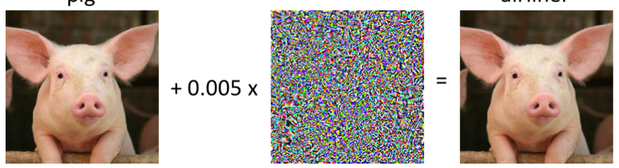
Adversarial Examples Are Not Bugs, They Are Features

Andrew Ilyas*, Shibani Santurkar*, Dimitris Tsipras*, Logan Engstrom*, Brandon Tran and Aleksander Madry

Massachusetts Institute of Technology

madry-lab.ml

Adversarial Examples: A Challenge for ML Systems



Why are ML models **so sensitive** to small perturbations?

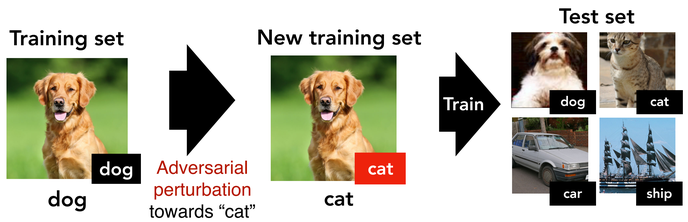
Useless features the model is unreasonably **sensitive** to

Useful features that are responsible for good classification

Adversary only changes these features to create an adversarial example

Prevailing theme: They stem from bugs/aberrations

A Simple Experiment

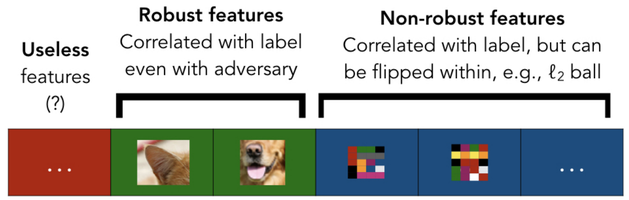


So: We train on a **totally "mislabeled"** dataset but expect performance on a **"correct"** dataset

Training Data	Dataset	
	CIFAR-10	ImageNet$_R$
Standard Dataset	95.3%	96.6%
"Mislabeled" Dataset	43.7%	64.4%

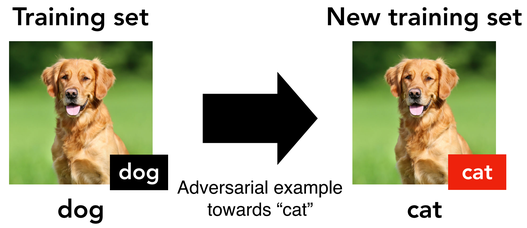
Result: nontrivial accuracy on the **original** task

The Robust Features Model



From "max accuracy" view: All features are good
If NRFs are (often) good: **Models want to use them**

Thus: Models use NRFs → adversarial examples



RFs misleading but **NRFs suffice for generalization**

Directly Manipulating Features

"Robust" Data: *Standard* training → *robust* models

Robust Optimization: Makes NRFs useless for learning

→ Need more data to learn from only RFs (cf. [Schmidt et al., 2018])

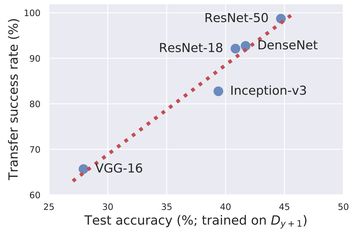
→ Trade-off between robustness/accuracy (cf. [Tsipras et al., 2019])

Implications

ML models do not work the way we expect them to

Adversarial examples: A "human-based" phenomenon?

Transfer Attacks: Models rely on similar NRFs



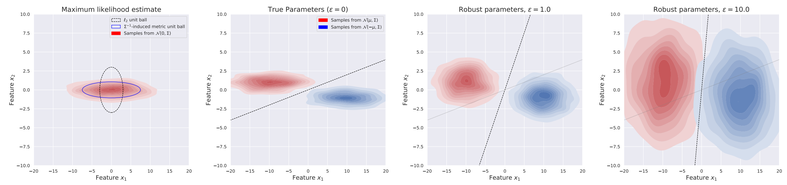
Interpretability: May need to be enforced at training time

A Theoretical Framework

→ We consider (robust) MLE classification between Gaussians

→ Vulnerability is misalignment between the data geometry and adversary's (ℓ_2) geometry

→ Shows that robust optimization *better aligns* these geometries



Moving Forward

→ Do we want our models to rely on NRFs?

→ How should we think of interpretability?

Robustness as a goal **beyond security/reliability?**

Paper	Blog	Python Library
arxiv:1905.02175	gradsci.org/adv	MadryLab/robustness