Limitations of Threat Modeling in Adversarial Machine Learning

Florian Tramèr
EPFL, December 19th 2019

Based on joint work with Jens Behrmannn, Dan Boneh, Nicholas Carlini, Pascal Dupré, Jörn-Henrik Jacobsen, Nicolas Papernot, Giancarlo Pellegrino, Gili Rusak

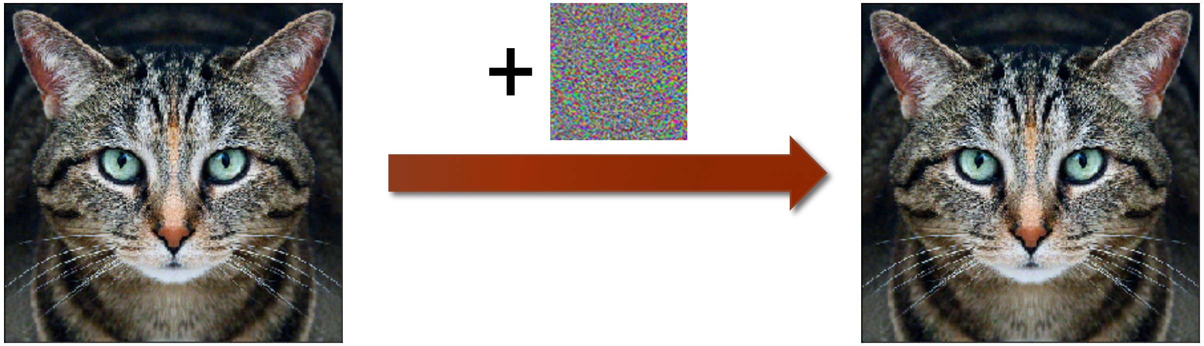
The state of adversarial machine learning

GANs vs Adversarial Examples

Maybe we need to write 10x more papers

Inspired by N. Carlini, “Recent Advances in Adversarial Machine Learning”, ScAINet 2019

Adversarial examples



88% Tabby Cat

99% Guacamole

Biggio et al., 2014
Szegedy et al., 2014
Goodfellow et al., 2015
Athalye, 2017

How?

- Training $\Longrightarrow$ "tweak model parameters such that $f($  $) = cat$"
- Attacking $\Longrightarrow$ "tweak input pixels such that $f($  $) = guacamole$"

The bleak state of adversarial examples

Elon Musk
@elonmusk

Never trust cynics, as they excuse their own bad deeds by telling themselves everyone does it

10:41 PM · Dec 18, 2019 · Twitter for iPhone

12.5K Retweets **91.1K** Likes

The bleak state of adversarial examples

- Most papers study a “toy” problem
  Solving it is not useful per se, but maybe we’ll find new insights or techniques
- Going beyond this toy problem (even slightly) is hard
- Overfitting to the toy problem happens and is harmful
- The “non-toy” version of the problem is not actually that relevant for computer security
  (except for ad-blocking)

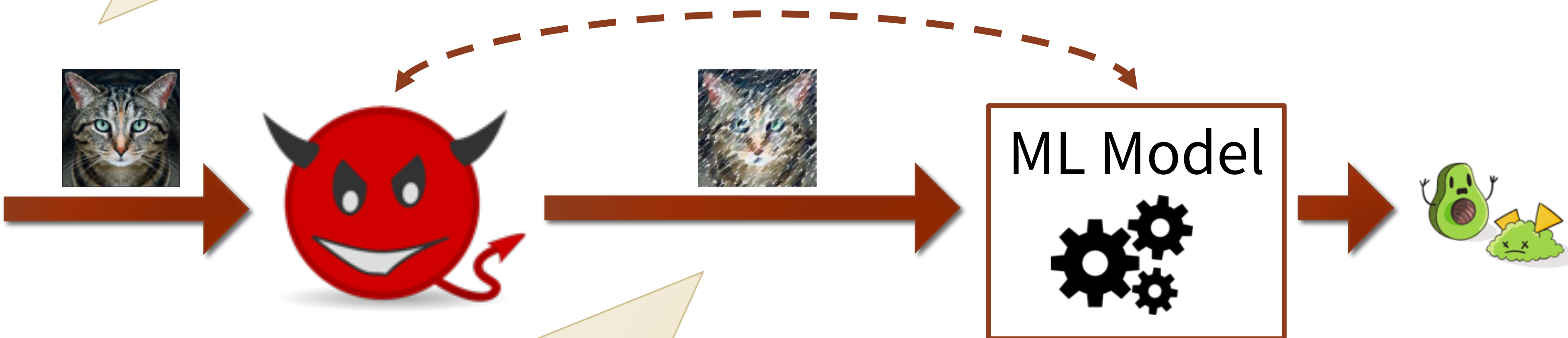
The bleak state of adversarial examples

- Most papers study a “toy” problem
  Solving it is not useful per se, but maybe we’ll find new insights or techniques
- Going beyond this toy problem (even slightly) is hard
- Overfitting to the toy problem happens and is harmful
- The “non-toy” version of the problem is not actually that relevant for computer security
  (except for ad-blocking)

The standard game [Gilmer et al. 2018]

Adversary is given an input **x** from a data distribution

Adversary has some info on model (white-box, queries, data)



Adversary produces adversarial example **x’**

Adversary wins if **x’** ≈ **x** and defender misclassifies

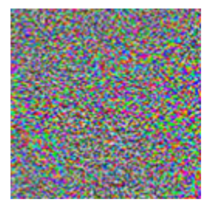
Relaxing and formalizing the game

How do we define **x'** ≈ **x** ?

- "Semantics" preserving, fully imperceptible?

Conservative approximations [Goodfellow et al. 2015]

- Consider noise that is clearly semantics-preserving

E.g.,  =  +  where $\|\boldsymbol{\delta}\|_\infty = \max \delta_i \le \epsilon$

x' **x** **δ**

- Robustness to this noise is *necessary* but not *sufficient*
- **Even this "toy" version of the game is hard, so let's focus on this first**

Progress on the toy game

- **Many** broken defenses [Carlini & Wagner 2017, Athalye et al. 2018]

- Adversarial Training [Szegedy et al., 2014, Madry et al., 2018]

$\Rightarrow$ For each training input ($\mathbf{x}$, y), train on worst-case adversarial input

$$\underset{\|\boldsymbol{\delta}\|_\infty \leq \epsilon}{\operatorname{argmax}} \text{Loss}(f(\boldsymbol{x} + \boldsymbol{\delta}), y)$$

- Certified Defenses

[Hein & Andriushchenko 2017, Raghunathan et al., 2018, Wong & Kolter 2018]

Progress on the toy game

Robustness to noise of small l_p norm is a "toy" problem

Solving this problem is not useful per se, unless it teaches us new insights

Solving this problem does not give us "secure ML"

Outline

- Most papers study a “toy” problem
  Solving it is not useful per se, but maybe we’ll find new insights or techniques

- Going beyond this toy problem (even slightly) is hard

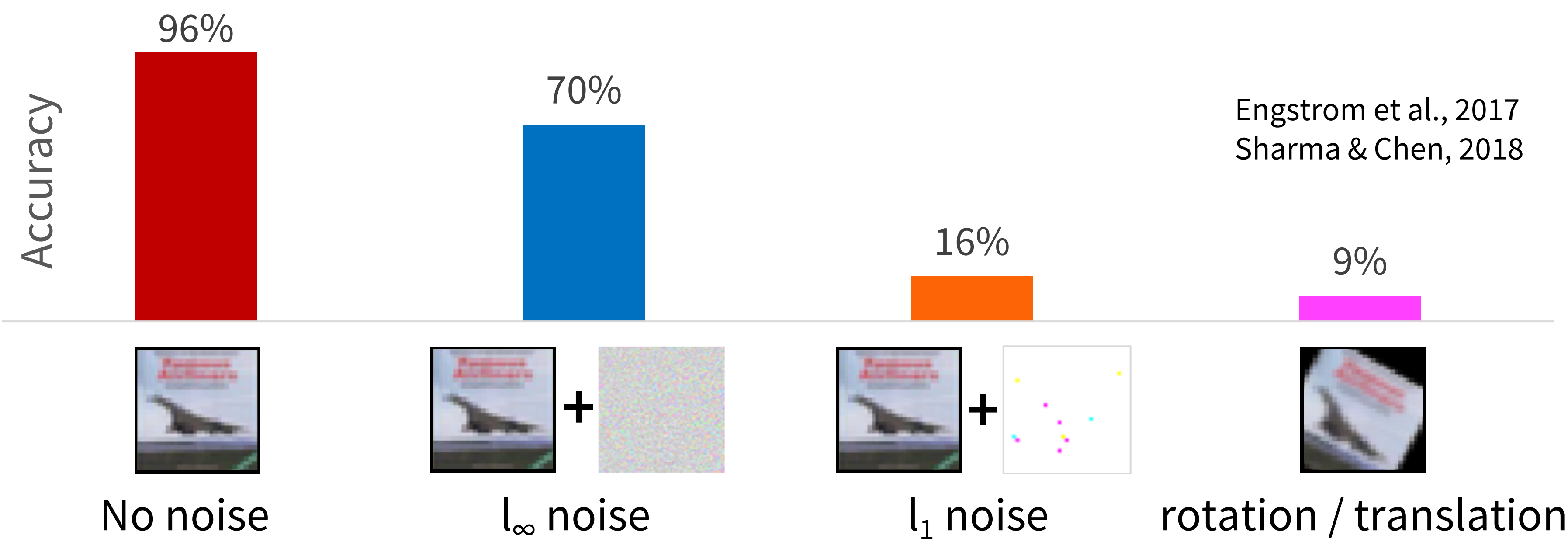
- Overfitting to the toy problem happens and is harmful

- The “non-toy” version of the problem is not actually that relevant for computer security
  (except for ad-blocking)

Beyond the toy game

Issue: defenses do not generalize

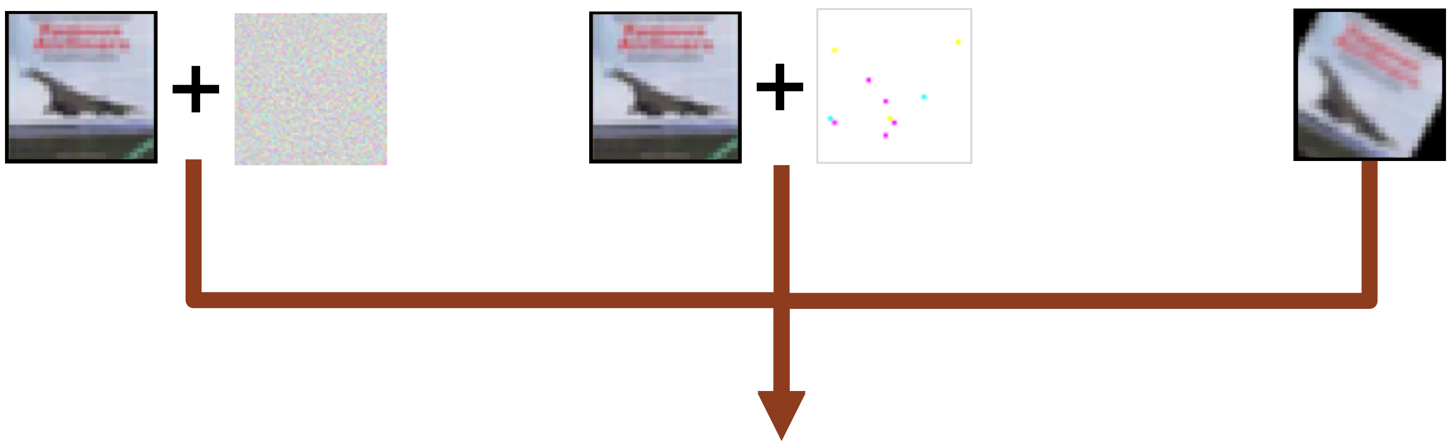
Example: training against l_∞-bounded noise on CIFAR10



Robustness to one type can **increase** vulnerability to others

Robustness to more perturbation types

$S_1 = \{\delta: \|\delta\|_\infty \leq \varepsilon_\infty\}$ $S_2 = \{\delta: \|\delta\|_1 \leq \varepsilon_1\}$ $S_3 = \{\delta:$ «small rotation»$\}$



$\mathbf{S} = S_1 \cup S_2 \cup S_3$

- Pick worst-case adversarial example from **S**
- Train the model on that example

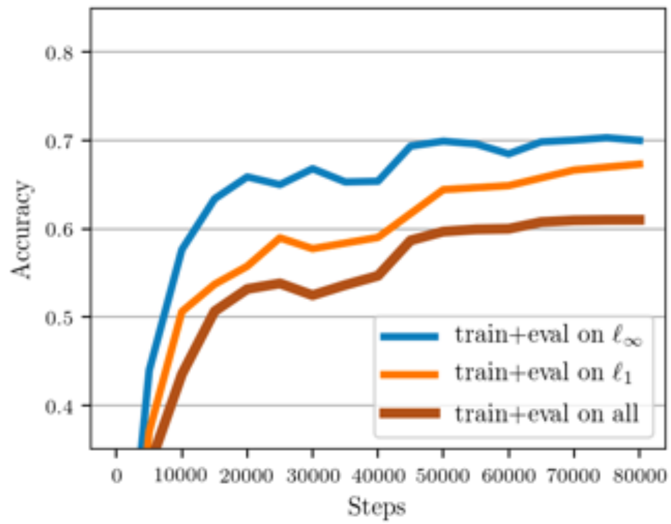
T & Boneh, "Adversarial Training and Robustness for Multiple Perturbations", NeurIPS 2019

Empirical multi-perturbation robustness

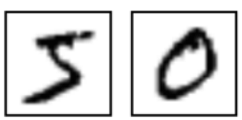
CIFAR10:

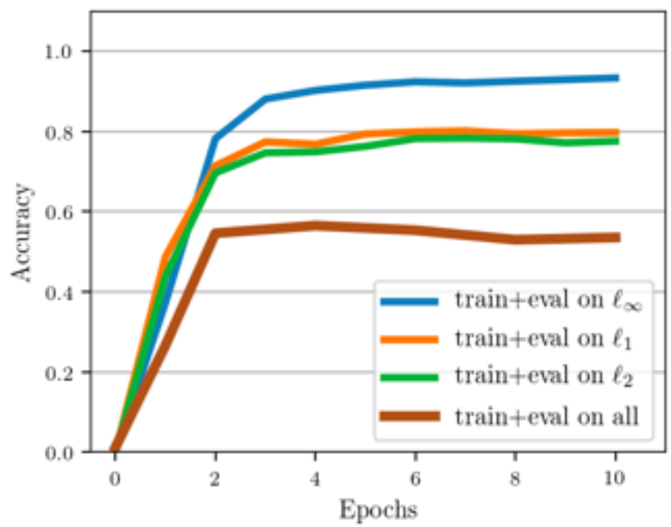
ship

dog



MNIST:





T & Boneh**,** “Adversarial Training and Robustness for Multiple Perturbations”, NeurIPS 2019

Empirical multi-perturbation robustness

CIFAR10:

ship dog

MNIST:

Current defenses scale poorly to multiple perturbations

We also prove that a robustness tradeoff is *inherent* for simple data distributions

T & Boneh, "Adversarial Training and Robustness for Multiple Perturbations", NeurIPS 2019

Outline

- Most papers study a “toy” problem
  Solving it is not useful per se, but maybe we’ll find new insights or techniques

- Going beyond this toy problem (even slightly) is hard

- Overfitting to the toy problem happens and is harmful

- The “non-toy” version of the problem is not actually that relevant for computer security
  (except for ad-blocking)

Invariance adversarial examples

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

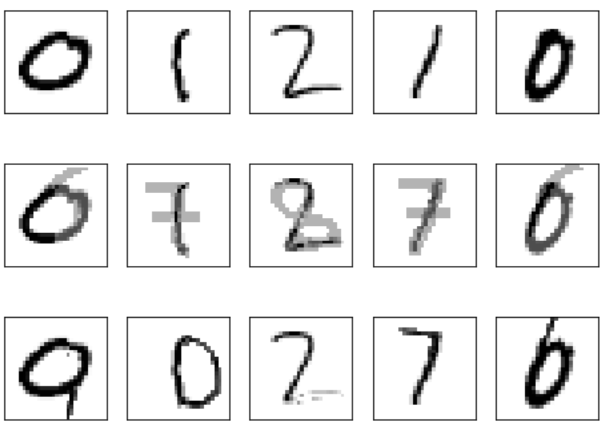
Highest robustness claims in the literature:

- 80% robust accuracy to $l_0 = 30$ 
- **Certified** 85% robust accuracy to $l_\infty = 0.4$ 

natural

$l_\infty \leq 0.4$

$l_0 \leq 30$



Robustness considered harmful

Jacobsen **et al.**, "Exploiting Excessive Invariance caused by Norm-Bounded Adversarial Robustness", 2019

Invariance adversarial examples

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

Highest robustness claims in the literature:

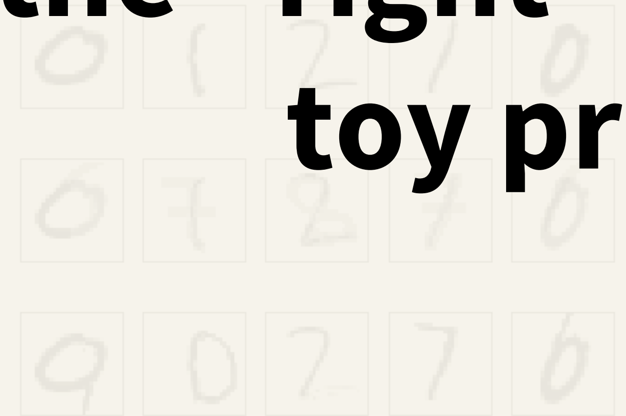
- 80% robust accuracy to $l_0 = 30$
- **Certified** 85% robust accuracy to $l_\infty = 0.4$

We do not even know how to set the "right" bounds for the toy problem

natural

$l_\infty \leq 0.4$

$l_0 \leq 30$



Robustness considered harmful

Jacobsen et al., "Exploiting Excessive Invariance caused by Norm-Bounded Adversarial Robustness", 2019

Adversarial examples are hard!

- Most current work: small progress on the relaxed game
- Moving towards the standard game is hard
  - Even robustness to 2-3 perturbations types is tricky
  - **How would we even enumerate all necessary perturbations?**
- Over-optimizing robustness is harmful
  - **How do we set the right bounds?**
- **We need a formal model of perceptual similarity**
  - But then we’ve probably solved all of computer vision anyhow…

Outline

- Most papers study a “toy” problem
  Solving it is not useful per se, but maybe we’ll find new insights or techniques

- Going beyond this toy problem (even slightly) is hard

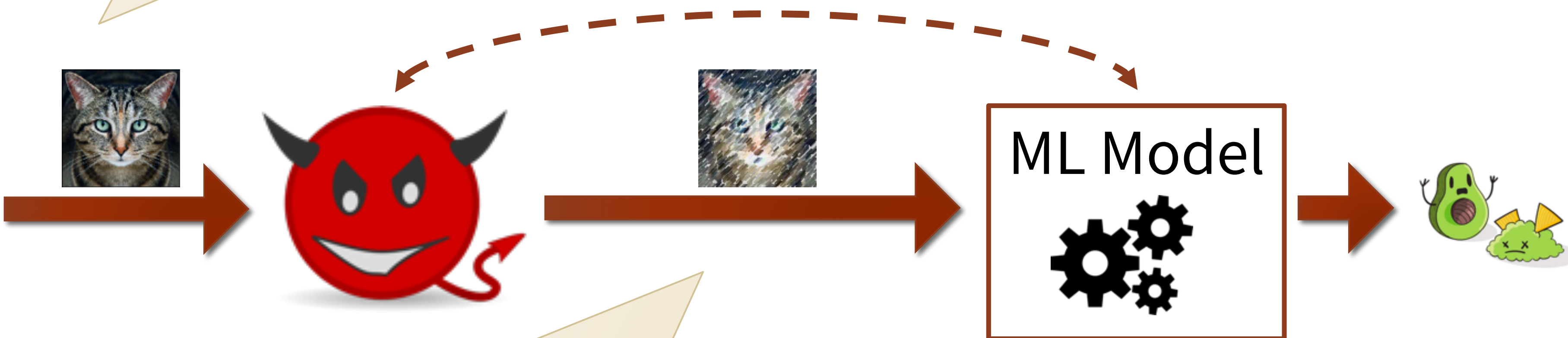
- Overfitting to the toy problem happens and is harmful

- The “non-toy” version of the problem is not actually that relevant for computer security
  (except for ad-blocking)

Recap on the standard game

Adversary is given an input **x** from a data distribution

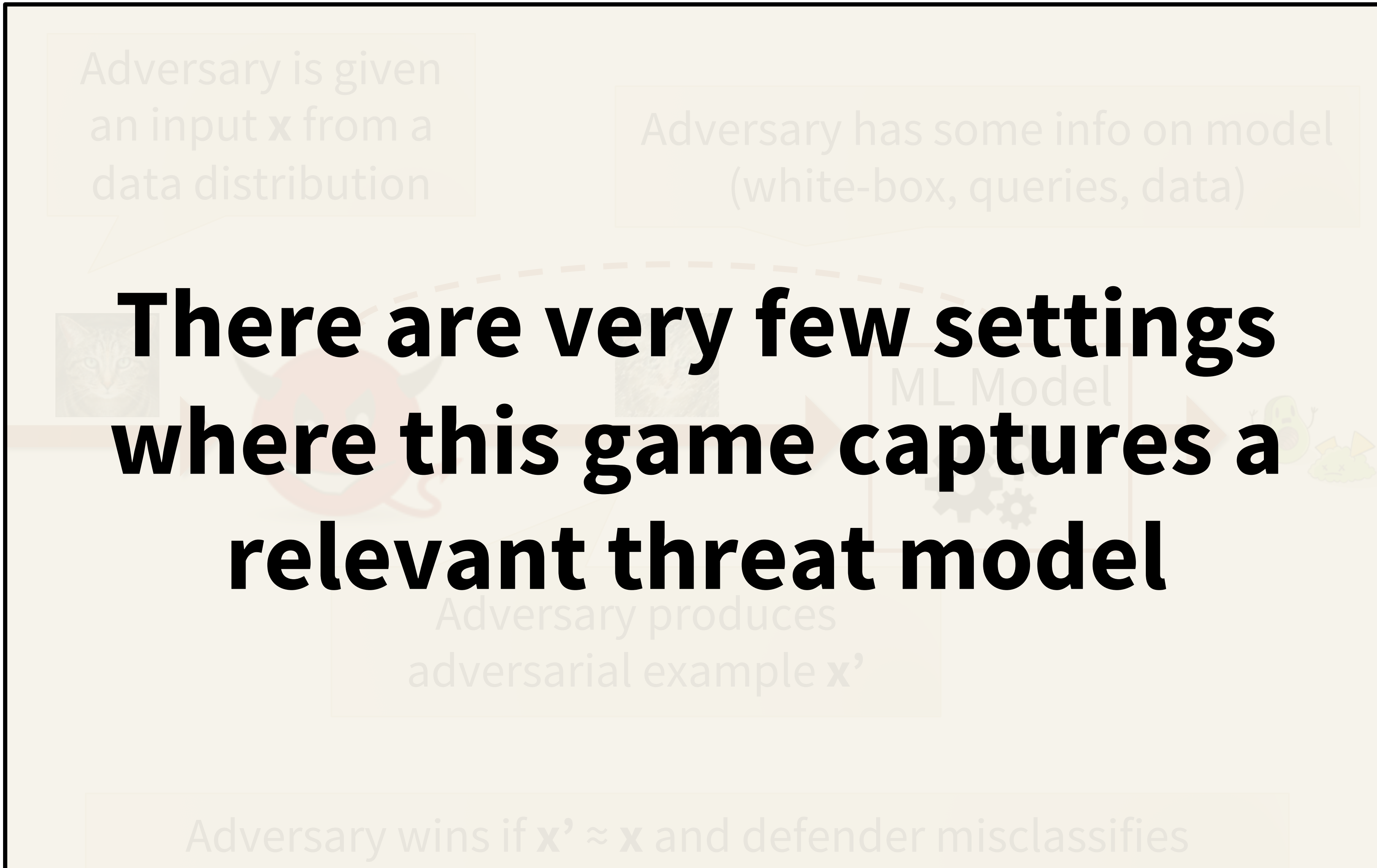
Adversary has some info on model (white-box, queries, data)



Adversary produces adversarial example **x’**

Adversary wins if **x’** ≈ **x** and defender misclassifies

Recap on the standard game

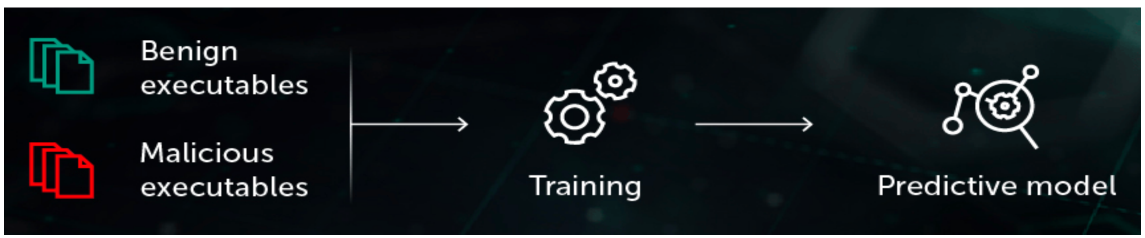


There are very few settings where this game captures a relevant threat model

ML in security/safety critical environments

Fool self-driving cars' street-sign detection

[Eykholt et al. 2017+**2018**]



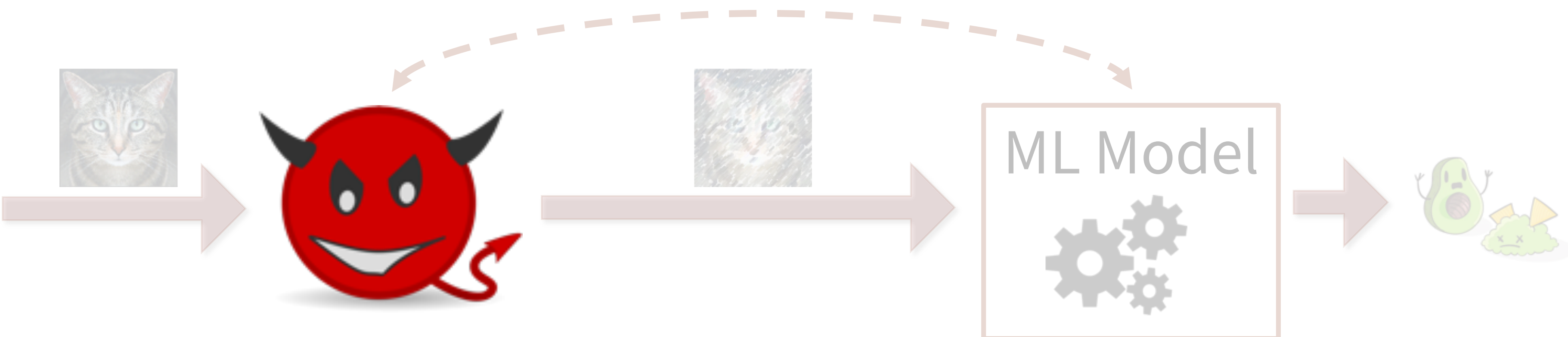
Evade malware detection

[Grosse et al. 2018]

Fool visual ad-blockers

[**T** et al. 2019]

Is the standard game relevant?

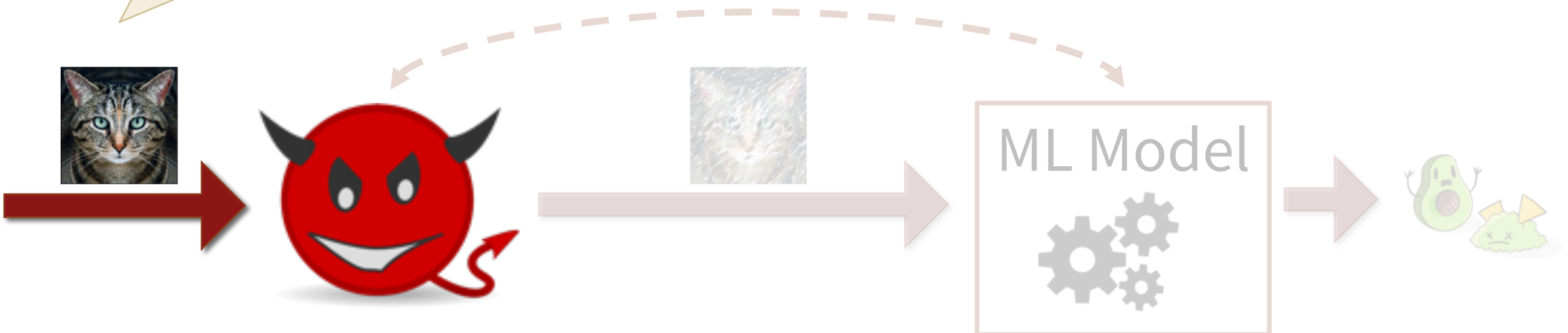

ML Model

Is the standard game relevant?

ABP

Is there an adversary?

Adversary is given an input **x** from a data distribution



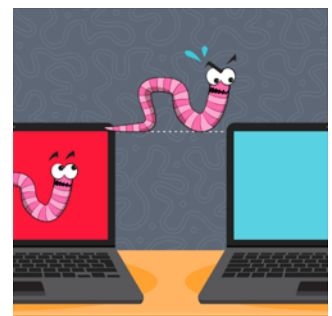
Is the standard game relevant?

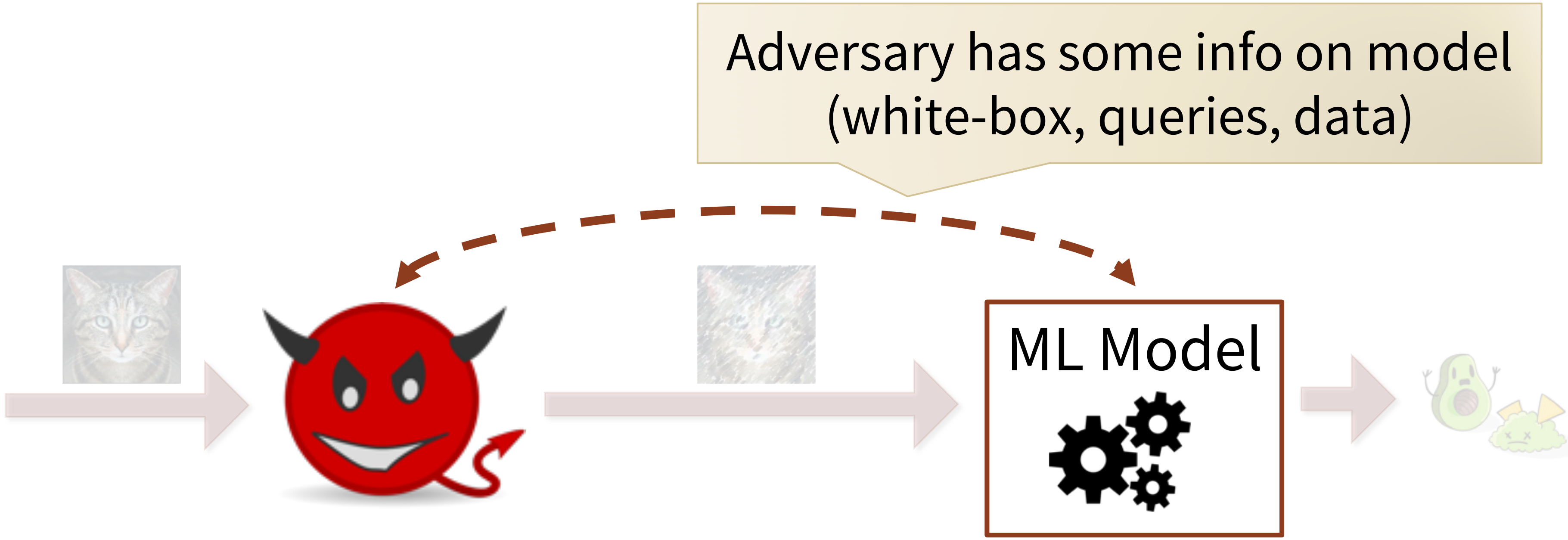
ABP

Is there an adversary?

Is average-case success important?

(Adv cannot choose which inputs to attack)




Adversary has some info on model
(white-box, queries, data)
ML Model

Is the standard game relevant?

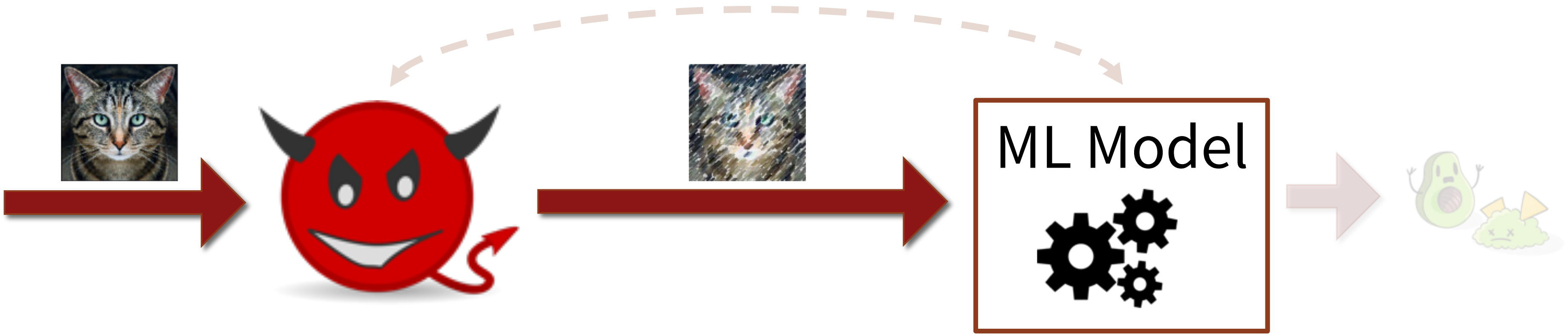
	Stop sign	Malware	ABP
Is there an adversary?	✗	✓	✓
Average-case success?	✗	✗	✓

Model access?
(white-box, queries, data)

App Review

Adblock
Addons

✗	✗	✓

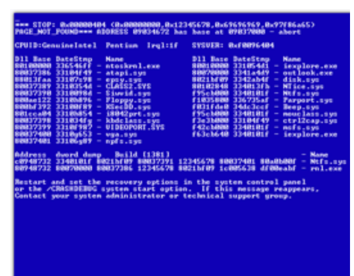


Adversary wins if **x’** ≈ **x** and defender misclassifies

Is the standard game relevant?

			ABP
Is there an adversary?	✗	✓	✓
Average-case success?	✗	✗	✓
Access to model?	✗	✗	✓

Should attacks preserve semantics?
(or be fully imperceptible)

		
✗	✓	✓

Is the standard game relevant?

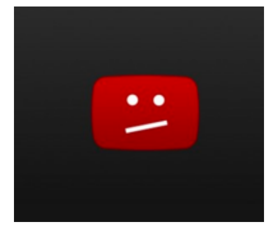
			ABP
Is there an adversary?	✗	✓	✓
Average-case success?	✗	✗	✓
Access to model?	✗	✗	✓
Semantics-preserving perturbations?	✗	✓	✓

Unless the answer to all these questions is *Yes*, the standard game of adversarial examples is not the right threat model

Where else could the game be relevant?

Technology

Inside YouTube's struggles to shut down video of the New Zealand shooting — and the humans who outsmarted its systems



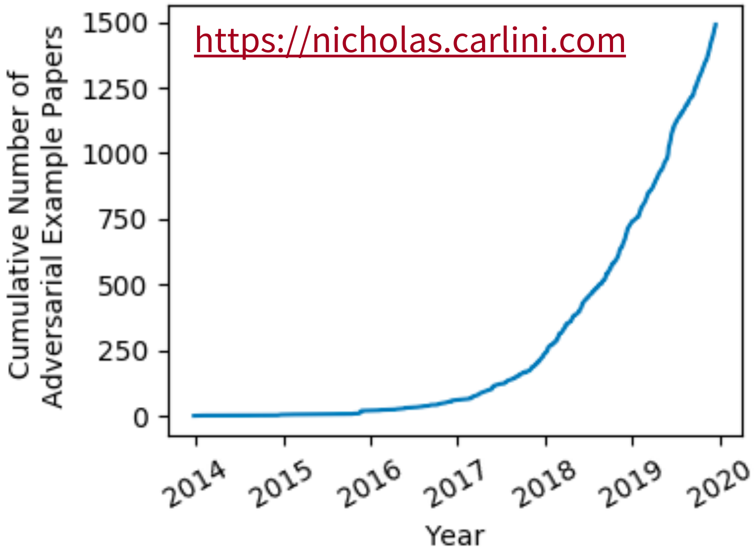
Anti-phishing

Content takedown

Common theme: human-in-the-loop!

(Adversary wants to fools ML without disrupting UX)

Steps forward



Most of these papers consider the relaxed game

Progress on this game is not useful ***per se***

For safety-critical ML (e.g., self-driving):

- There is no adversary (but worst-case analysis can be useful)
- Consider "natural" perturbations (fog, snow, lighting, angles, etc.)

For *real* security-critical ML (e.g., malware detection):

- Attackers often care about breaking in once (analyzing static classifiers is not very useful)
- Security through obscurity (restricted model access) "works" in practice

Maybe we do not need 10x more papers… just the right ones

Backup slides

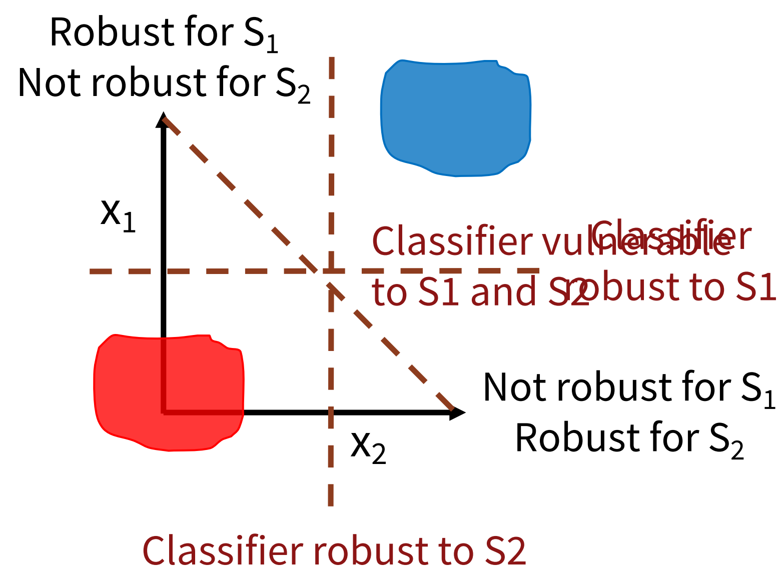
The multi-perturbation robustness trade-off

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

Answer: in general, NO!

There exist "mutually exclusive perturbations" (MEPs)

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

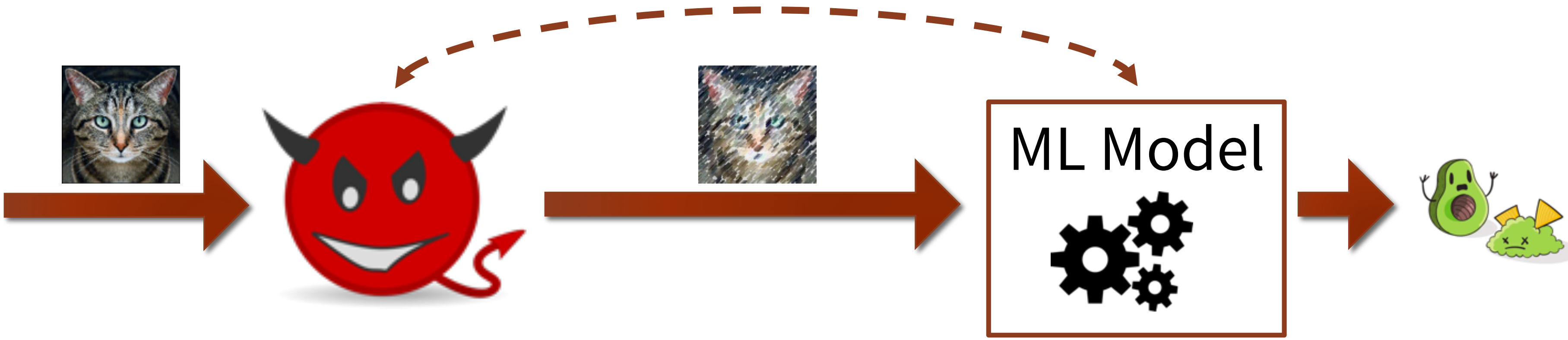


Formally, we show that for a simple Gaussian binary classification task:

- l_1 and l_∞ noise are MEPs
- l_∞ noise and spatial perturbations are MEPs

T & Boneh, "Adversarial Training and Robustness for Multiple Perturbations", NeurIPS 2019

The standard game [Gilmer et al. 2018]



1. Adversary is given input **x** from some data distribution
2. Adversary gets some information on model:
   - Access to model parameters (white-box)
   - Query access
   - Access to similar training data
3. Adversary outputs an adversarial example **x’**
4. Defender classifies **x’**

Adversary wins if **x’** ≈ **x** and defender misclassifies