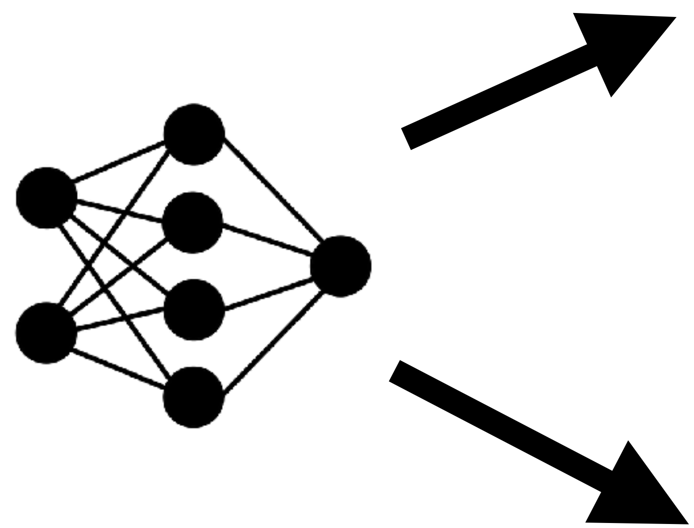


Image is meaningless

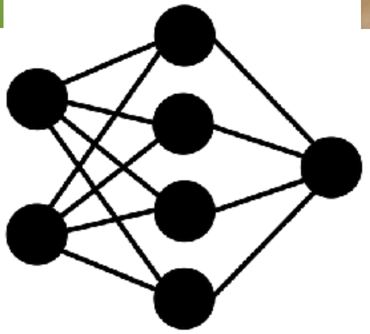
Classes are meaningless

Only goal:

Max (test) accuracy

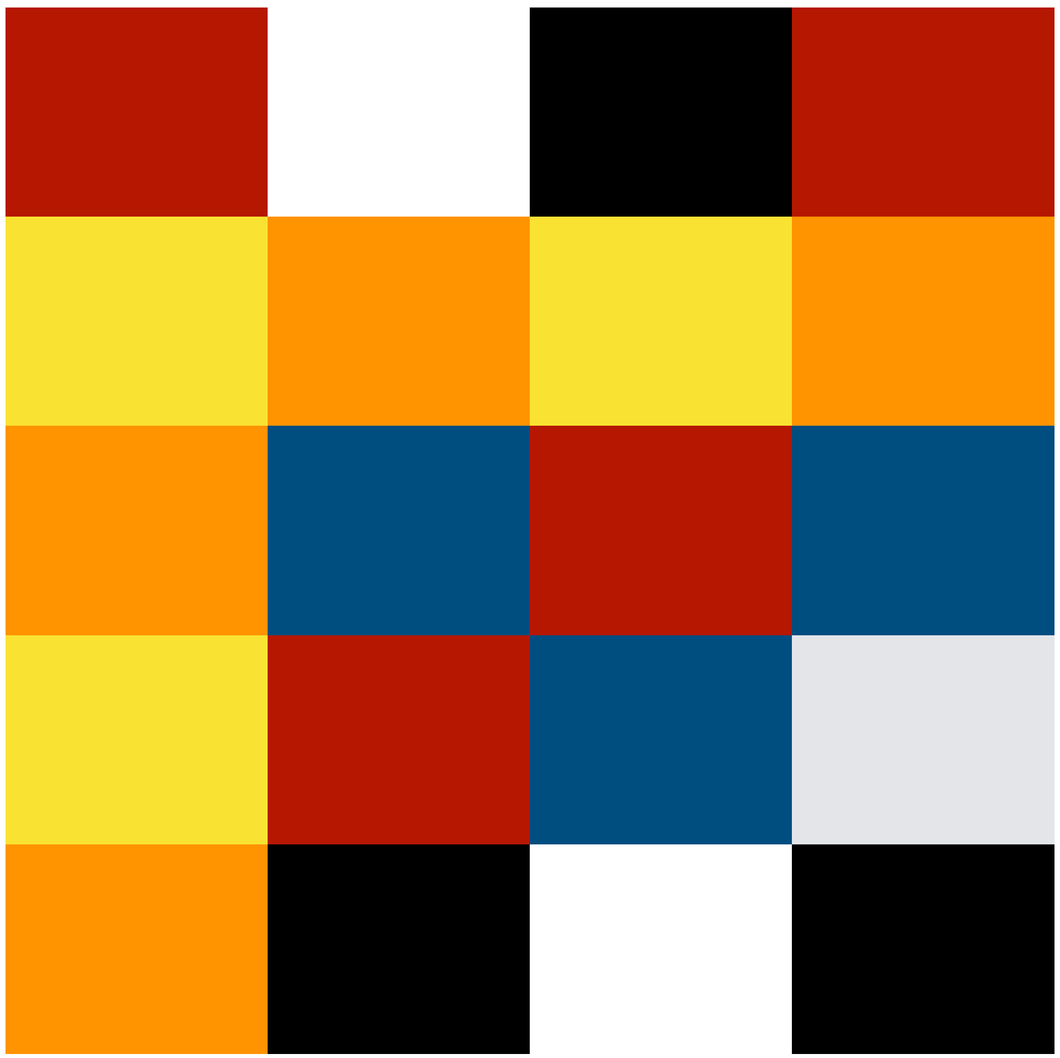
ML Perspective

dog

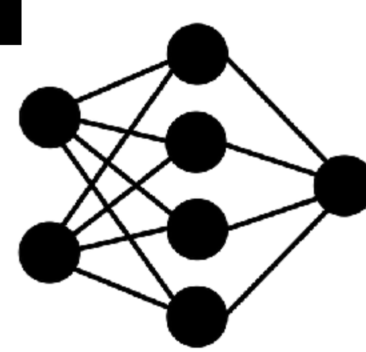


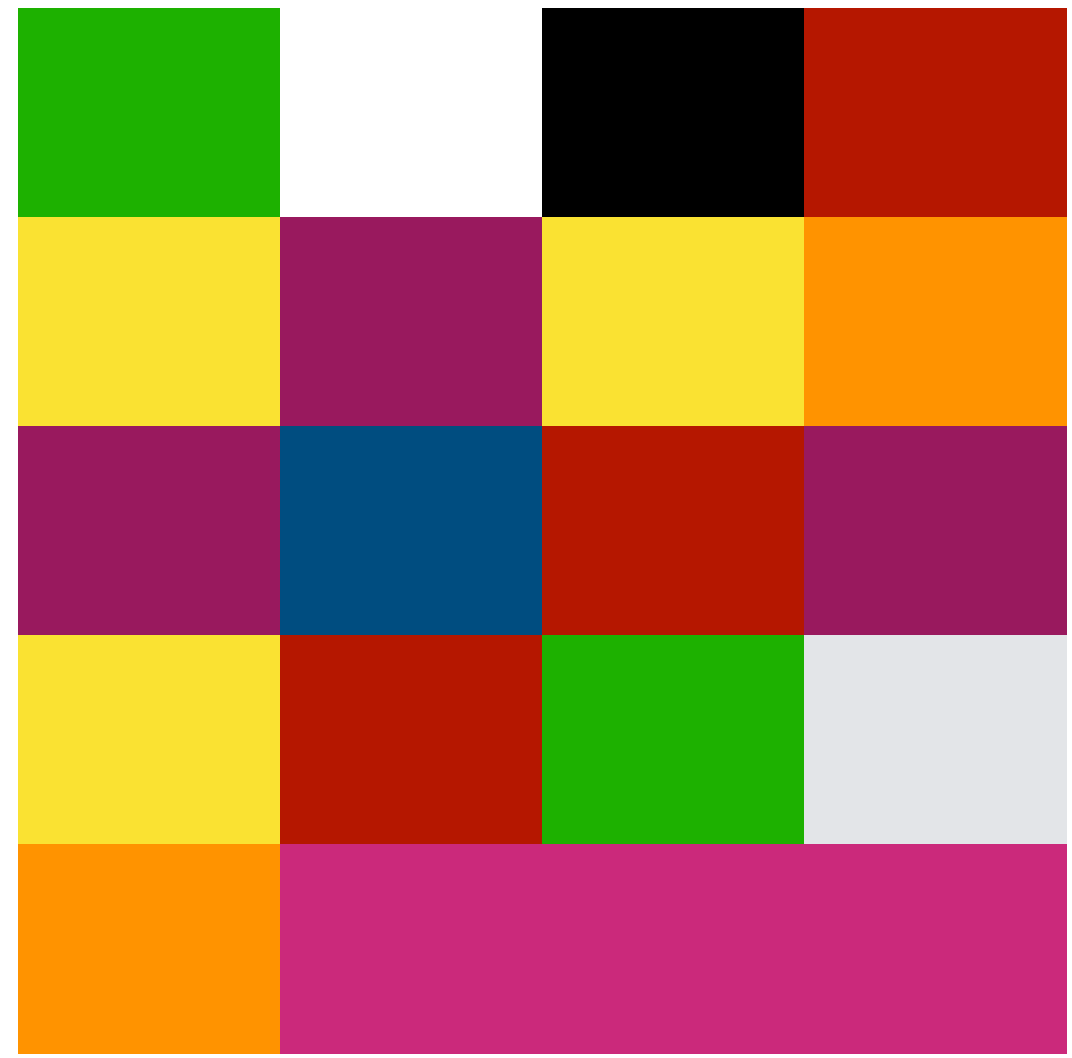
cat

ML Perspective



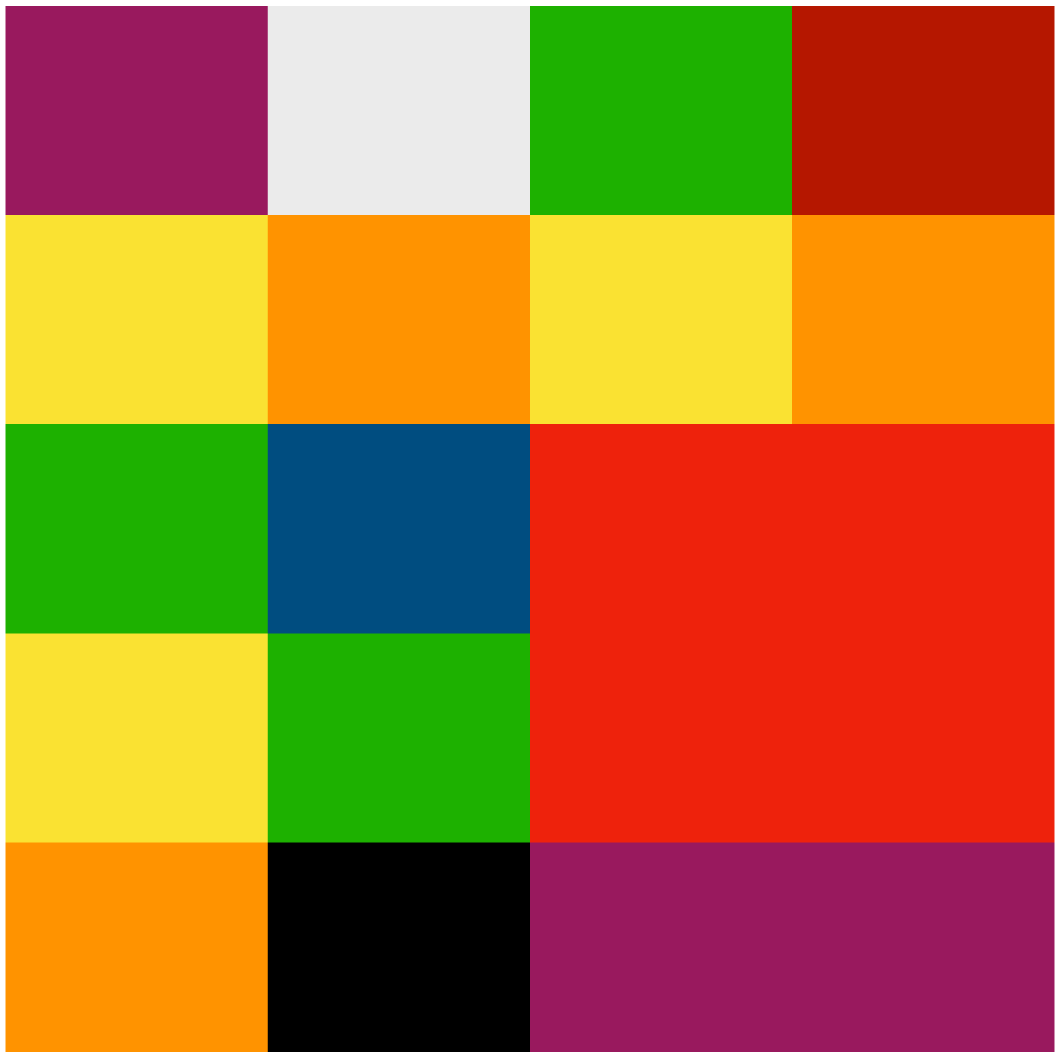
tap



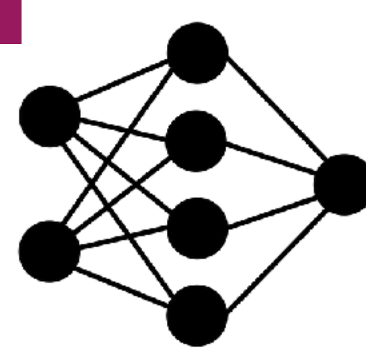


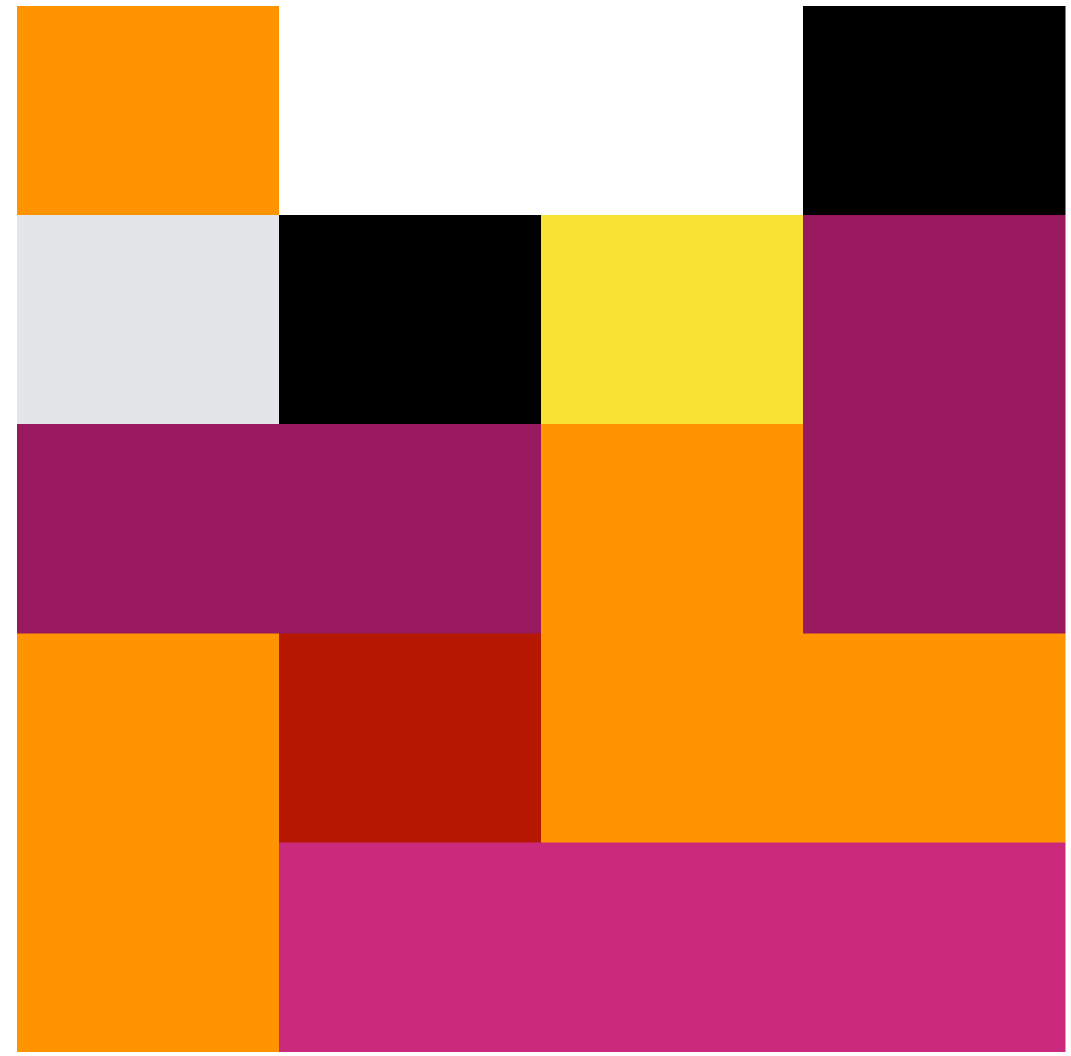
toc

ML Perspective



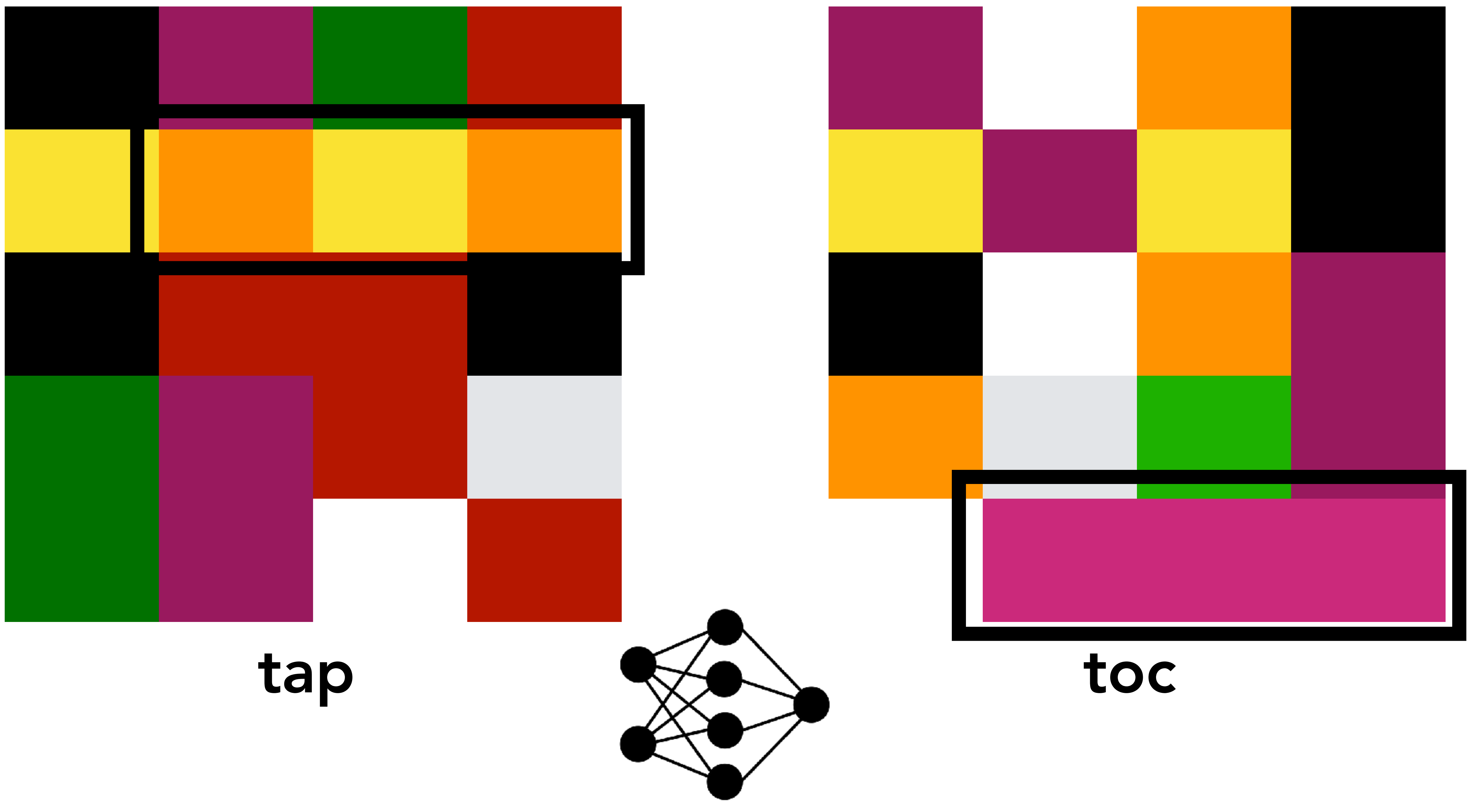
tap



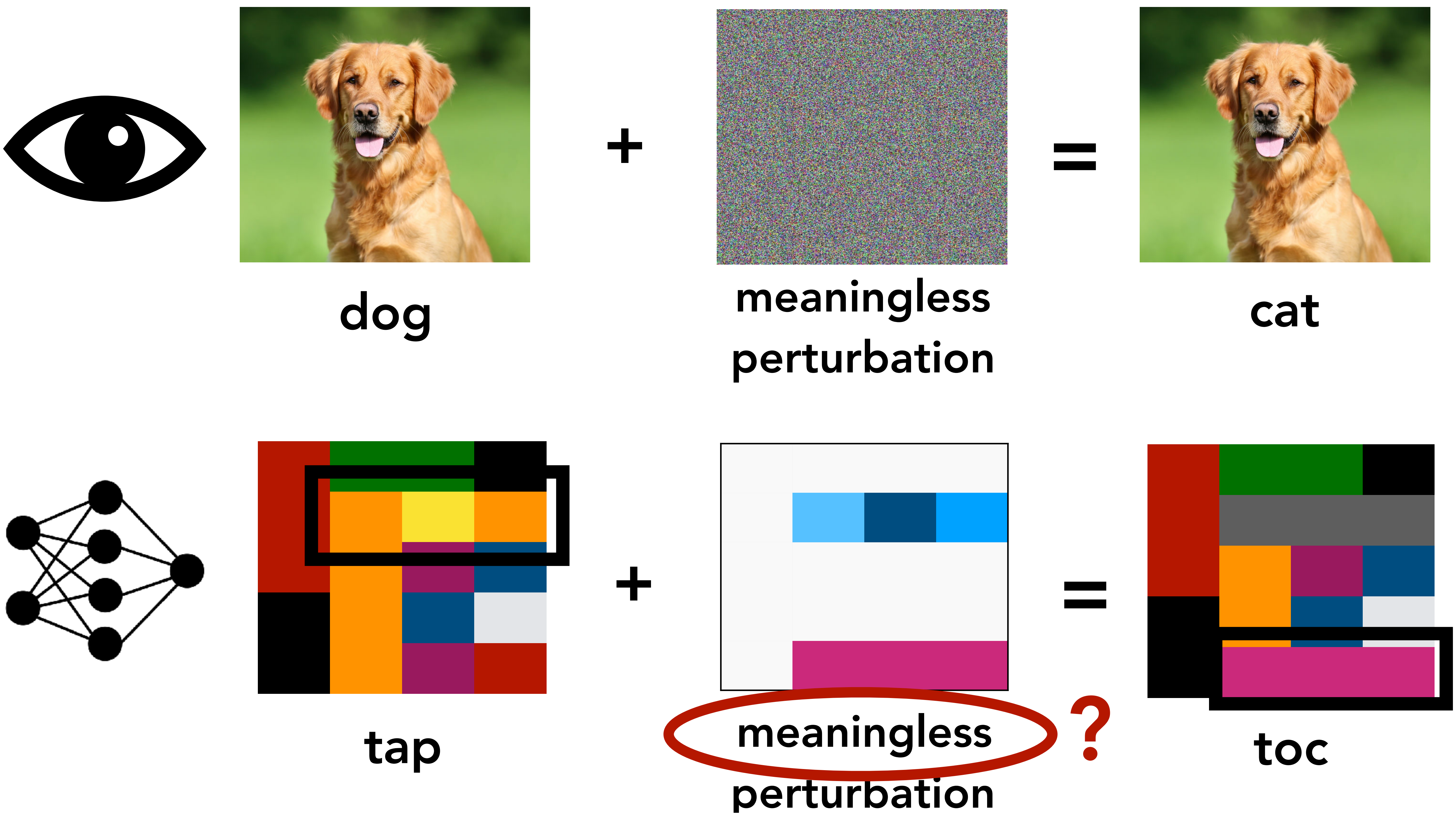


toc

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 **original** 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)

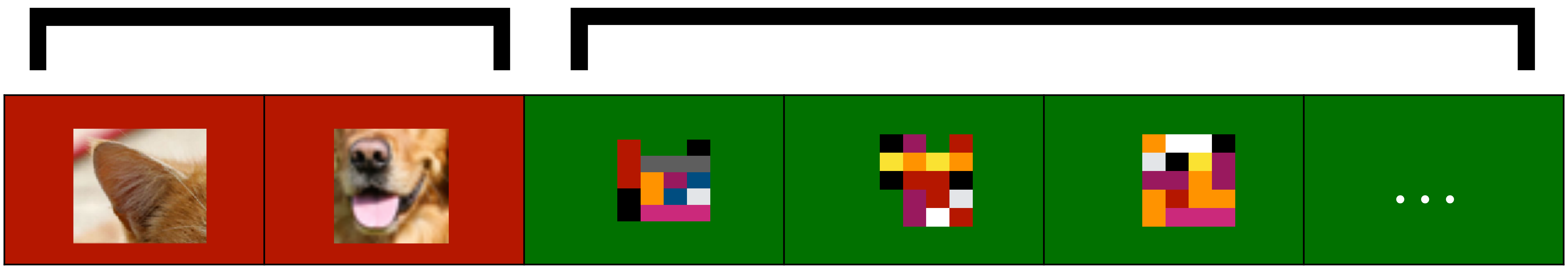
What's going on?

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

(Original) 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

A Simple Theoretical Setting:

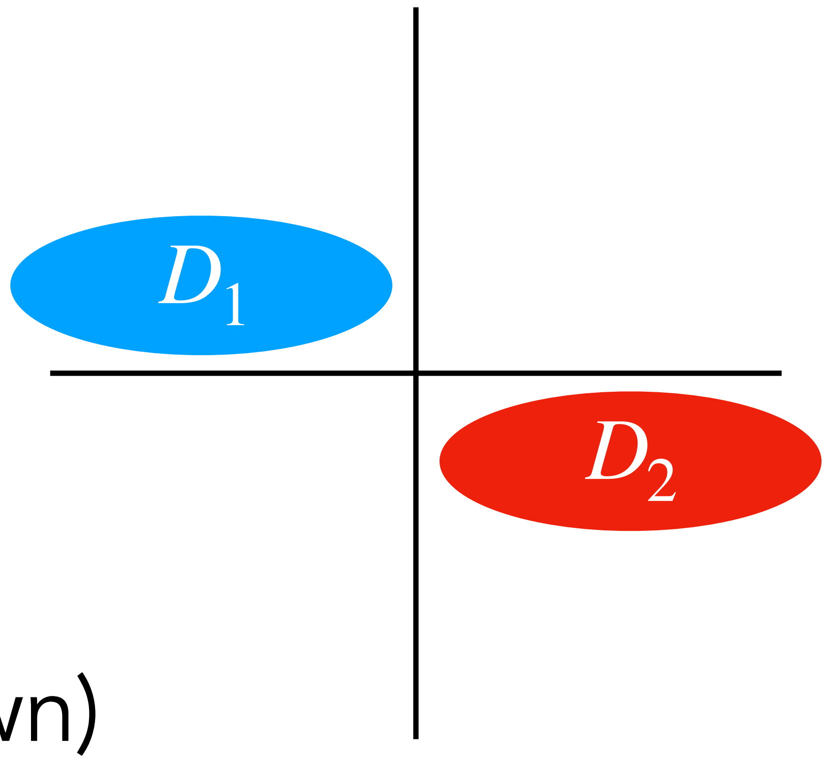
Max Likelihood Gaussian Classification

Distribution:

$y \sim \{-1, +1\}$

$x \sim \mathcal{N}(y \cdot \mu_*, \Sigma_*)$

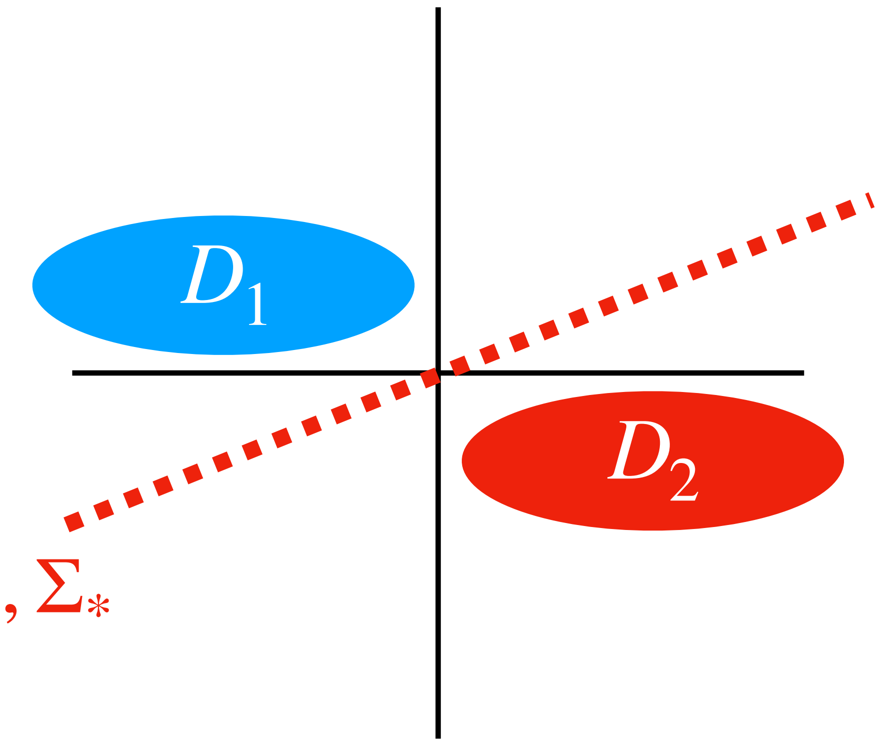
(Infinite sample regime = (μ_*, Σ_*) known)



Goal: Given a new sample **x**, estimate the most likely **y**

A Simple Theoretical Setting:
Max Likelihood Gaussian Classification

Standard approach:



→ Find max likelihood parameters

$$\hat{\mu}, \hat{\Sigma} = \arg\min_{\mu,\Sigma} \mathbb{E}_y \left[\mathbb{E}_{x\sim\mathcal{N}(y\cdot\mu,\Sigma)} \left[\ell(x; y\cdot\mu, \Sigma) \right] \right] = \mu_*, \Sigma_*$$

→ Classify via likelihood test:

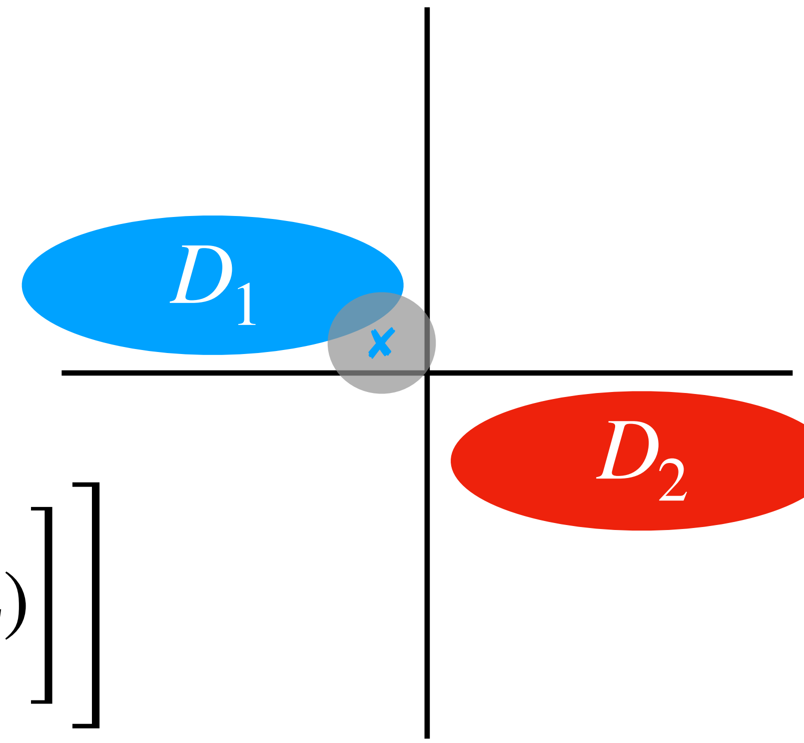
$$C(x) = \arg\max_y \ell(x; y\cdot\hat{\mu}, \hat{\Sigma}) = \textbf{sign}(x^\top \Sigma_*^{-1} \mu_*)$$

But: What if we want to do it in an ℓ_2-**robust** way?

A Simple Theoretical Setting:

Max Likelihood Gaussian Classification

ℓ_2-robust approach:



→ Find max likelihood parameters

$$\hat{\mu}_R, \hat{\Sigma}_R = \arg\min_{\mu,\Sigma} \mathbb{E}_y \left[\mathbb{E}_{x\sim\mathcal{N}(y\cdot\mu,\Sigma)} \left[\max_{\|\delta\|_2=\varepsilon} \ell(x+\delta;\mu,\Sigma) \right] \right]$$

→ Classify via likelihood test:

What is $\hat{\mu}_R$ and $\hat{\Sigma}_R$?

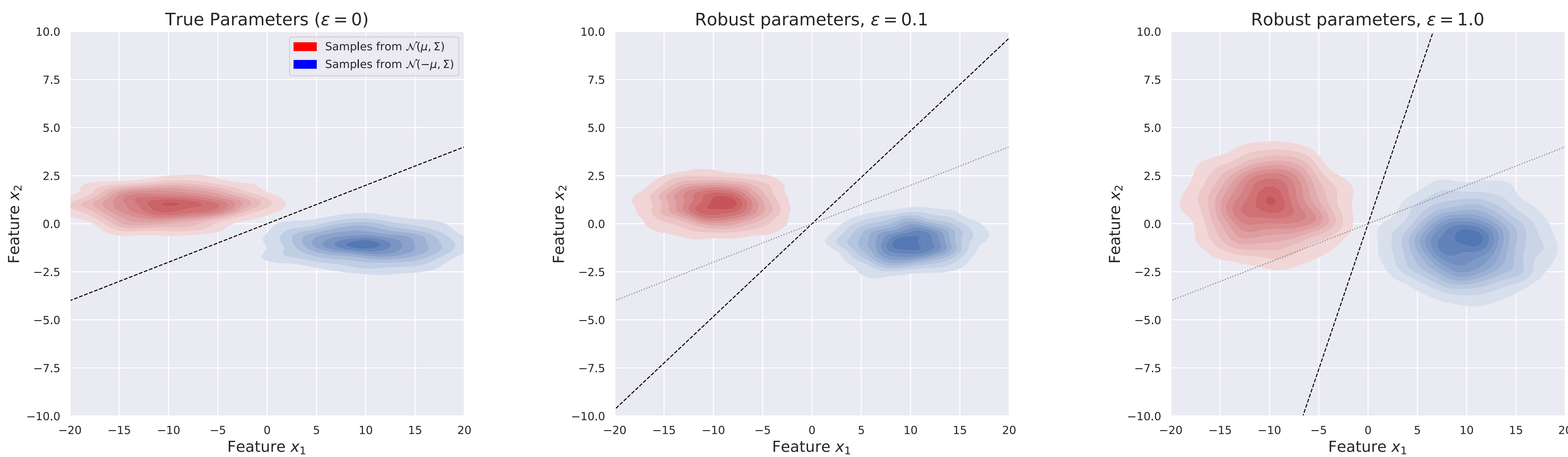
$$C(x) = \arg\max_y \ell(x; y\cdot\hat{\mu}_R, \hat{\Sigma}_R) = \textbf{sign}(x^\top \hat{\Sigma}_R^{-1}\hat{\mu}_R)$$

Note: If Σ_*^{-1} too far from **I**, adversary can move small distance wrt perturbation set, but large distance wrt (natural) features

A Simple Theoretical Setting:
Max Likelihood Gaussian Classification

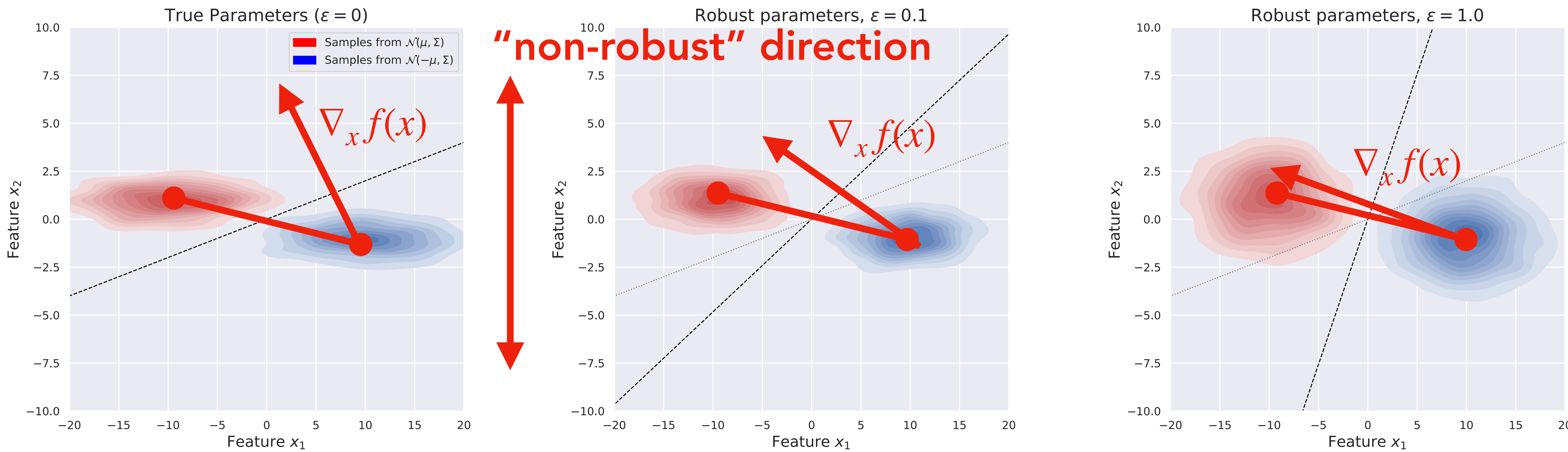
Theorem: We have that $\hat{\mu}_R = \mu_*$ and

$$\hat{\Sigma}_R = \frac{1}{2}\Sigma_* + \frac{1}{\lambda}\cdot\mathbf{I} + \sqrt{\frac{1}{\lambda}\cdot\Sigma_* + \frac{1}{4}\Sigma_*^2}$$ where $1/\lambda$ grows with ε.



Intuition: We "blend" Σ_* with **I** to "align" features wrt adversary

A Simple Theoretical Setting:
Max Likelihood Gaussian Classification



More 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

What now?

A new perspective on
adversarial robustness

(Provides insights into other questions too)

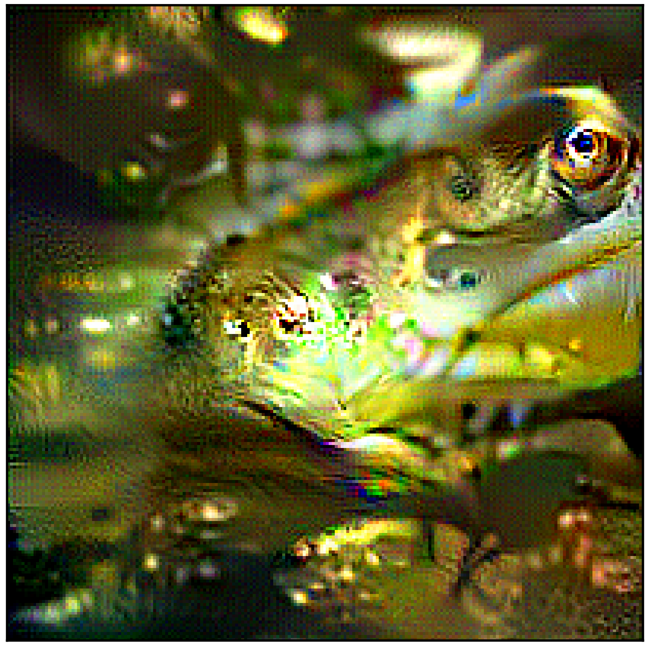
New capability: Robustification

Training set

frog

Restrict to features of robust model

New training set

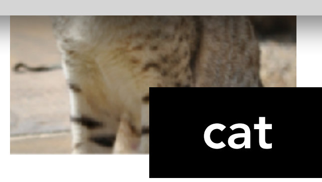


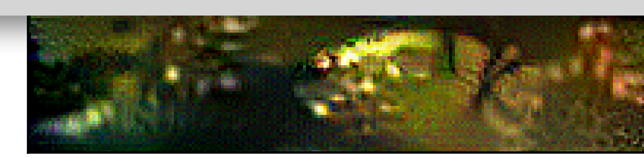
"robustified" frog

New capability: Robustification

(Original)

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

A Natural Consequence:
Transferability

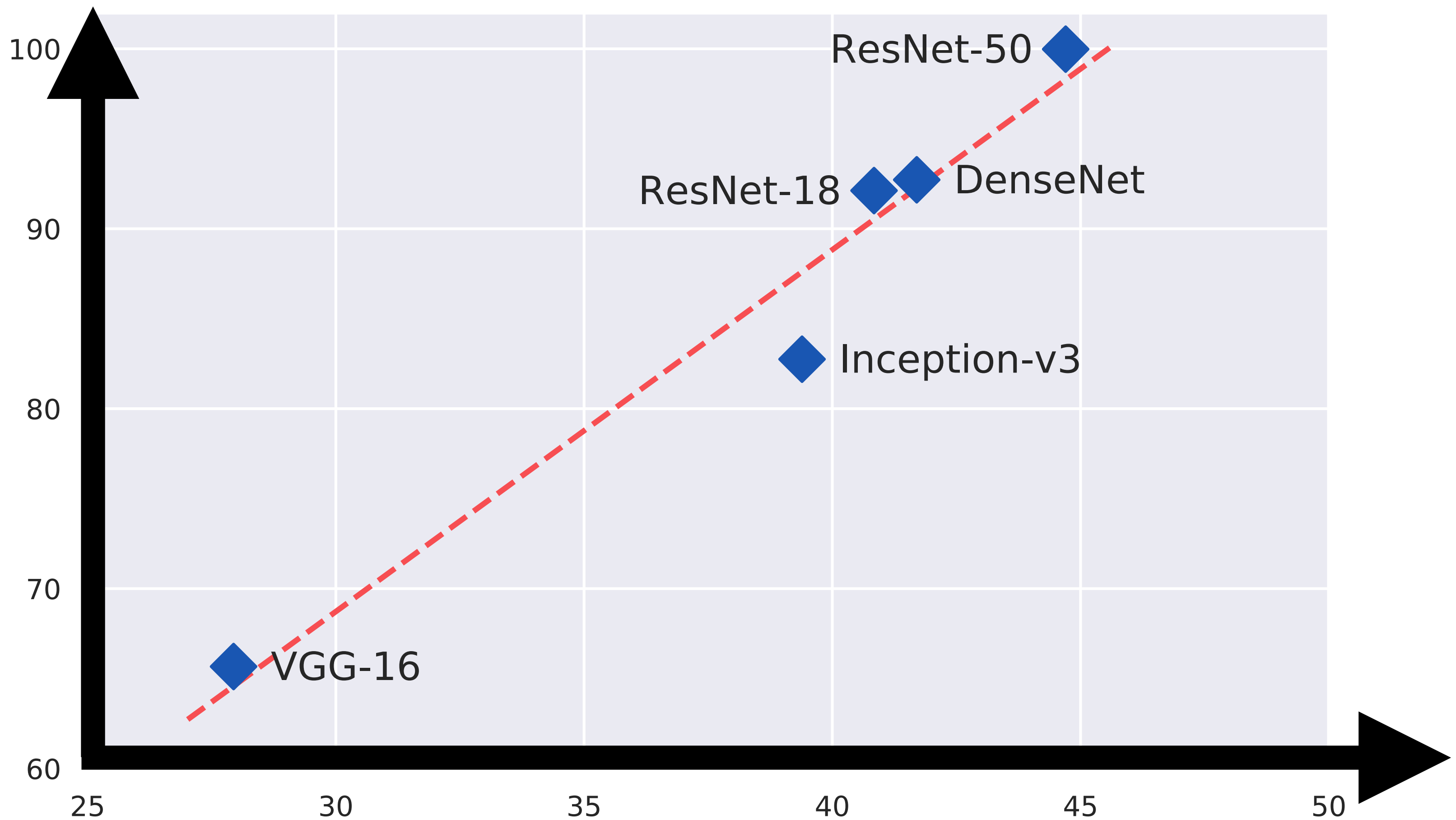
Adversarial perturbations = altering non-robust features

Features are a property of the **dataset**
(models just need to be able to capture them)

If non-robust features are useful, **many** models use them
→ **adversarial perturbations transfer**

A Natural Consequence:
Transferability

Adversarial Transferability (ResNet-50→X)



Test accuracy of X trained on non-robust features from ResNet-50

The Role of **Robust Training**

[Goodfellow Shlens Szegedy ’15] [M Makelov Schmidt Tsipras Vladu ’18]

Standard ERM $\min_{\theta} \mathbb{E}_{(x,y)\sim \widehat{D}}[\ell(\theta; x, y)]$

Robust ERM $\min_{\theta} \mathbb{E}_{(x,y)\sim \widehat{D}}[\max_{\delta \in \Delta} \ell(\theta; x + \delta, y)]$

→ Model can’t depend on anything that changes too much within Δ

Makes features that are non-robust w.r.t. Δ **useless**

New Take on **Randomized Smoothing**

[Cohen Rosenfeld Kolter ’19] [Lecuyer Atlidakis Geambasu Hsu Jana ’19]
[Salman Yang Li Zhang Zhang Razenshteyn Bubeck ’19]

Randomized Smoothing:
Train your model via **standard** ERM but on inputs with **large noise (from Δ)** added

→ Added noise **overwhelms** signal that is sensitive to perturbations in Δ

Makes features that are non-robust w.r.t. Δ **useless**

Robustness and Data Efficiency

Robust models can only leverage **robust** features

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

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

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

But: Is leveraging non-robust features even desirable?

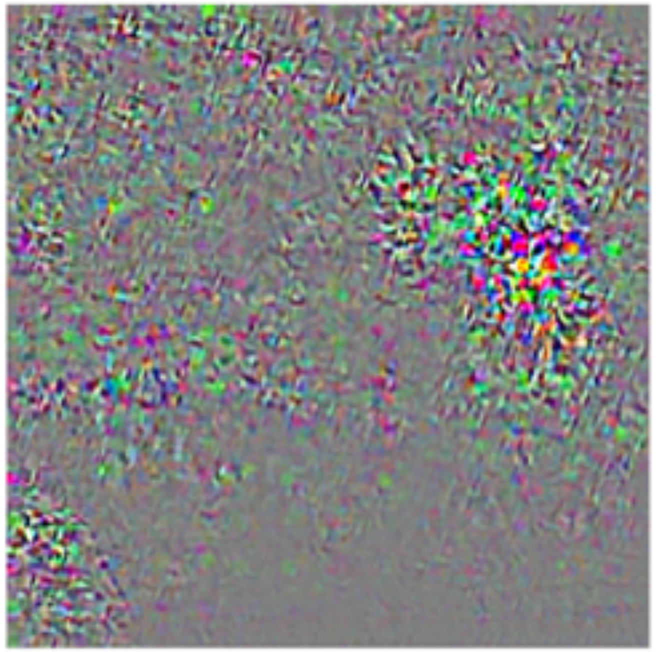
What if we **prevent** models from learning **non-robust** features?

[Tsipras Santurkar Engstrom Turner M ’18]

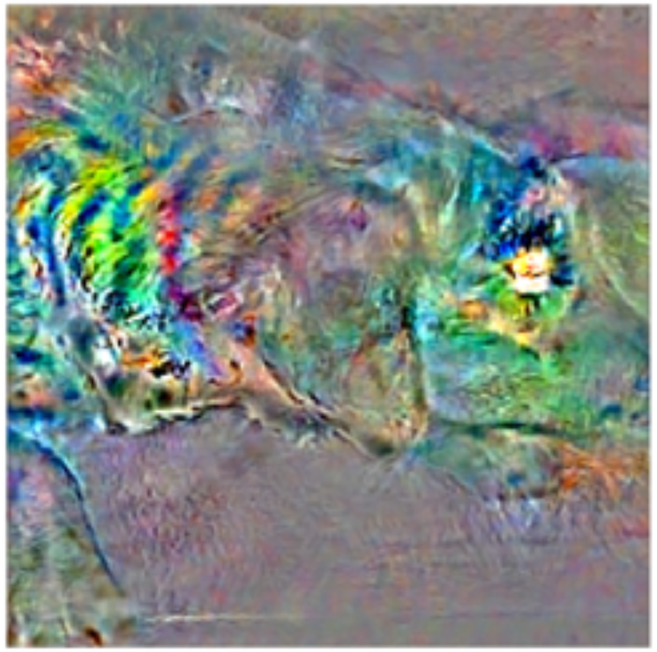
[Engstrom Ilyas Santurkar Tsipras Tran M ’19]

Robustness → Perception Alignment

Input



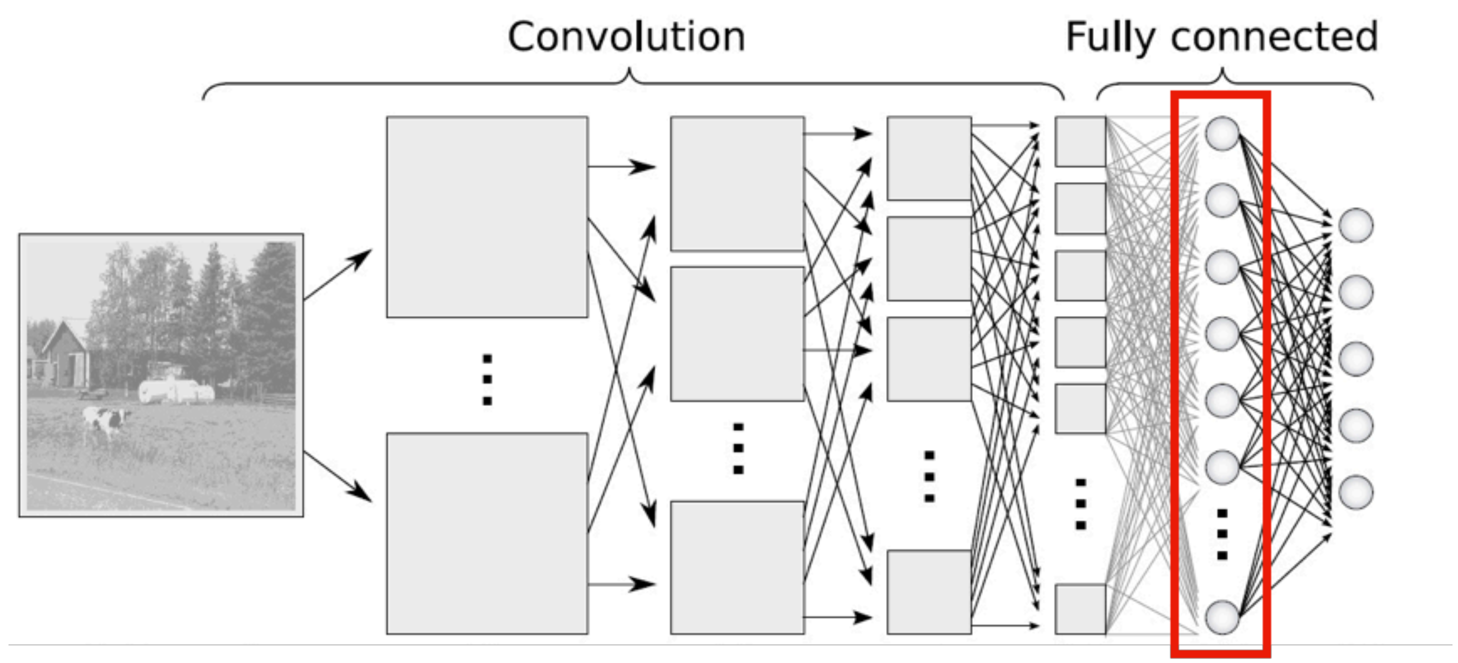
Gradient of
standard model

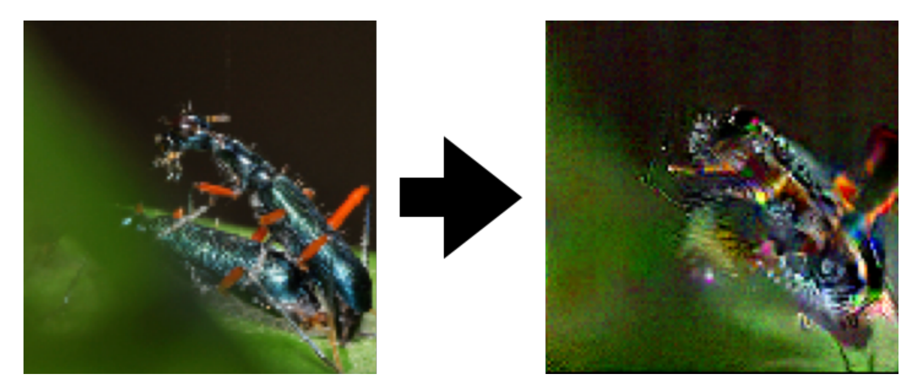


Gradient of
adv. robust model

→ 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]

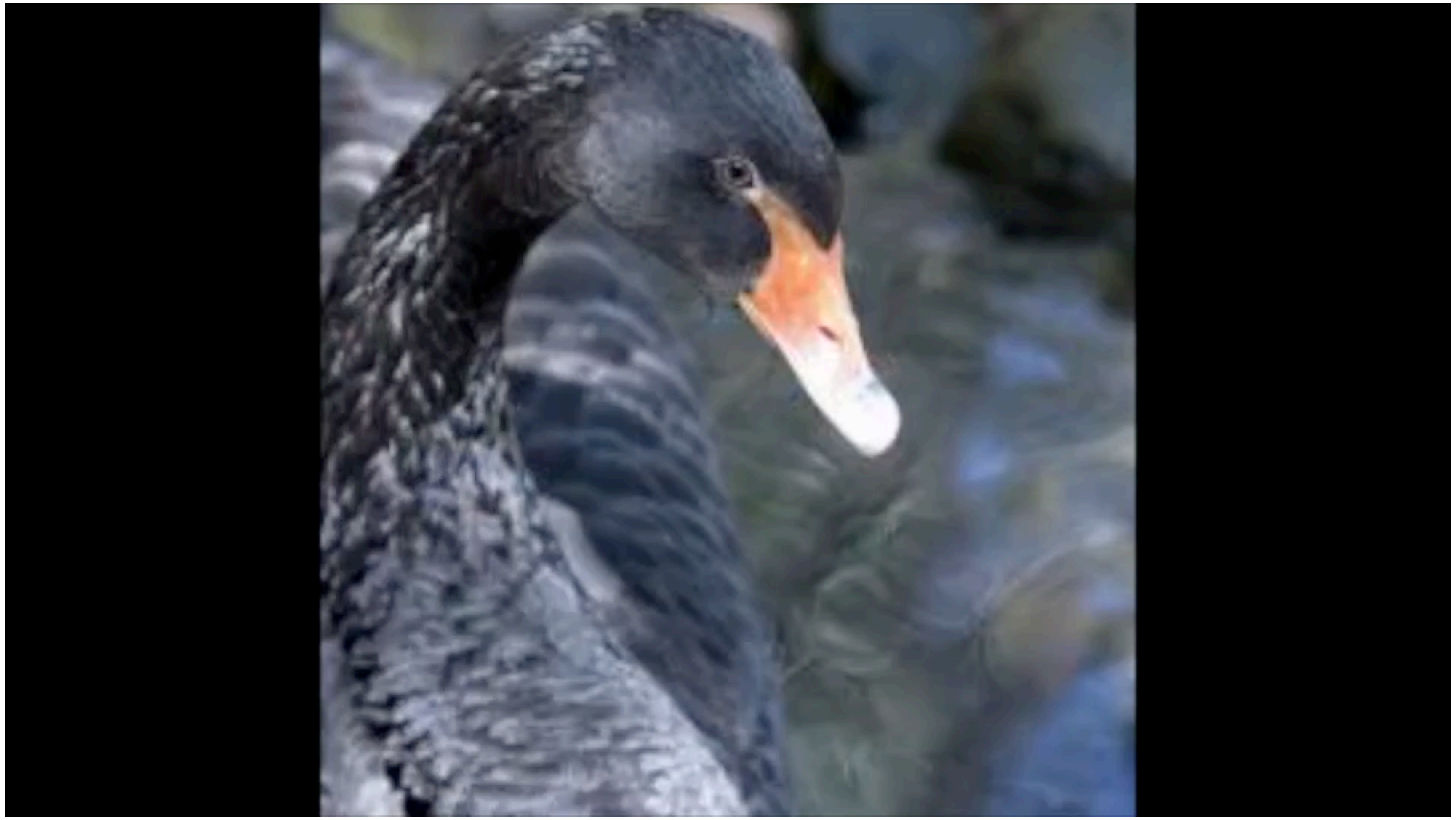
But here:

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

→ Everything boils down to simple optimization primitives

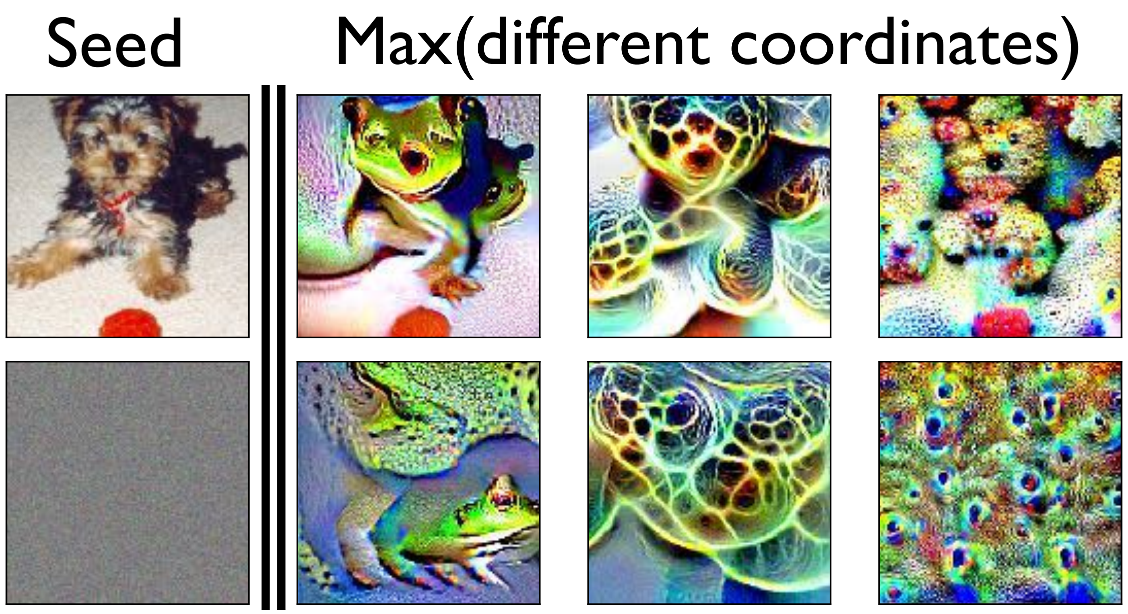
→ No priors, no regularization, no post-processing
(and thus we are fully faithful to the model)

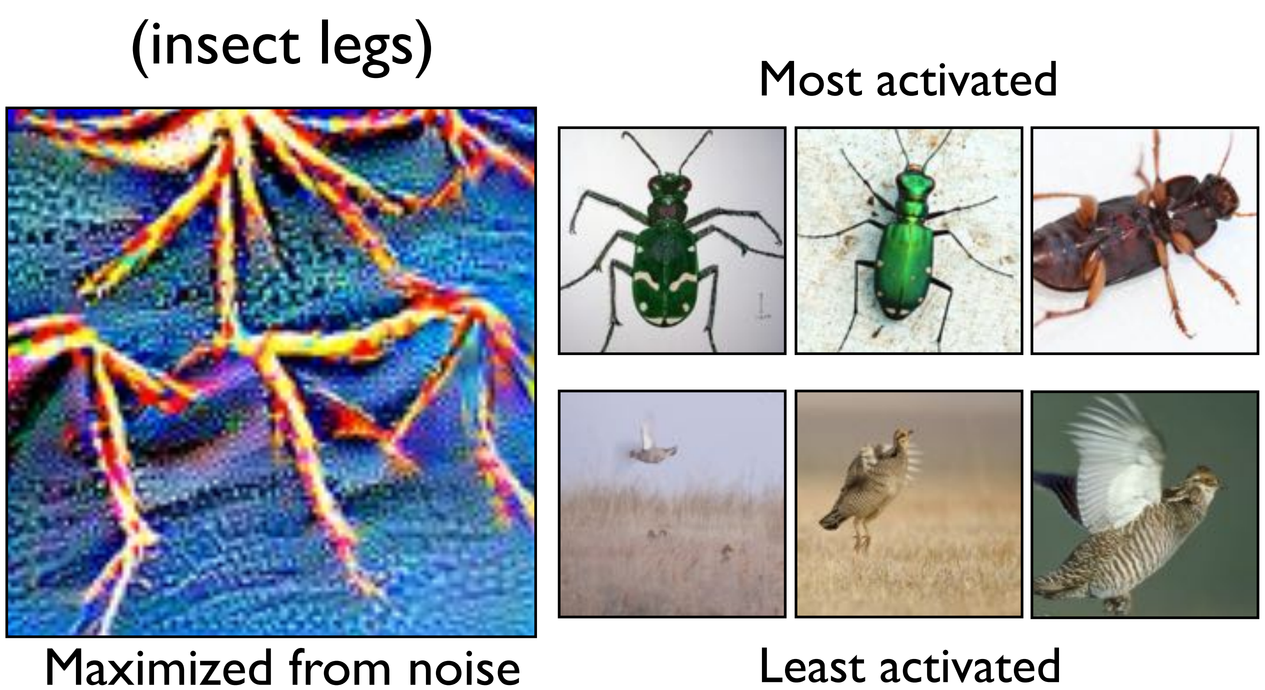
Robustness → Better Representations



Interpolation between **any** two inputs

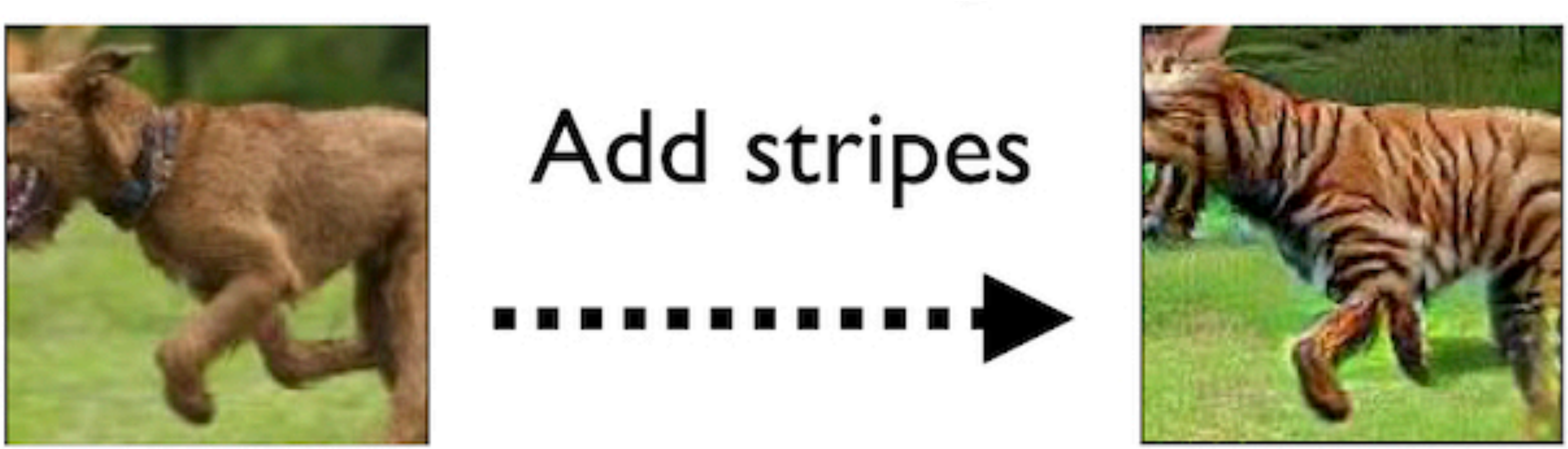
Robustness → Better Representations





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]

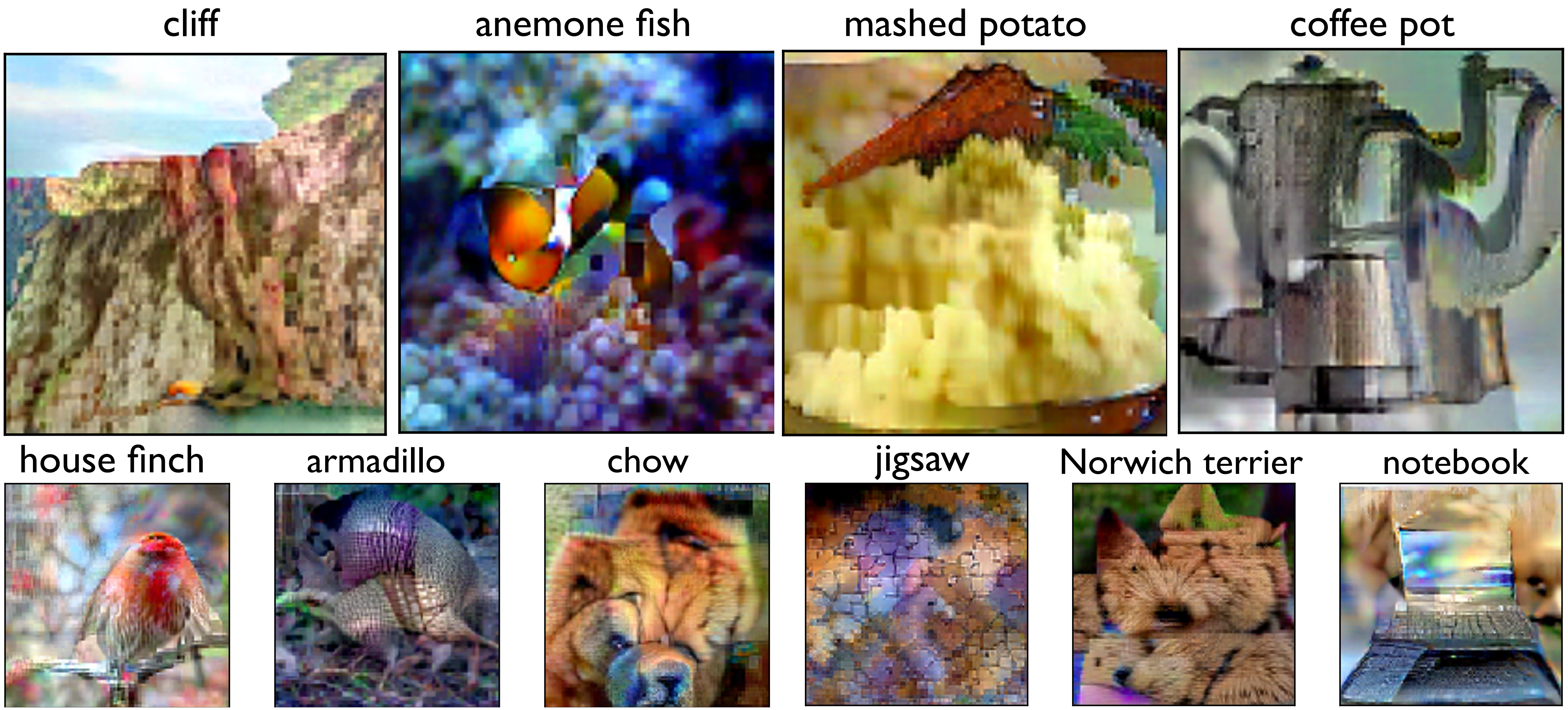
Robustness → CV Applications

A **single robust classifier** suffices to perform a wide range of computer vision (image synthesis) tasks

In fact: (Again) the simplest possible approach is enough

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

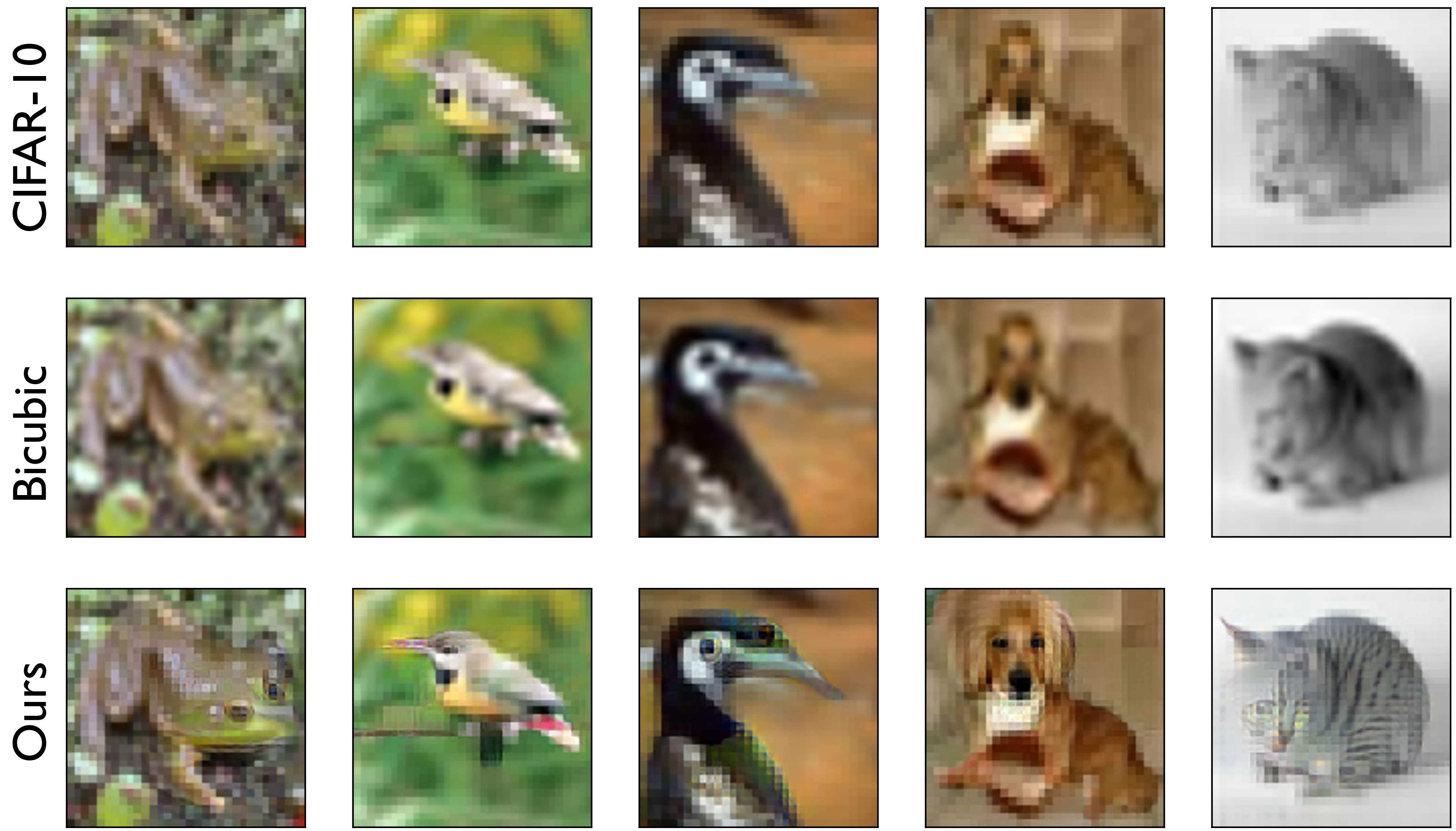
Robustness → CV Applications



(Random samples, 1K training images, no tuning)

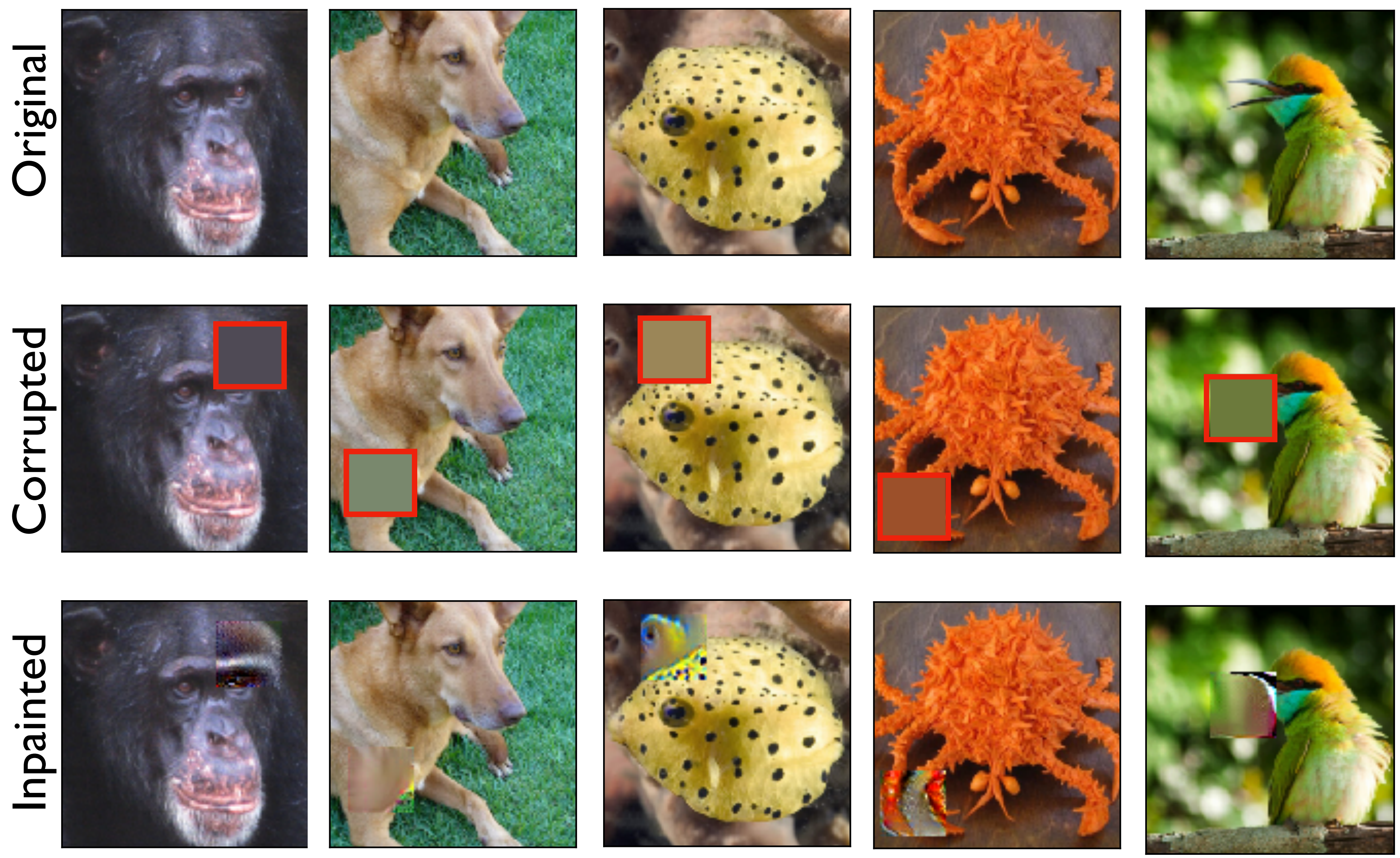
Generative models (that work **better** on **large** datasets)

Robustness → CV Applications



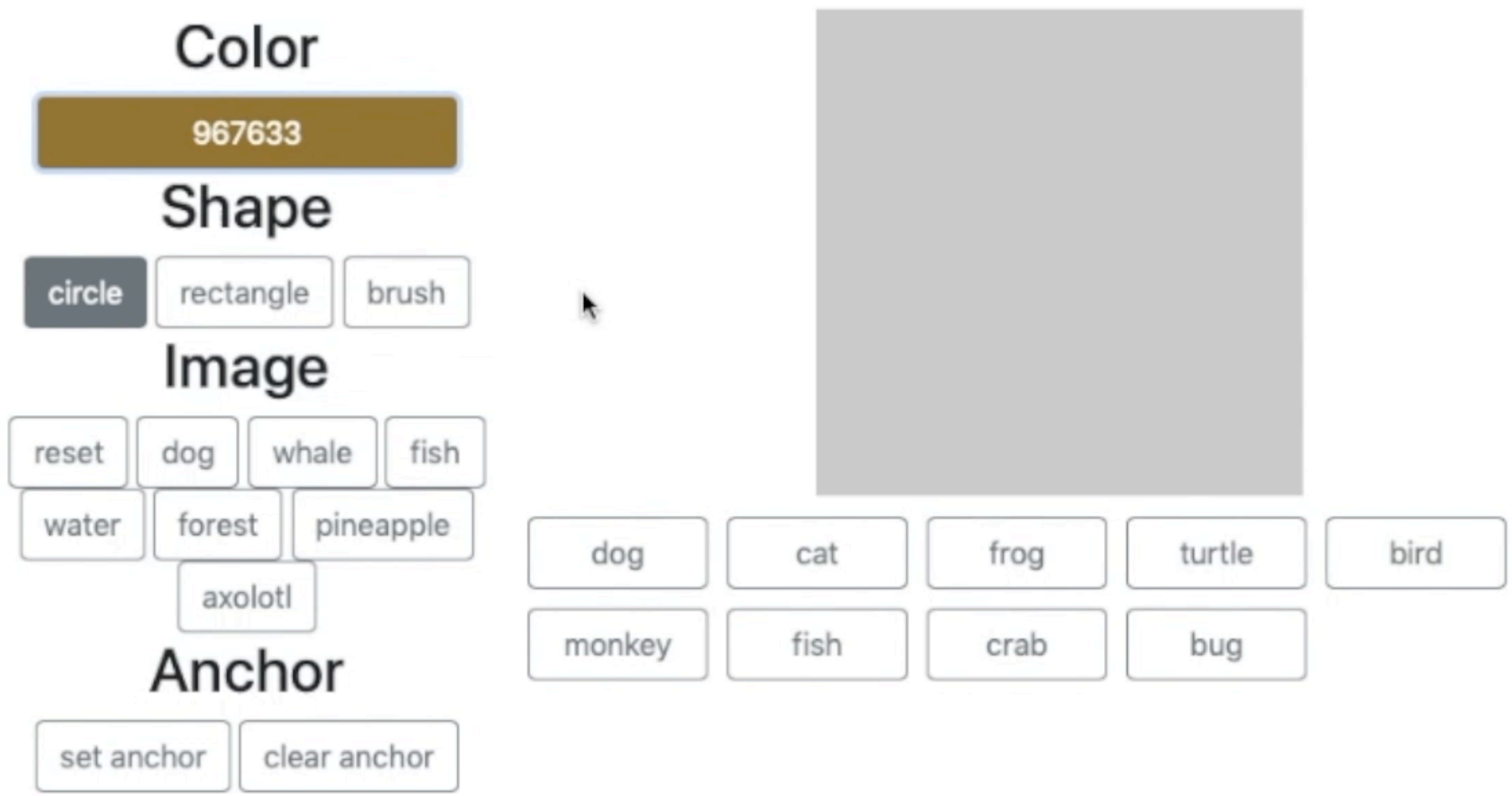
Super-Resolution

Robustness → CV Applications



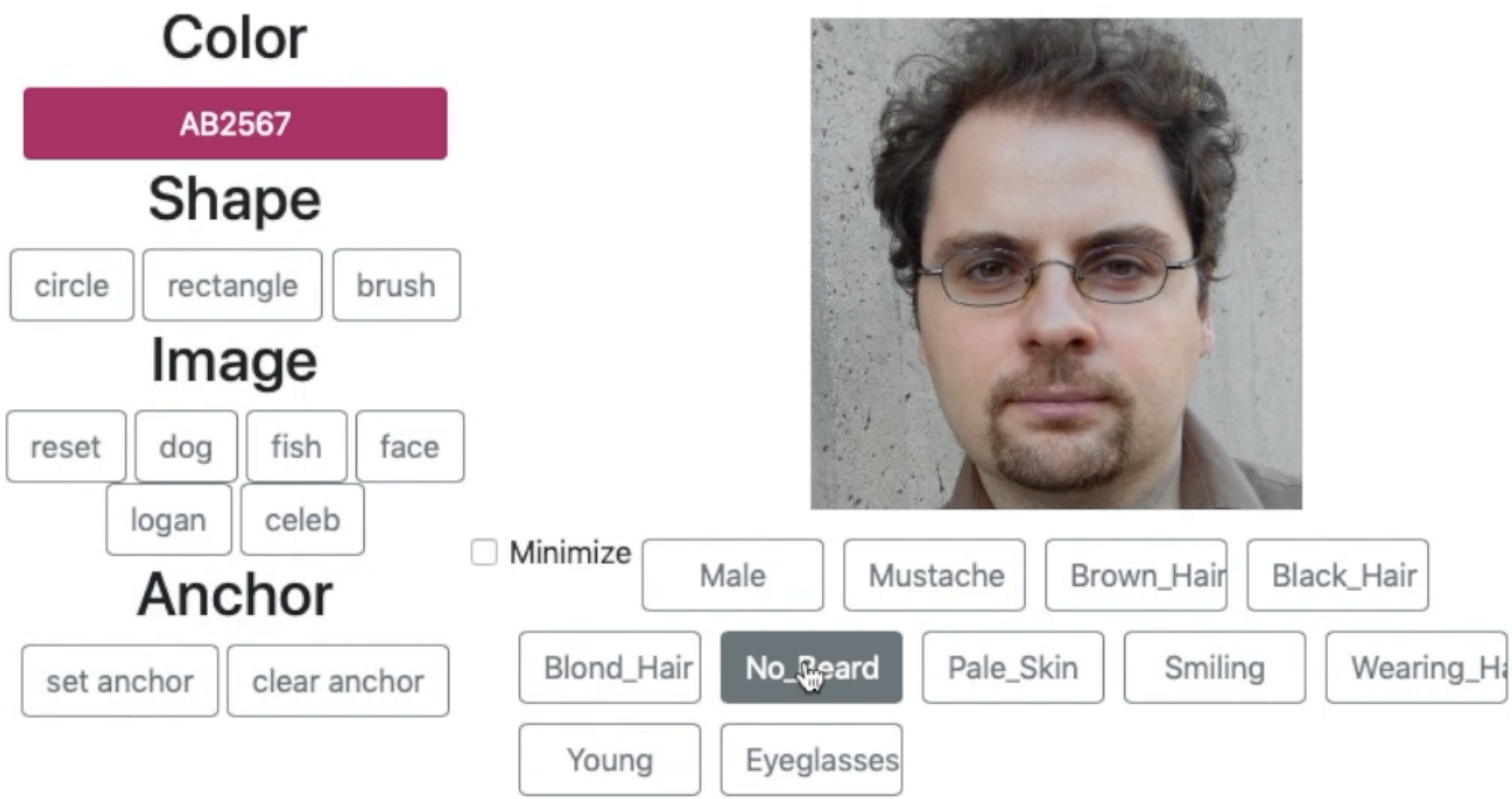
In-Painting

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"