Adversarial Training and Robustness for Multiple Perturbations

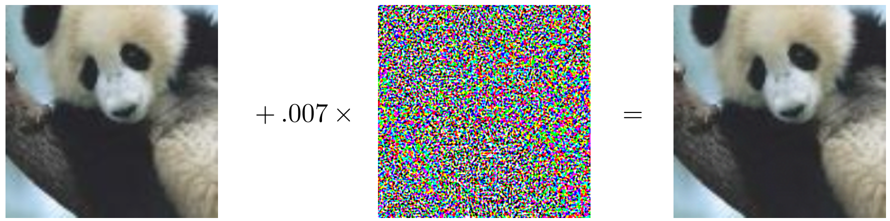
Florian Tramèr

Stanford security lunch

May 22nd 2019

Joint work with Dan Boneh

Adversarial examples: what (we think) we know



(Szegedy et al. 2013, Goodfellow et al. 2015)

Pretty sure this is a panda

I'm certain this is a gibbon

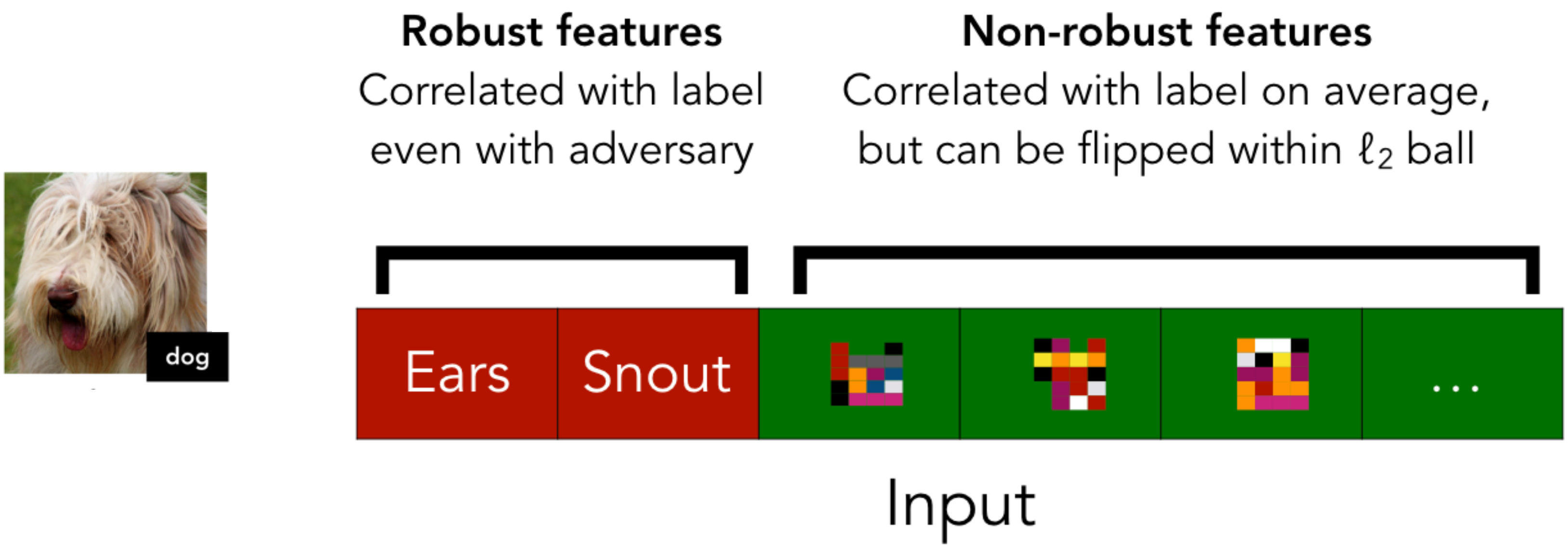
- Affects all ML models & domains
  (images, speech, text, etc.)
- Perturbations transfer between models
  (mostly on images)
- Explanations:
  - Local linearity of models (Goodfellow et al. 2015)
  - High dimensionality of data (Fawzi et al. 2018, Gilmer et al. 2018)
  - Superficial features (Jo & Bengio 2017, Jetley et al. 2018, Ilyas et al. 2019)

Adversarial examples as superficial features

Thesis: Data contains *imperceptible, yet generalizable* features

$\Rightarrow$ *A model trained with ERM will use these features to get better accuracy*

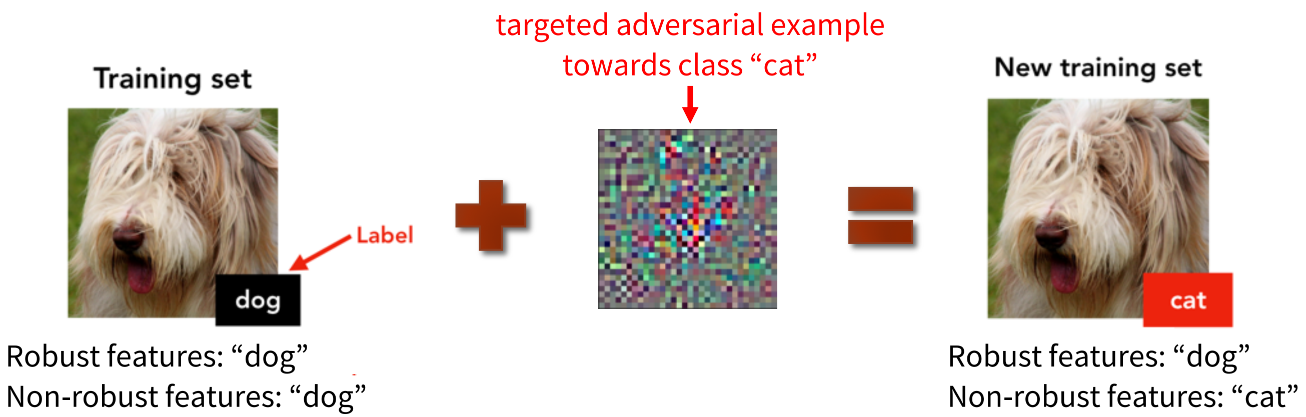
$\Rightarrow$ *Adversarial examples manipulate these features*



(Ilyas et al. 2019)

Adversarial examples as superficial features

Experiment:



New training set: all dogs mislabeled as "cat", all cats mislabeled as "dog"

What could a model trained on this new dataset learn?

1) Robust features of a dog means "cat"
2) Non-robust features of a cat means "cat"

⇒ **A model trained on the new training set has high accuracy on the original unperturbed and correctly labeled test set!**

⇒ Conclusion: the model learned to associate each class with imperceptible yet generalizable features, which correspond to adversarial examples

(Ilyas et al. 2019)

Adversarial training

How do we "force" a model to ignore non-robust features?

$\Rightarrow$ Train the model to be invariant to changes in these features

$\Rightarrow$ For each training input (**x**, y), find worst-case adversarial input

$$\underset{\boldsymbol{x}' \in S(\boldsymbol{x})}{\operatorname{argmax}} \ \mathrm{Loss}(f(\boldsymbol{x}'), y)$$

A set of allowable perturbations of **x**
e.g., $\{\mathbf{x}' : \| \mathbf{x} - \mathbf{x}' \|_\infty \le \varepsilon\}$

(e.g., using Projected Gradient Descent on the model loss)

$\Rightarrow$ Train the model on (**x**', y)

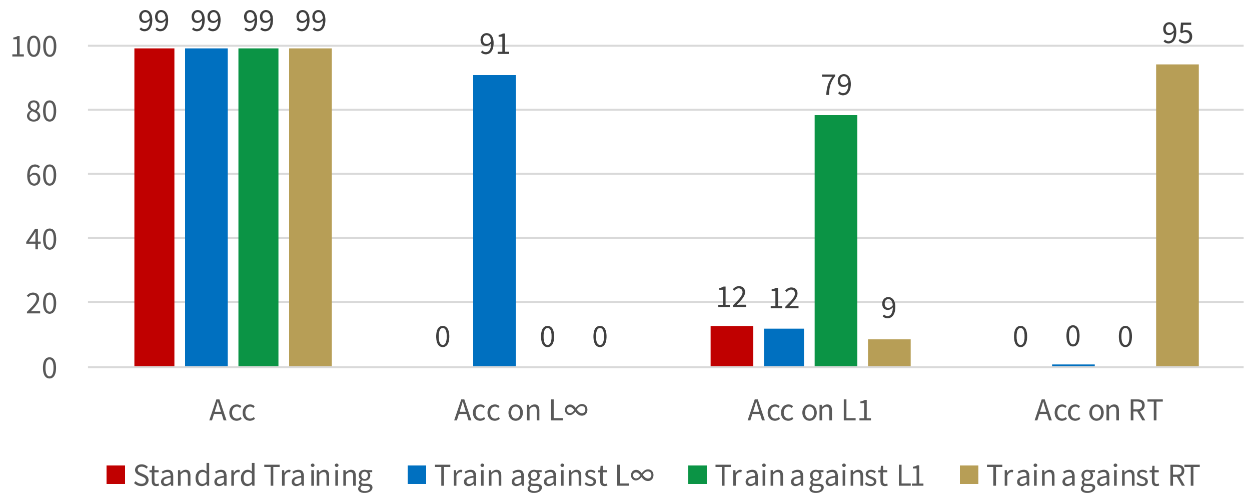
Worst-case data augmentation by manipulating non-robust features

Multi-perturbation robustness

The “robustness” of a feature depends on the considered perturbation set S(**x**)

- What we want: S(**x**) = “all perturbations that don’t affect class semantics”
- What we have: S(**x**) = “a small L_p ball around **x**” or
  S(**x**) = “small rotations & translations of **x**”

MNIST:



Robustness to one perturbation type ≠ robustness to all
Robustness to one type can increase vulnerability to others

The multi-perturbation robustness trade-off

If there exist models with high robust accuracy for perturbation sets $S_1, S_2, \ldots, Sn$, does there **exist** a model robust to perturbations from $\cup_{i=1}^{n} S_i$?

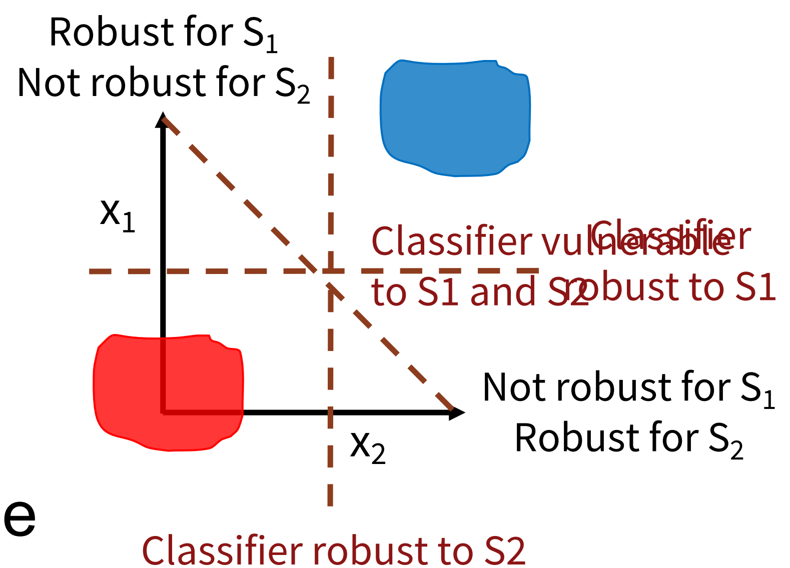
<u>Answer:</u> in general, NO!

There exist "mutually exclusive perturbations" (MEPs)

(robustness to S_1 implies vulnerability to S_2 and vice-versa)

Formally, we show that for a simple Gaussian binary classification task:

- L_1 and L_∞ perturbations are MEPs
- L_∞ and spatial perturbations are MEPs



Experiments on real data

Can we train models to be robust to multiple perturbation types simultaneously?

Adversarial training for multiple perturbations:

⇒ For each training input (**x**, y), find worst-case adversarial input

$$\underset{\boldsymbol{x}' \in \bigcup_{i=1}^{n} S_i}{\operatorname{argmax}} \ \operatorname{Loss}(f(\boldsymbol{x}'), y)$$

⇒ "Black-box" approach:

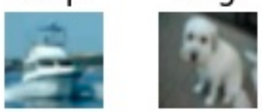
Scales linearly in number of perturbation sets

$$\underset{\boldsymbol{x}' \in \bigcup_{i=1}^{n} S_i}{\operatorname{argmax}} \ \operatorname{Loss}(f(\boldsymbol{x}'), y) = \underset{1 \le i \le n}{\operatorname{argmax}} \left\{ \underbrace{\underset{\boldsymbol{x}' \in S_i}{\operatorname{argmax}} \ \operatorname{Loss}(f(\boldsymbol{x}'), y)}_{\text{Use existing attack tailored to } S_i} \right\}$$

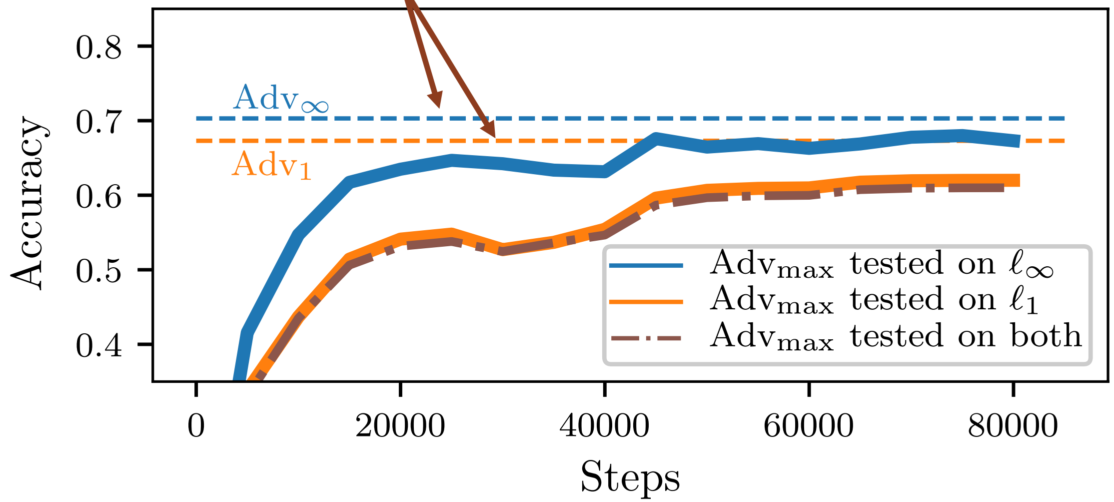
Results

CIFAR10:

ship dog

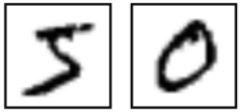


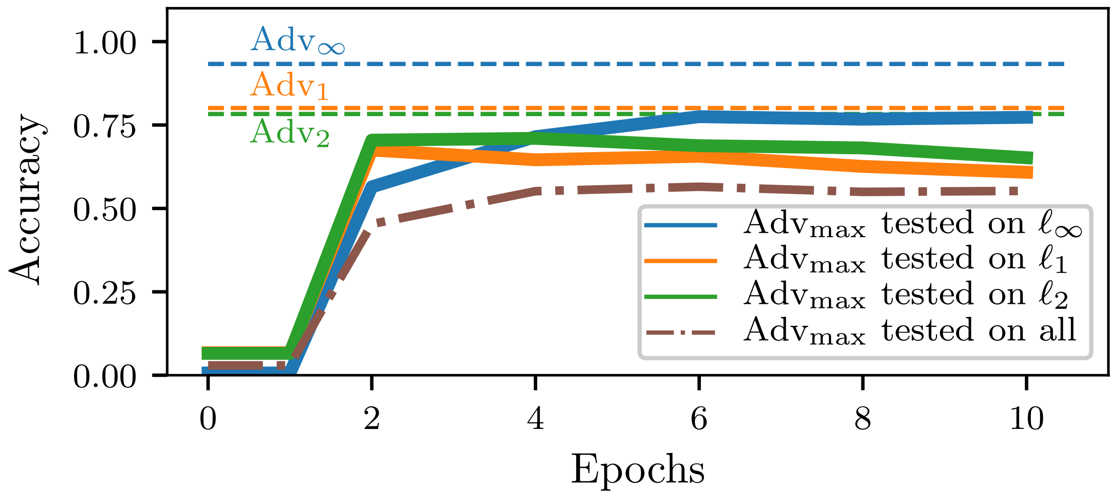
Robust accuracy when training/evaluating on a single perturbation type



Loss of ~**5%** accuracy

MNIST:

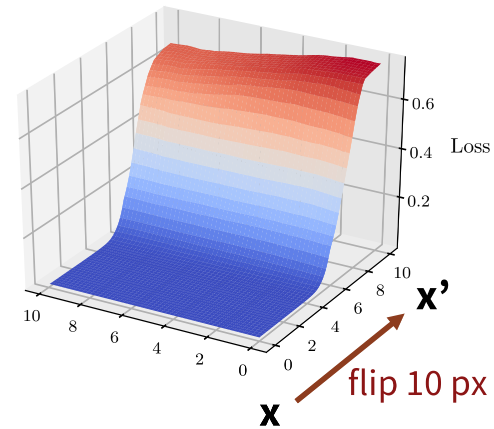




Loss of ~**20%** accuracy

MNIST and gradient masking

 $\in \{0,1\}^{784}$



How to get robustness against L_∞ noise?

$\Rightarrow$ Threshold the input, e.g., $f(\mathbf{x}) \approx f'(\text{sign}(\mathbf{x}))$

$\Rightarrow$ <u>Problem:</u> $\nabla_x f = \mathbf{0}$ so gradient-based L_1 and L_2 attacks also fail

When we train against gradient-based L_1 or L_2 attacks, the model does not learn to do thresholding!

$\Rightarrow$ This would be a valid minimizer of the training objective

$\Rightarrow$ The model is actually robust to L_1 or L_2 noise without gradient masking

When we train against L_∞, L_1 and L_2 attacks simultaneously, the model uses thresholding again...

$\Rightarrow$ The model is not robust to gradient-free L_1 or L_2 attacks

$\Rightarrow$ Open problem: how to get rid of gradient masking in an efficient way

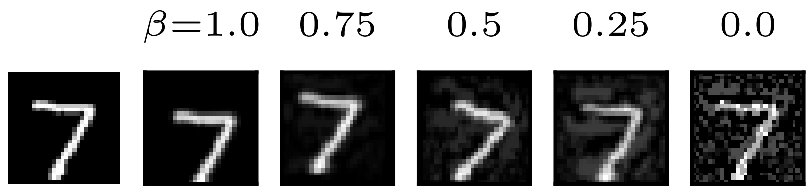
Affine adversaries

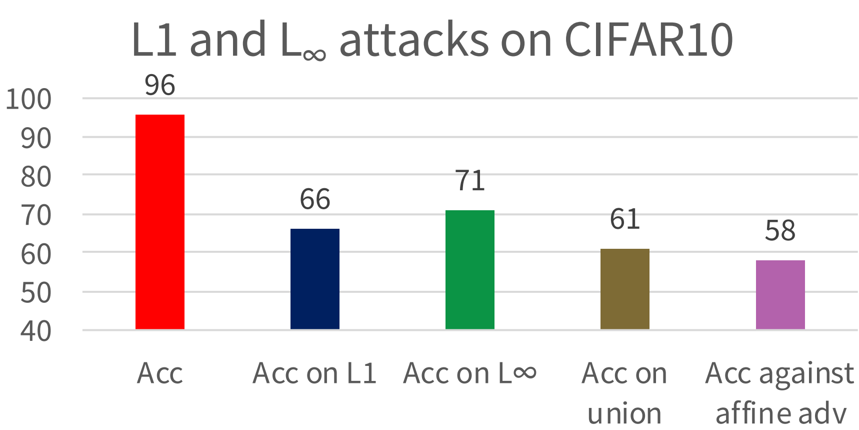
Instead of picking perturbations from $S_1 \cup S_2$ why not combine them?

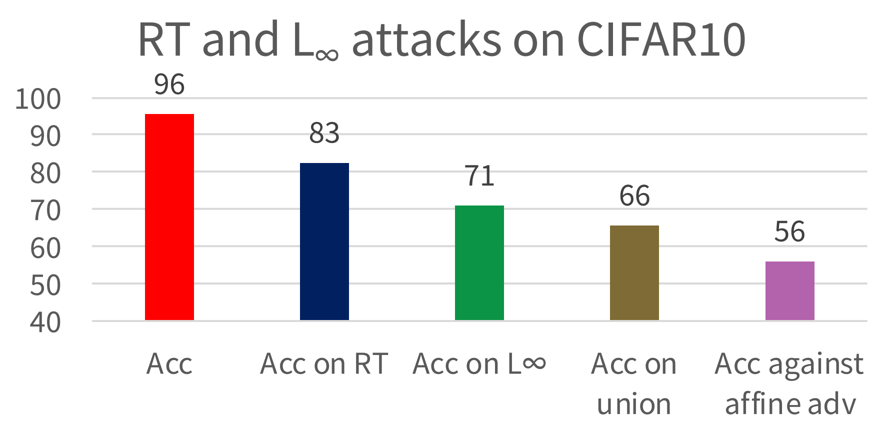
E.g., small L_1 noise + small L_∞ noise

or small rotation/translation + small L_∞ noise

Affine adversary picks perturbation from $\beta S_1 + (1-\beta)S_2$, for $\beta \in [0,1]$







Open problems

How do we get models to ignore non-robust features?

How do we express which features are robust / non-robust to humans in the first place?

- I.e., how do we “define” non-robust features?
- Currently, simple proxies: L_p norms, rotations, etc.
  These are neither sufficient nor <u>necessary</u>! (upcoming slide)

How do we **efficiently** get models to ignore multiple types of non-robust features

- Our current approach: train on worst-case example from union of perturbation sets ⇒ scales linearly in number of perturbation types
- Can we get something sublinear?

More problems with L_p perturbations

Let's look at MNIST again:

(Simple dataset, centered and scaled, non-trivial robustness is achievable)

 $\in \{0, 1\}^{784}$

Using adversarial training, models have been trained to "extreme" levels of robustness

(E.g., robust to L_1 noise > 30 or L_∞ noise > 0.3)

natural

L_1 perturbed

L_∞ perturbed

For such examples, humans agree more often with an undefended model than with an overly robust model

Jacobsen **et al.** "Exploiting Excessive Invariance caused by Norm-Bounded Adversarial Robustness"