A Tour of Machine Learning Security

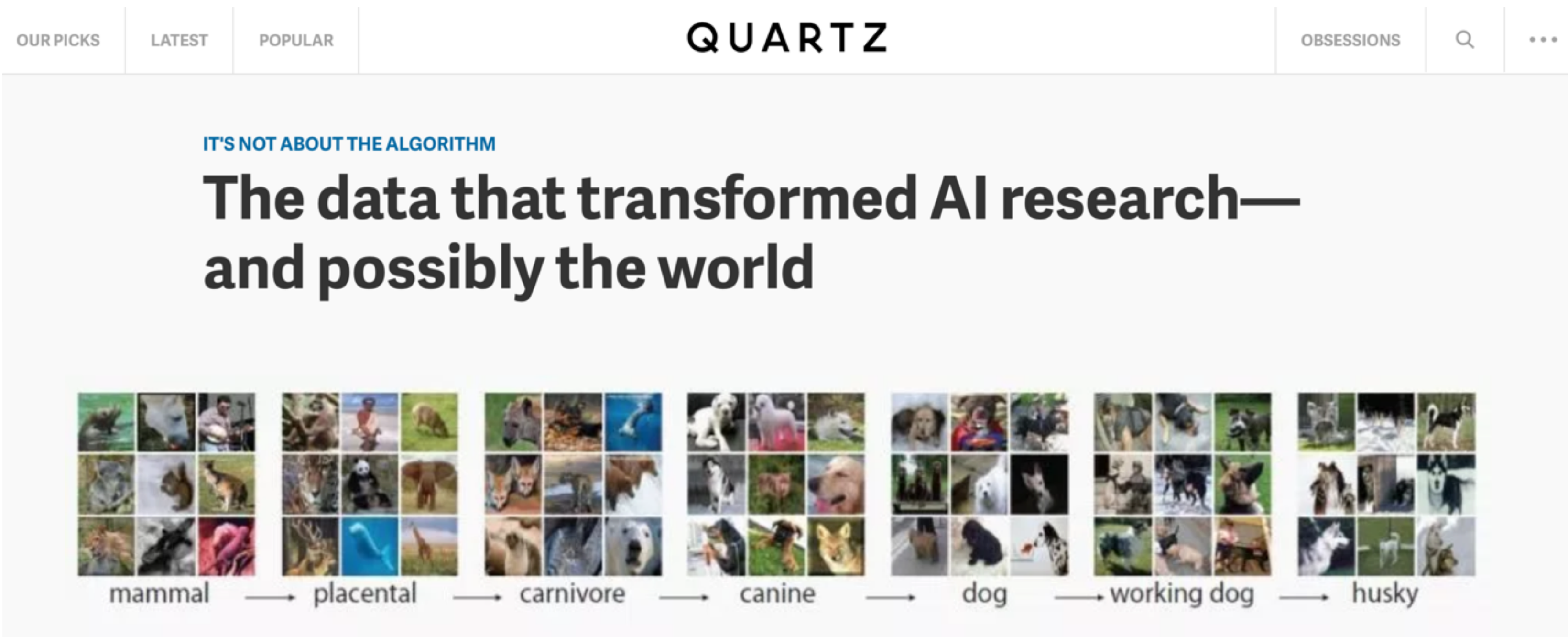
Florian Tramèr

Intel, Santa Clara, CA
August 30th 2018

Stanford University

The Deep Learning Revolution

First they came for images…



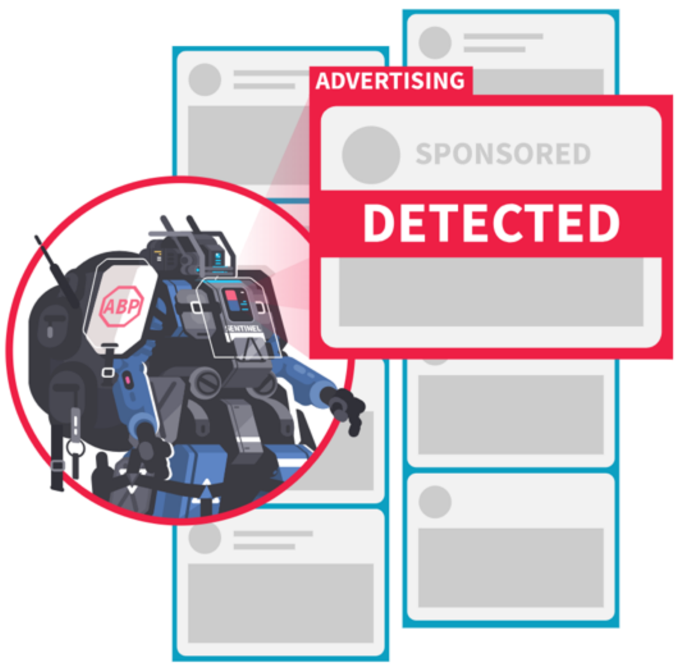
The Deep Learning Revolution

And then everything else…

The ML Revolution

Including things that likely won't work…

DEVELOPED BY
Adblock **Plus**



Meet Sentinel

the artificial intelligence ad detector.

With your help, Sentinel could be the future of ad blocking.

Sentinel uses machine learning to detect Facebook ads visually. The more Facebook screenshots you submit, the faster Sentinel will learn.

Team up with Sentinel for the future of ad blocking!

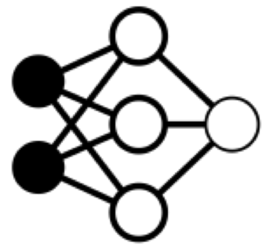
FEED SENTINEL

What does this mean for privacy & security?

Crypto, Trusted hardware

Privacy & integrity

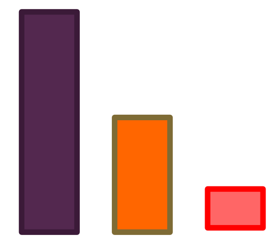
Outsourced learning



Differential privacy

Data inference
Model theft

Test outputs



Training data

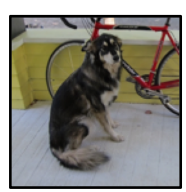
Data poisoning

Robust statistics

Bl ??? in

Adversarial examples

Test data



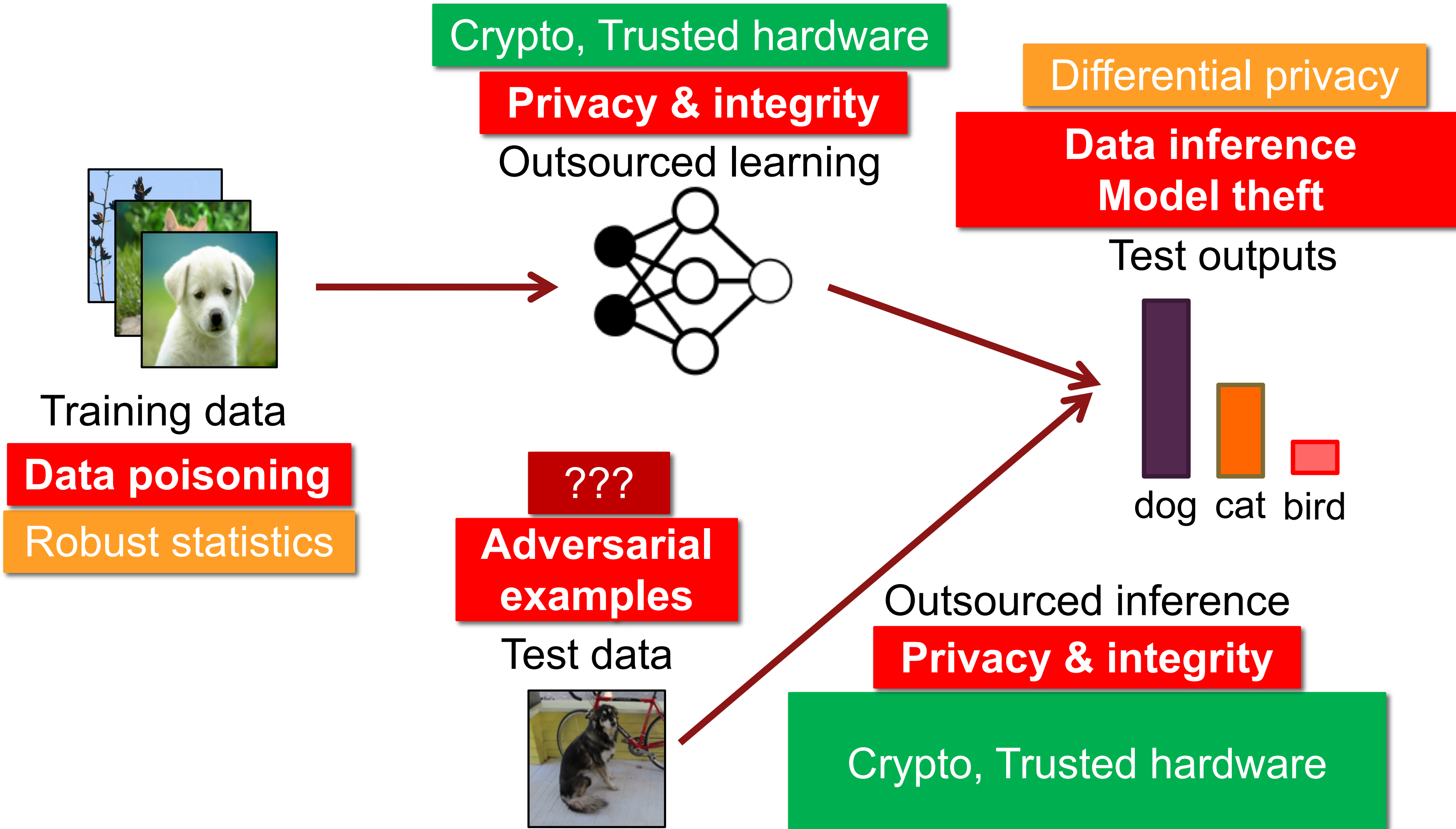
Outsourced inference

Privacy & integrity

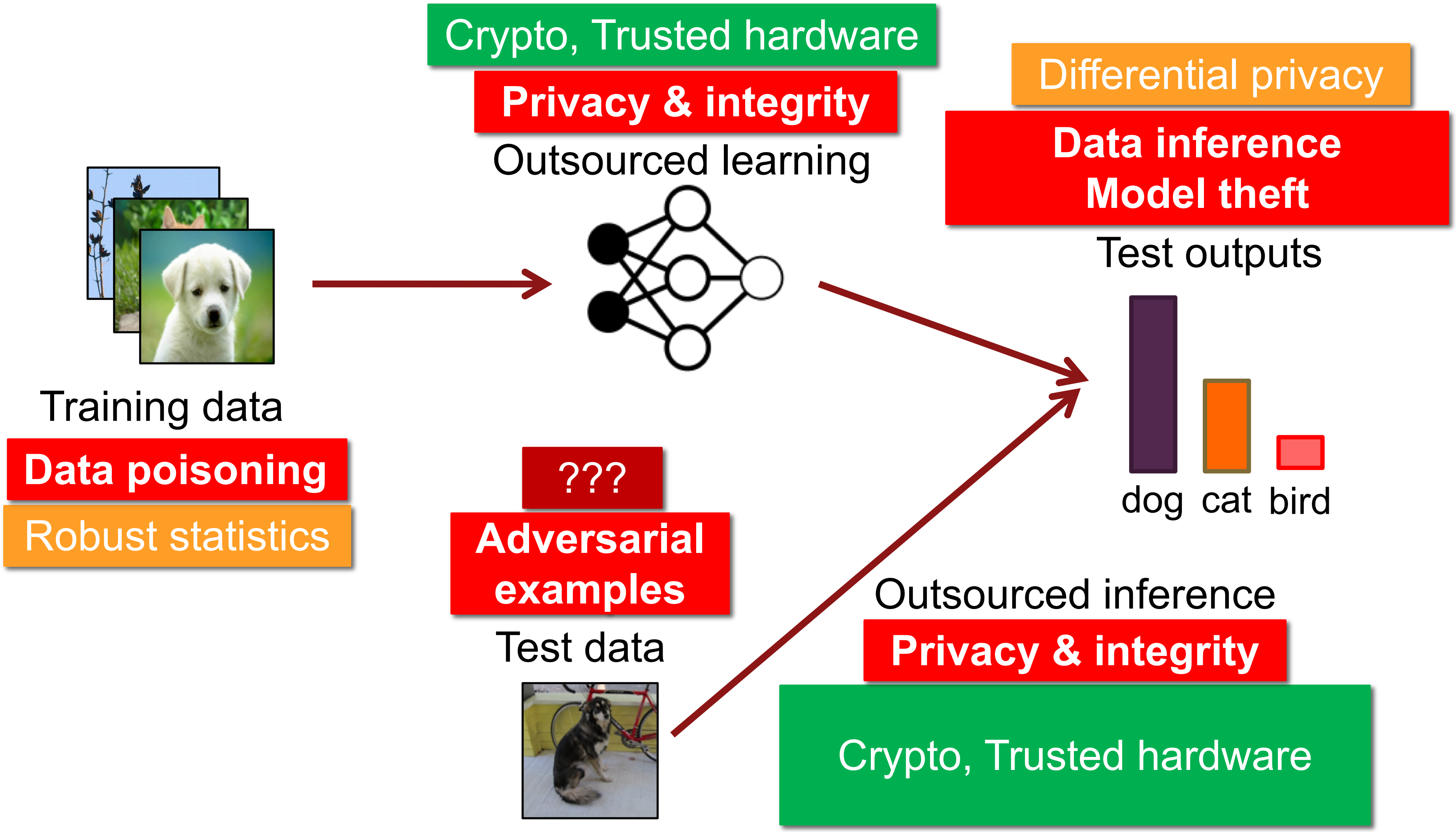
Crypto, Trusted hardware

Adapted from (Goodfellow 2018)

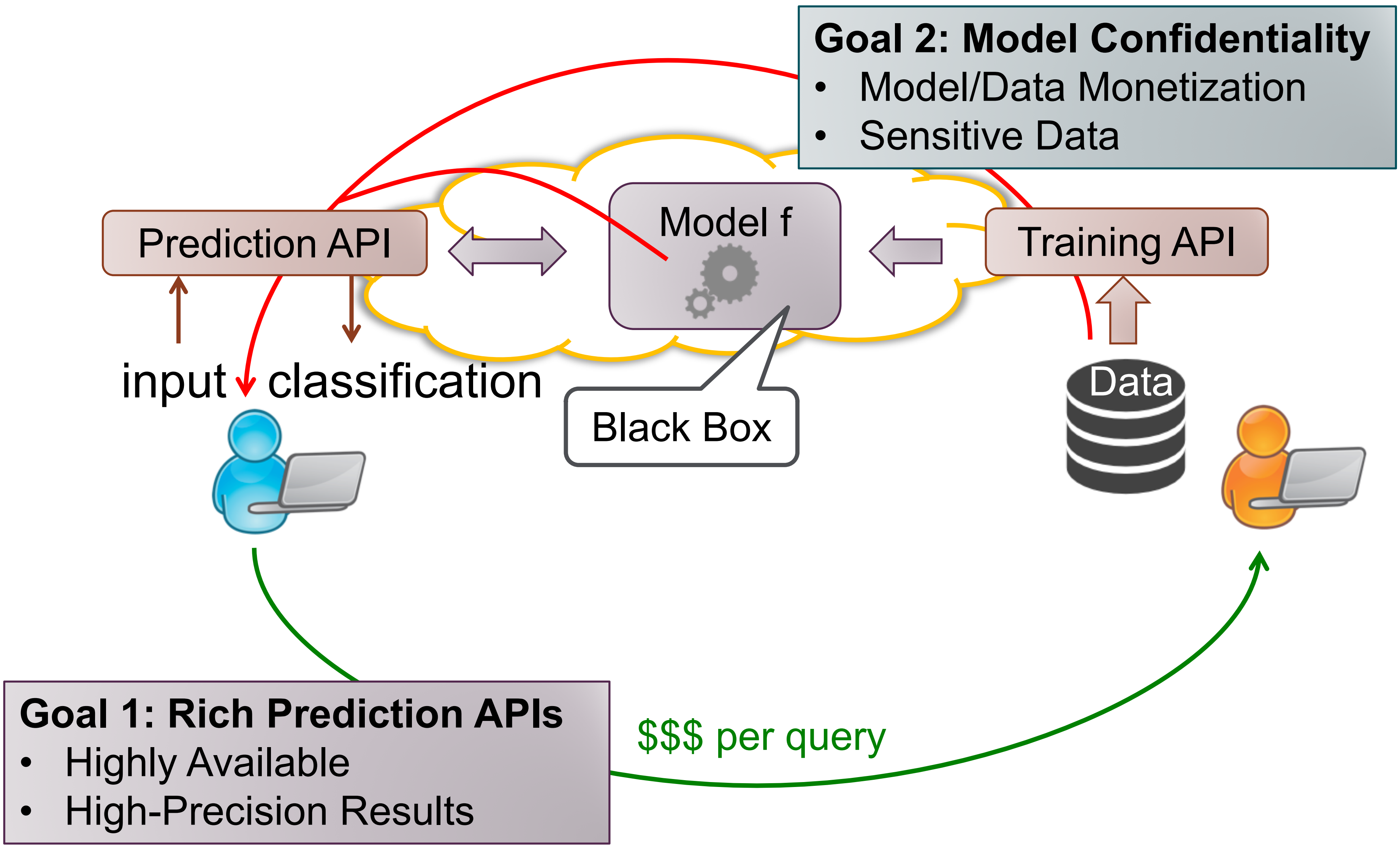
This talk: security of deployed models



Stealing ML Models

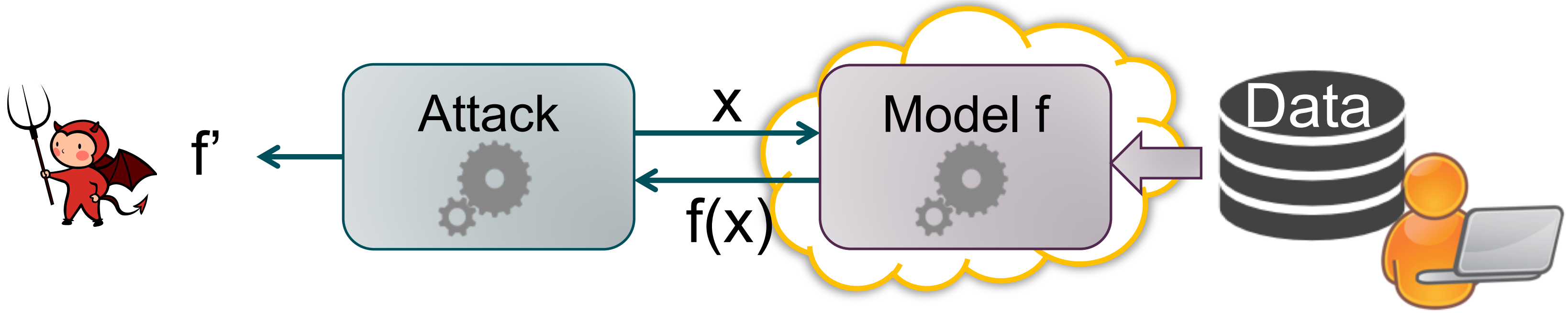


Machine Learning as a Service



Model Extraction

Goal: Adversarial client learns close approximation of f using as few queries as possible



Applications:

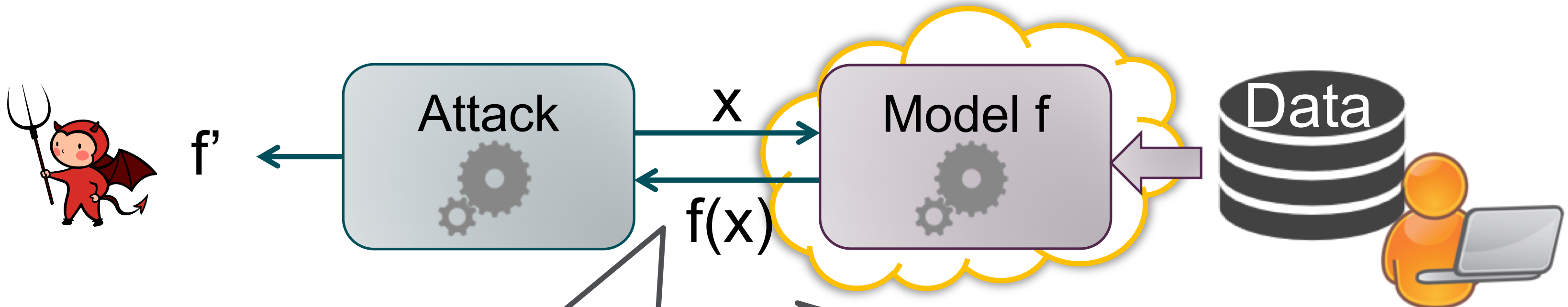
1) Undermine pay-for-prediction pricing model

2) "White-box" attacks:

- Infer private training data
- Model evasion (adversarial examples)

Model Extraction

Goal: Adversarial client learns close approximation of f using as few queries as possible



No! Prediction APIs return fine-grained information that makes extracting **much easier** than learning

Isn't this "just Machine Learning"?

Learning vs Extraction

	Learning f(x)	Extracting f(x)
Function to learn	Noisy real-world phenomenon	“Simple” deterministic function f(x)

Learning vs Extraction

	Learning f(x)	Extracting f(x)
Function to learn	Noisy real-world phenomenon	“Simple” deterministic function f(x)
Available labels	hard labels (e.g., “cat”, “dog”, …)	Depending on API: - Hard labels - Soft labels (class probas) - Gradients (Milli et al. 2018)

Learning vs Extraction

	Learning f(x)	Extracting f(x)
Function to learn	Noisy real-world phenomenon	"Simple" deterministic function f(x)
Available labels	hard labels (e.g., "cat", "dog", …)	Depending on API: - Hard labels - Soft labels (class probas) - Gradients (Milli et al. 2018)
Labeling function	Humans, real-world data collection	Query f(x) **on any input x** => No need for labeled data => Queries can be **adaptive**

Learning vs Extraction for specific models

	Learning f(x)	**Extracting f(x)**
Logistic Regression	|Data| ≈ 10 * |Features|	- Hard labels only: (Loyd & Meek) - With confidences: simple system of equations (**T** et al.) |Data| = |Features| + cte

Learning vs Extraction for specific models

	Learning f(x)	**Extracting f(x)**
Logistic Regression	|Data| ≈ 10 * |Features|	- Hard labels only: (Loyd & Meek) - With confidences: simple system of equations (**T** et al.) |Data| = |Features| + cte
Decision Trees	- NP-hard in general - polytime for Boolean trees (Kushilevitz & Mansour)	“Differential testing” algorithm to recover the full tree (**T** et al.)

Learning vs Extraction for specific models

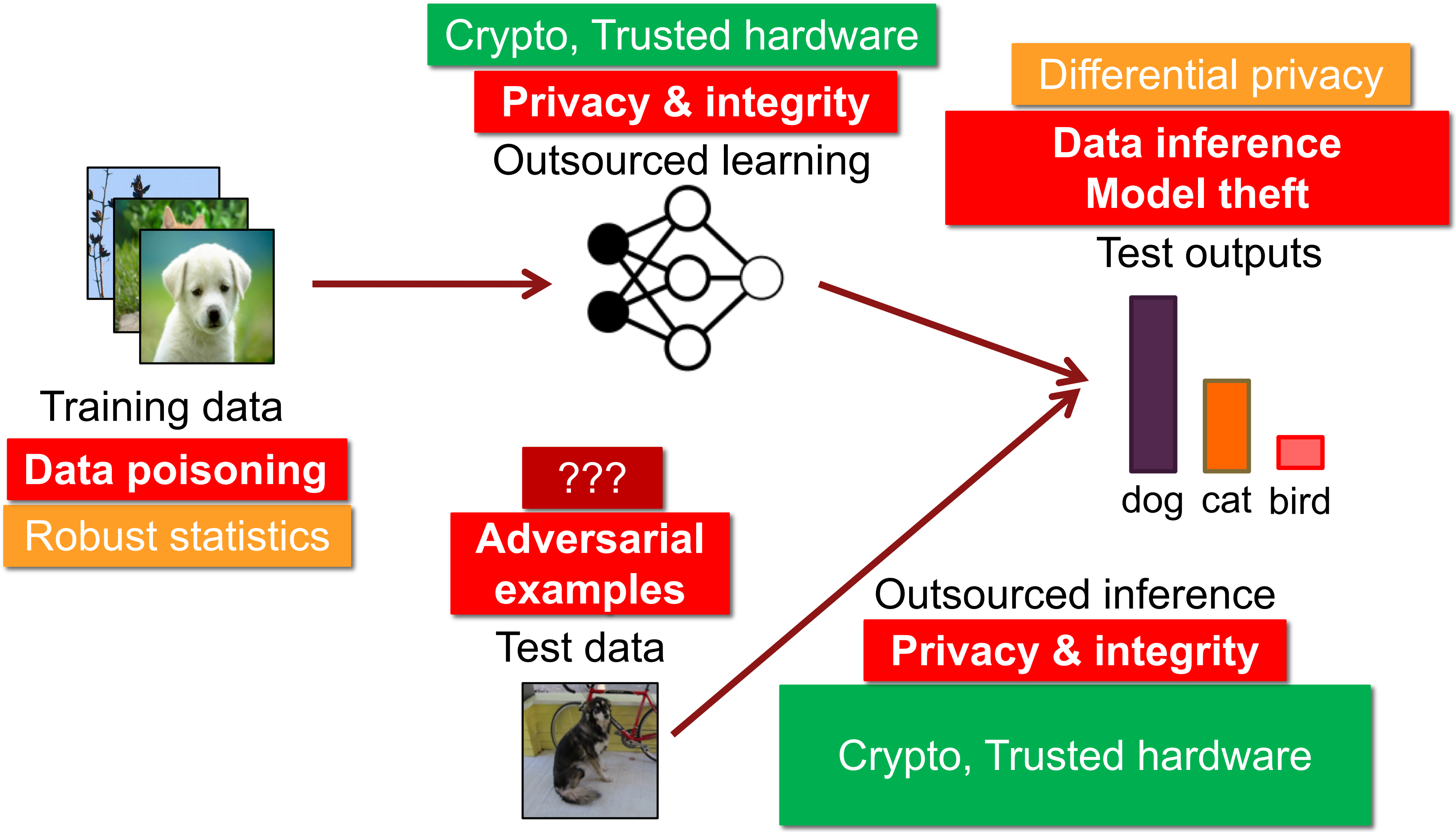
	Learning f(x)	**Extracting f(x)**
Logistic Regression	|Data| ≈ 10 * |Features|	- Hard labels only: (Loyd & Meek) - With confidences: simple system of equations (**T** et al.) |Data| = |Features| + cte
Decision Trees	- NP-hard in general - polytime for Boolean trees (Kushilevitz & Mansour)	“Differential testing” algorithm to recover the full tree (**T** et al.)
Neural Networks	Large models required “The more data the better”	- Distillation (Hinton et al.) Make smaller copy of model from confidence scores - Extraction from hard labels (Papernot et al., **T** et al.)

No quantitative analysis for large neural nets yet

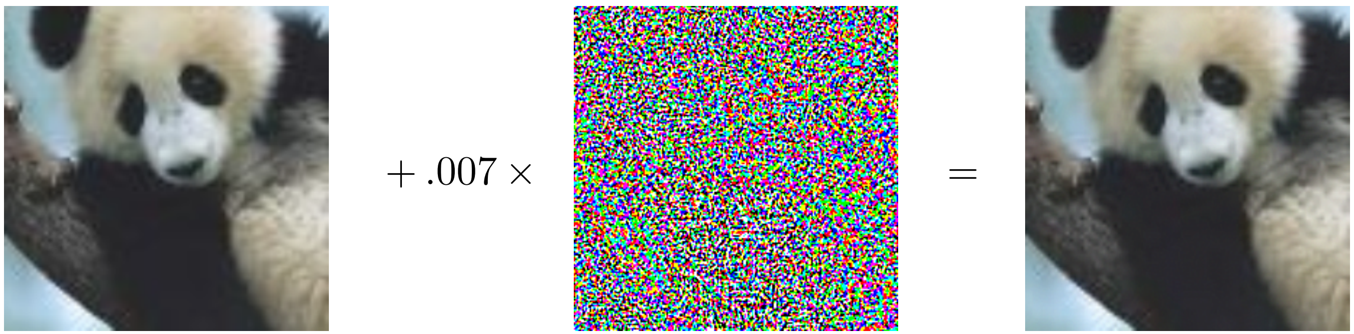
Takeaways

- A “learnable” function cannot be private
- Prediction APIs expose fine-grained information that facilitate model stealing
- Unclear how effective model stealing is for large-scale models

Evading ML Models



ML models make surprising mistakes



Pretty sure this is a panda

I’m certain this is a gibbon

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

Where are the defenses?

- Adversarial training
  Szegedy et al. 2013, Goodfellow et al. 2015,
  Kurakin et al. 2016, **T** et al. 2017,
  Madry et al. 2017, Kannan et al. 2018

Prevent "all/most attacks" **for a given norm ball**

- Convex relaxations with provable guarantees
  Raghunathan et al. 2018, Kolter & Wong 2018, Sinha et al. 2018

- A lot of broken defenses…

Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods

Nicholas Carlini David Wagner

Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples

Anish Athalye [*1] **Nicholas Carlini** [*2] **David Wagner** [2]

Do we have a realistic threat model? (no…)

Current approach:

1. Fix a ”toy” attack model (e.g., some l_∞ ball)
2. Directly optimize over the robustness measure

$\Rightarrow$ Defenses do not generalize to other attack models

$\Rightarrow$ Defenses are meaningless for applied security

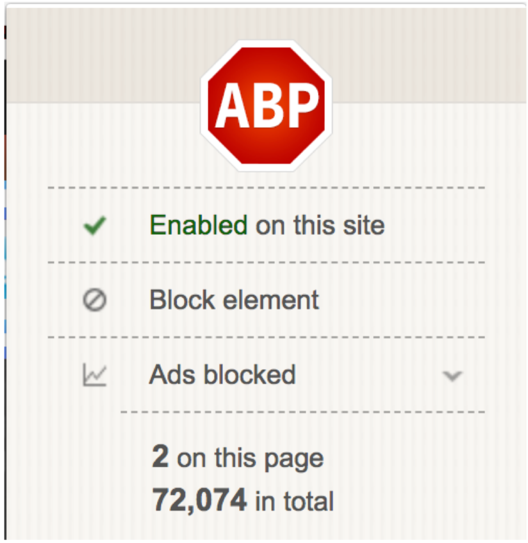
What do we want?

- Model is “always correct” (sure, why not?)
- Model has blind spots that are “hard to find”
    - “Non-information-theoretic” notions of robustness?
    - CAPTCHA threat model is interesting to think about

Adversarial examples are here to stay!

For many things that humans can do "robustly", ML will fail miserably!

A case study on ad blocking



Ad blocking is a "cat & mouse" game

1. Ad blockers build crowd-sourced filter lists
2. Ad providers switch origins / DOM structure
3. Rinse & repeat

(4?) Content provider (e.g., Cloudflare) hosts the ads

A case study on ad blocking

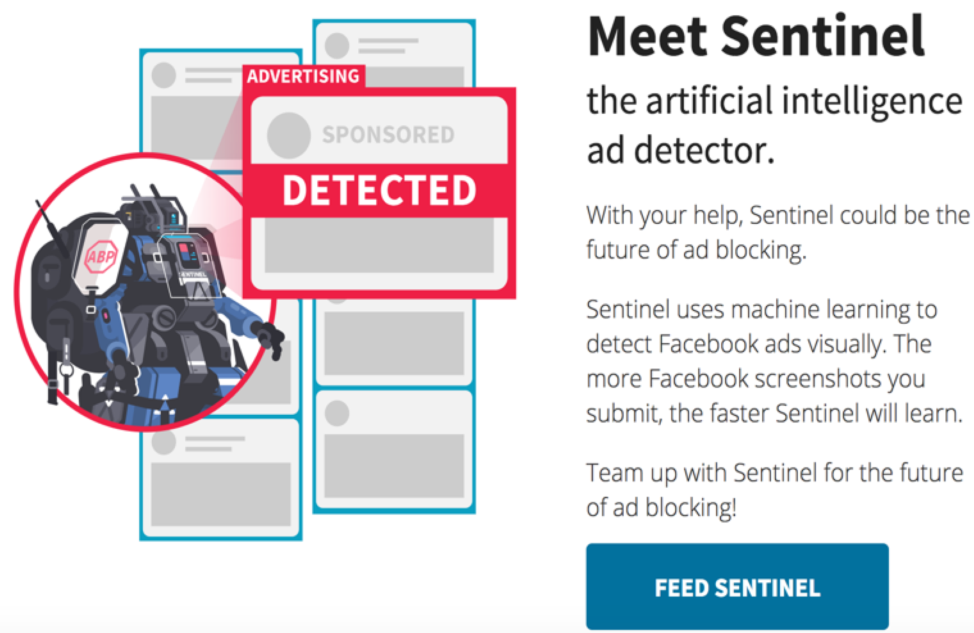
New method: perceptual ad-blocking (Storey et al. 2017)

- Industry/legal trend: ads have to be clearly indicated to humans

If humans can detect ads, so can ML!

*"[…] **we deliberately ignore all signals invisible to humans**, including URLs and markup. Instead we consider visual and behavioral information. […] **We expect perceptual ad blocking to be less prone to an "arms race."***

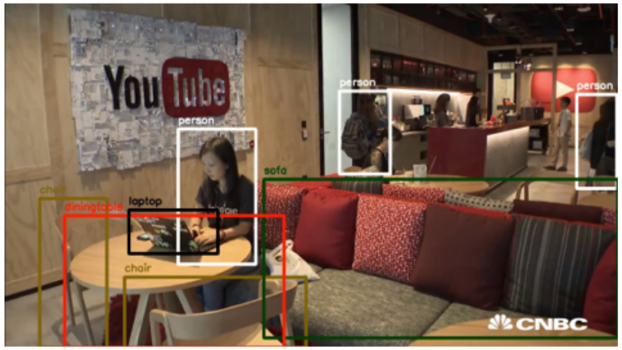
(Storey et al. 2017)



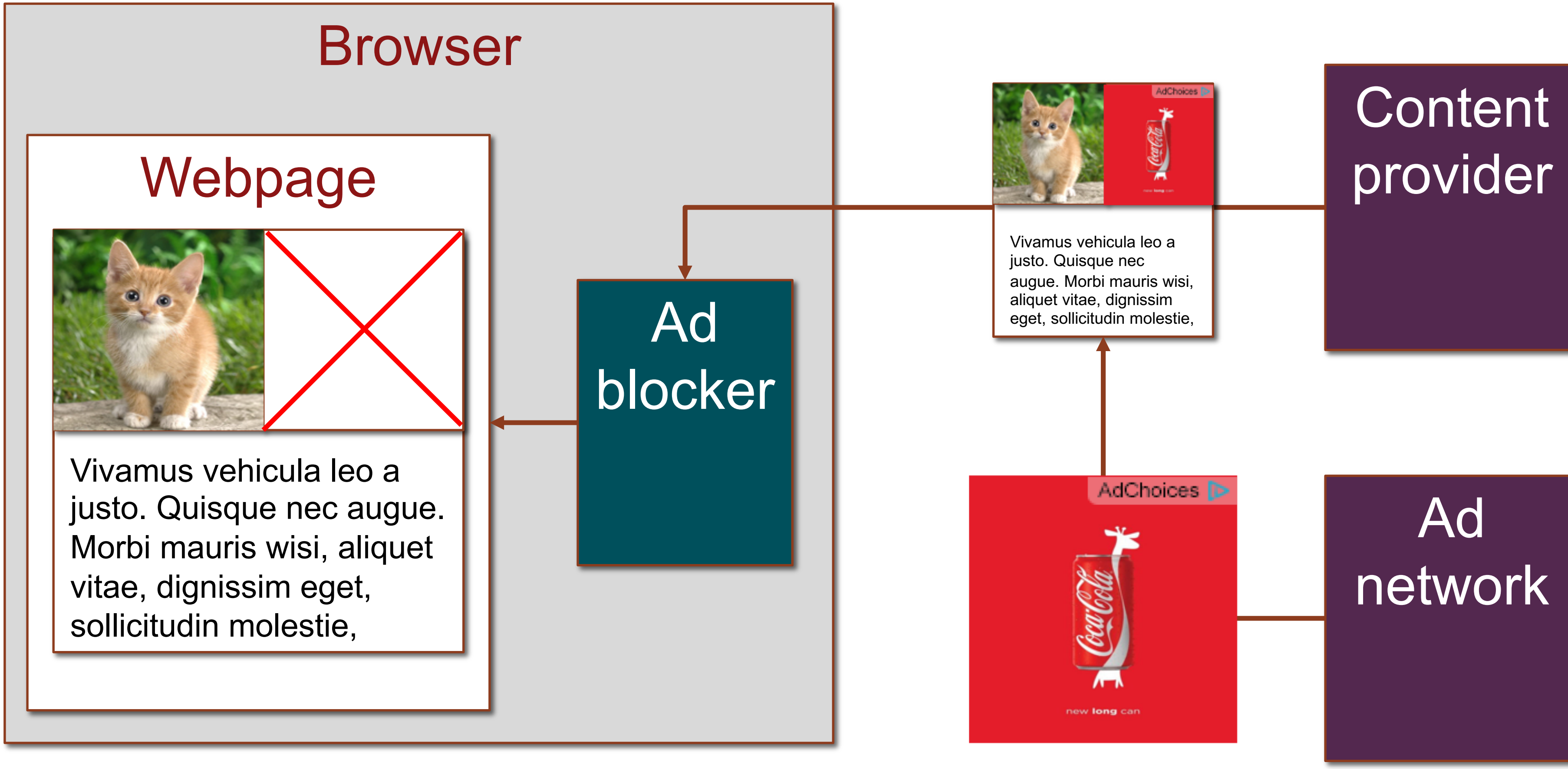
How to detect ads?

1. **“DOM based”**
   - Look for specific ad-cues in the DOM
   - E.g., fuzzy hashing, OCR (Storey et al. 2017)

2. **Machine Learning on full page content**
   - Sentinel approach: train object detector (YOLO) on annotated screenshots

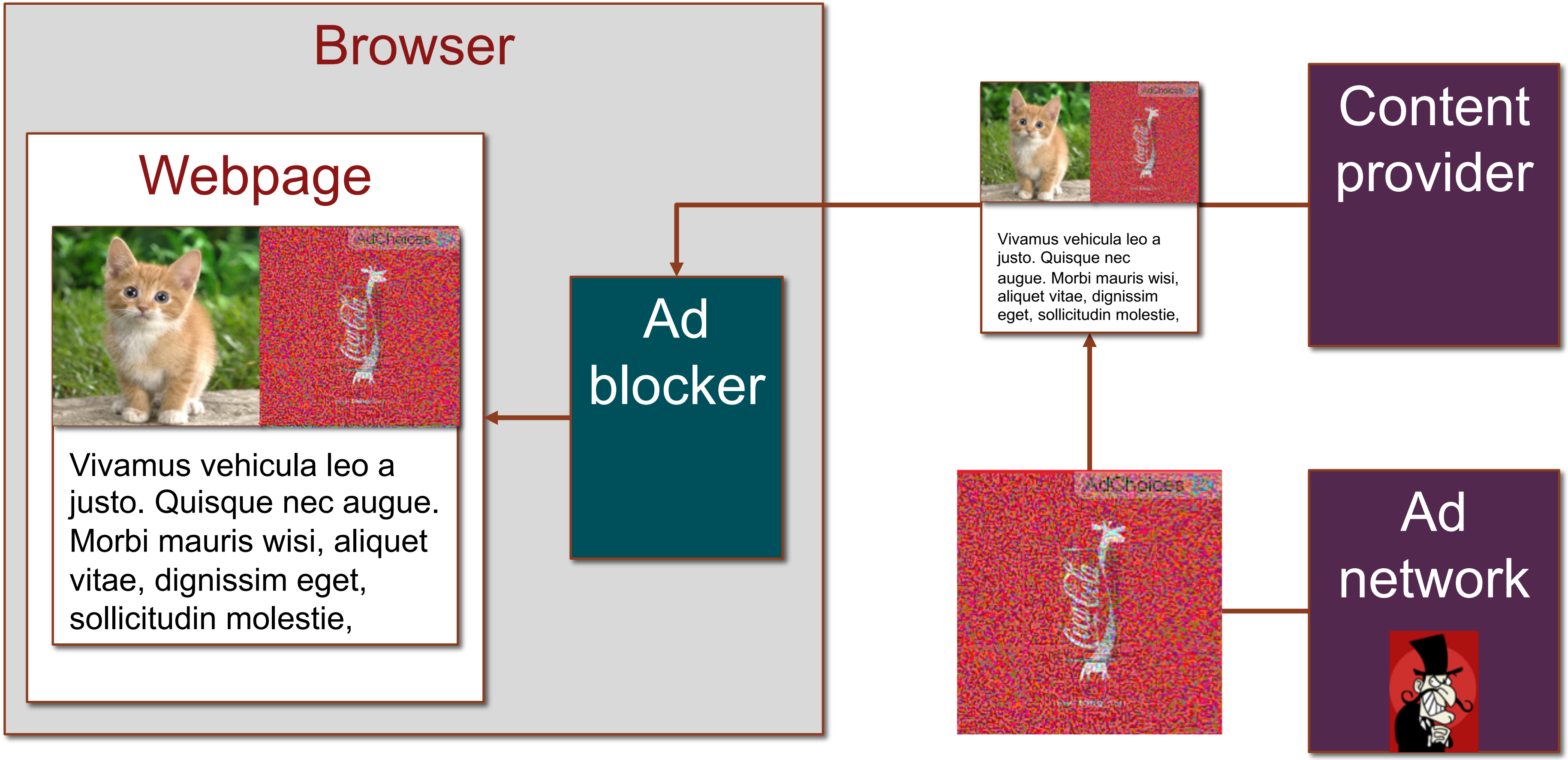


What’s the threat model for perceptual ad-blockers?



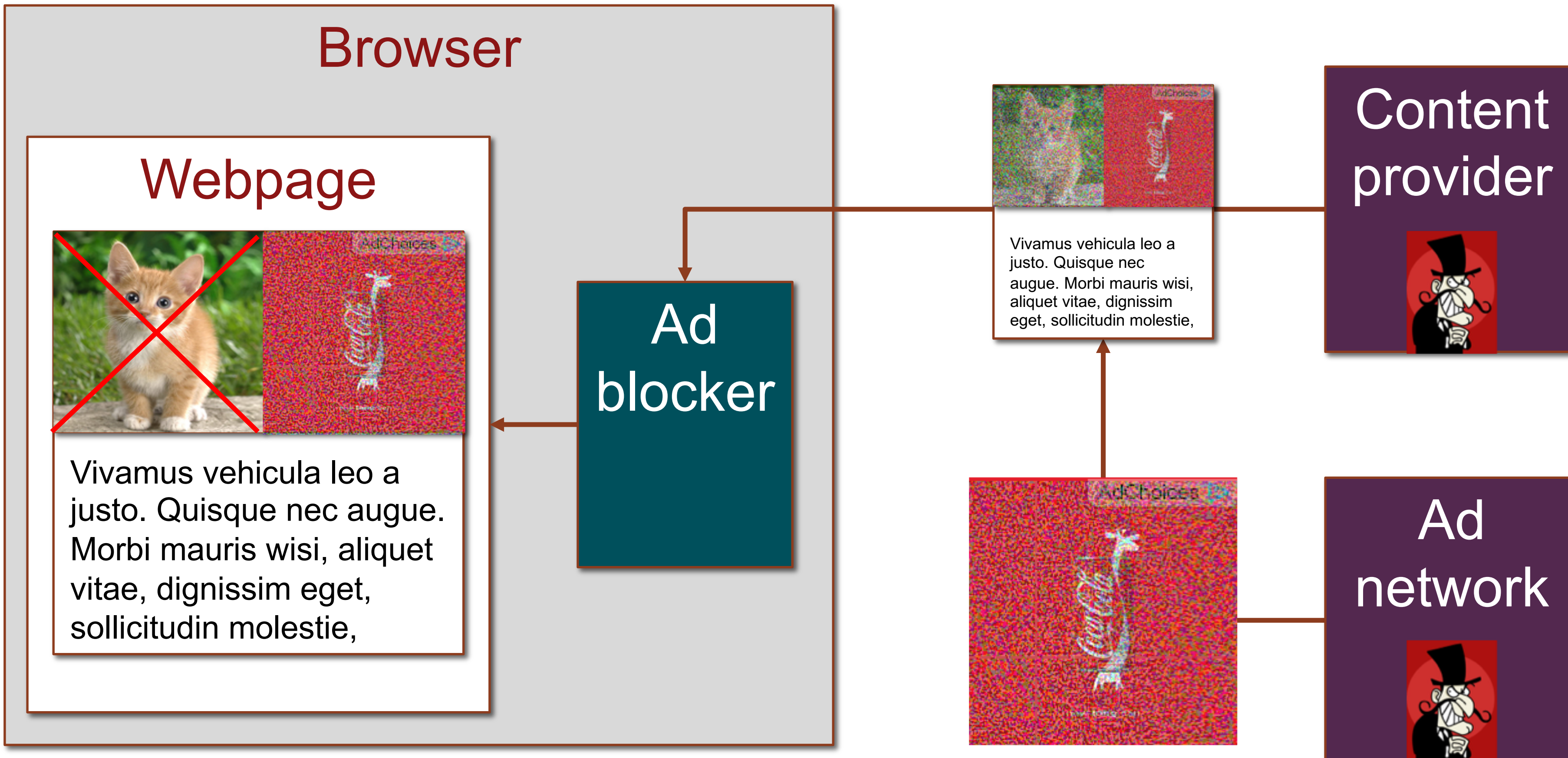
What’s the threat model for perceptual ad-blockers?

1. False Negatives



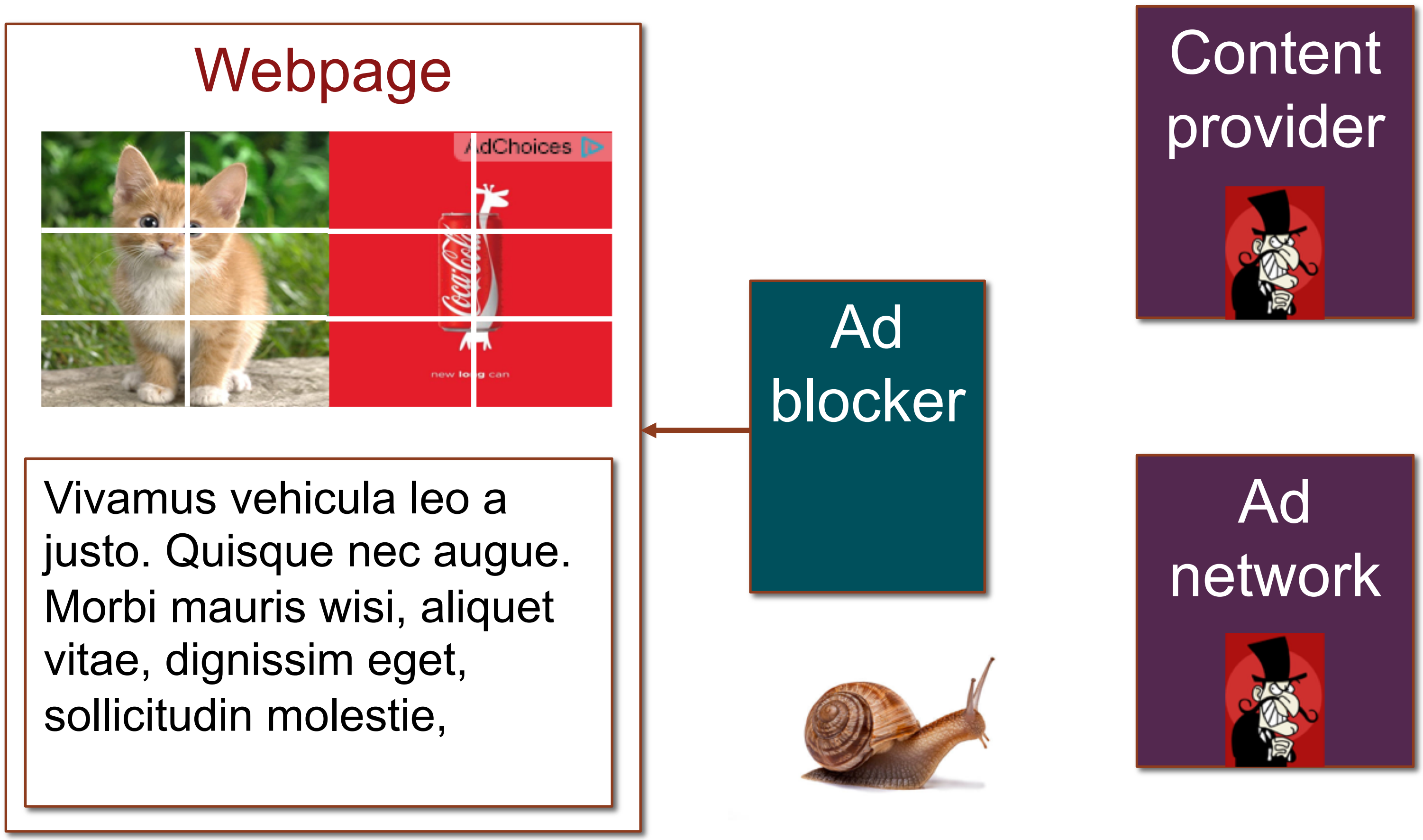
What's the threat model for perceptual ad-blockers?

2. False Positives ("DOS", or ad-blocker detection)



What's the threat model for perceptual ad-blockers?

3. Resource exhaustion (for DOM-based techniques)



What's the threat model for perceptual ad-blockers?

Pretty much the worst possible!

1. **Ad blocker is white-box** (browser extension)
   $\Rightarrow$ Alternative would be a privacy & bandwidth nightmare

2. **Ad blocker operates on (large) digital images**
   $\Rightarrow$ Or can exhaust resources by injecting many small elements

3. **Ad blocker needs to resist adversarial false positives and false negatives**
   $\Rightarrow$ Perturb ads to evade ad blocker
   $\Rightarrow$ Discover ad-blocker by embedding false-negatives
   $\Rightarrow$ Punish ad-block users by perturbing benign content

4. **Updating is more expensive than attacking**

An interesting contrast: CAPTCHAs

Deep ML models can solve text CAPTCHAs!

$\Rightarrow$ Why don’t CAPTCHAs use adversarial examples?

$\Rightarrow$ CAPTCHA $\simeq$ adversarial example for OCR systems

	Model access	Vulnerable to false positives, resource exhaustion	Model Updates
Ad blocker	White-box	Yes	Expensive
CAPTCHA	“Black-box” (not even query access)	No	Cheap (None)

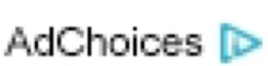
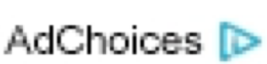
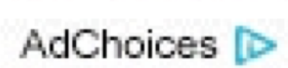
Attacks on perceptual ad-blockers

DOM-based

- Facebook already obfuscates text indicators!

⇒ Cat & mouse game on text obfuscation
⇒ Final step: use a picture of text

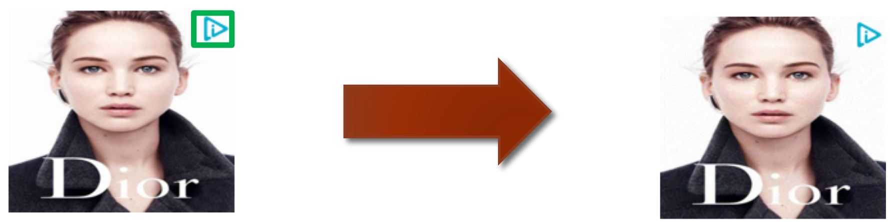
- Dealing with images is hard(er)
  - Adversarial examples
  - DOS (e.g., OCR on 100s of images)

	Original	False positive	False negative
OCR			
Fuzzy hashing			

Attacks on perceptual ad-blockers

ML based

- YOLO to detect AdChoice logo



- YOLO to detect ads “end-to-end” (it works!)

Conclusions

- ML revolution ⇒ rich pipeline with interesting security & privacy problems at every step

- Model stealing
  - One party does the hard work (data labeling, learning)
  - Copying the model is easy with rich prediction APIs
  - Model monetization is tricky

- Model evasion
  - Everything's broken once you add an adversary (and an interesting attack model)
  - Perceptual ad blocking
    - Mimicking human perceptibility is very challenging
    - Ad blocking has the "worst" possible threat model

THANKS