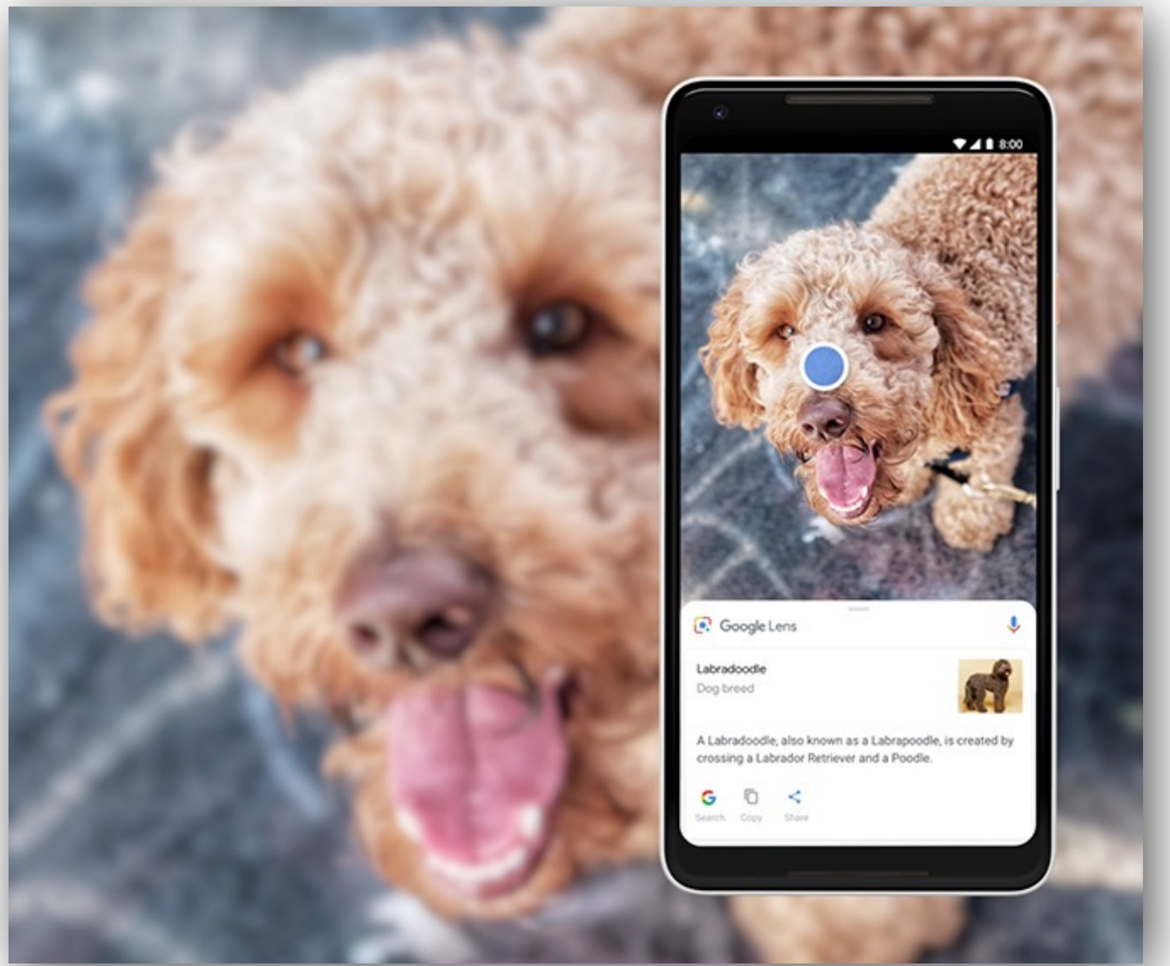
Measuring and Enhancing the Security of Machine Learning

Florian Tramèr

Stanford University

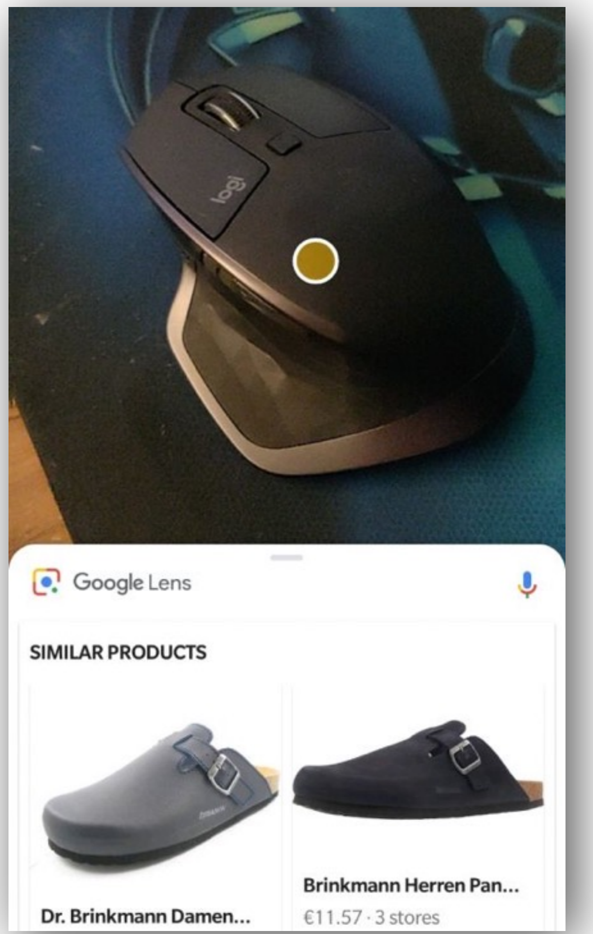
Machine learning works.

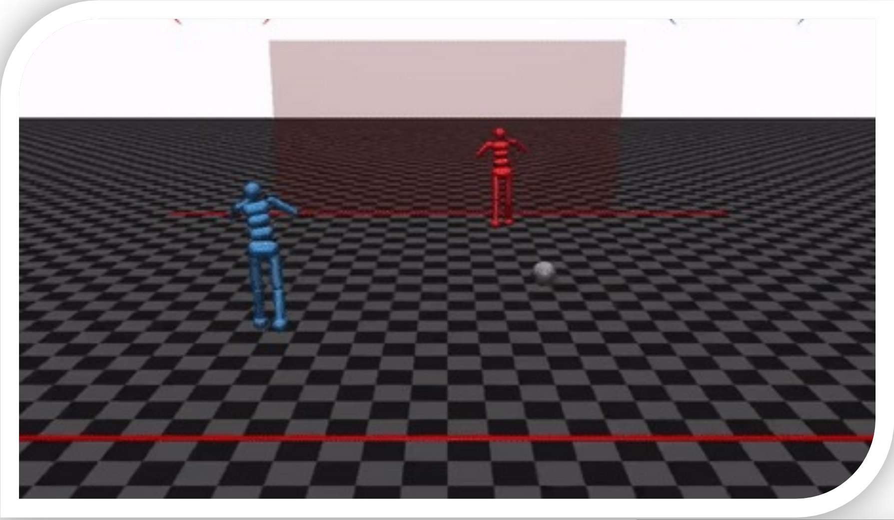


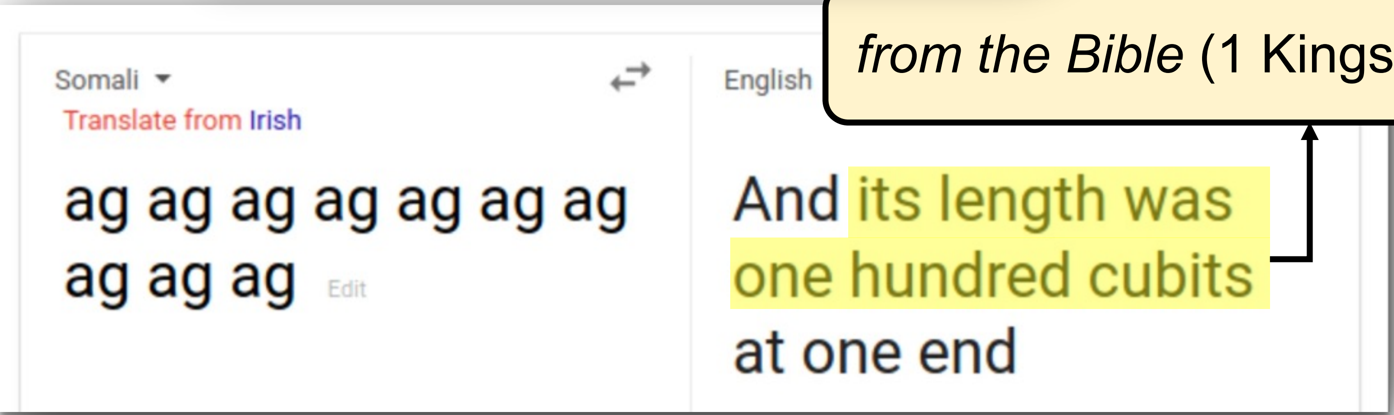
Google Translate

Machine learning works **most of the time!**

many applications tolerate occasional failures







from the Bible (1 Kings 7:2)

Machine learning can also fail disastrously.

Critical mistakes...

theguardian

Uber crash shows 'catastrophic failure' of self-driving technology, experts say

Machine learning can also fail disastrously.

Critical mistakes...

theguardian

Uber crash shows 'catastrophic failure' of self-driving technology, experts say

Direct attacks...

The New York Times

Microsoft Created a Twitter Bot to Learn From Users. It Quickly Became a Racist Jerk.

Machine learning can also fail disastrously.

Critical mistakes...

theguardian

Uber crash shows 'catastrophic failure' of self-driving technology, experts say

Direct attacks...

The New York Times

Microsoft Created a Twitter Bot to Learn From Users. It Quickly Became a Racist Jerk.

Private data leaks...

Does GPT-2 Know Your Phone Number?

Eric Wallace, Florian Tramèr, Matthew Jagielski, and Ariel Herbert-Voss

Challenge: understand and improve the worst-case behavior of machine learning (ML)

Approach: I study ML from an adversarial perspective

- ➢ to improve robustness and privacy of ML in adversarial settings
- ➢ to build ML that is *better*

My work: measuring and enhancing ML security

Evaluations

Evading ML models (NeurIPS ‘20) (ACM CCS ‘19)
Influenced design changes in Adblock Plus

Extracting private data (IEEE S&P ‘21)

Defenses

Training private models (ICLR ‘21 *spotlight*)

Training robust models (NeurIPS ‘19 *spotlight*) (ICLR ‘18)

Deploying private models (ICLR ‘19 *oral*)

Foundations

Stealing ML models (USENIX ‘16)
Microsoft’s top 3 threats to AI systems

Threat models for evasion (ICML ‘20)

My work: measuring and enhancing ML security

Evaluations

Evading ML models (NeurIPS '20) (ACM CCS '19)
Influenced design changes in Adblock Plus
Extracting private data (IEEE S&P '21)

Defenses

Training private models (ICLR '21 *spotlight*)
Training robust models (NeurIPS '19 *spotlight*) (ICLR '18)
Deploying private models (ICLR '19 *oral*)

Foundations

Stealing ML models (USENIX '16)
Microsoft's top 3 threats to AI systems
Threat models for evasion (ICML '20)

My work: measuring and enhancing ML security

Evaluations

Evading ML models (NeurIPS ‘20) (ACM CCS ‘19)
Influenced design changes in Adblock Plus
Extracting private data (IEEE S&P ‘21)

Defenses

Training private models (ICLR ‘21 *spotlight*)
Training robust models (NeurIPS ‘19 *spotlight*) (ICLR ‘18)
Deploying private models (ICLR ‘19 *oral*)

Foundations

Stealing ML models (USENIX ‘16)
Microsoft’s top 3 threats to AI systems
Threat models for evasion (ICML ‘20)

My work: measuring and enhancing ML security

Evaluations

Evading ML models (NeurIPS ‘20) (ACM CCS ‘19)
Influenced design changes in Adblock Plus

Extracting private data (IEEE S&P ‘21)

Defenses

Training private models (ICLR ‘21 *spotlight*)

Training robust models (NeurIPS ‘19 *spotlight*) (ICLR ‘18)

Deploying private models (ICLR ‘19 *oral*)

Foundations

Stealing ML models (USENIX ‘16)
Microsoft’s top 3 threats to AI systems

Threat models for evasion (ICML ‘20)

My work: measuring and enhancing ML security

Evaluations

Evading ML models (NeurIPS ‘20) (ACM CCS ‘19)
Influenced design changes in Adblock Plus

Extracting private data (IEEE S&P ‘21)

Defenses

Training private models (ICLR ‘21 *spotlight*)

Training robust models (NeurIPS ‘19 *spotlight*) (ICLR ‘18)

Deploying private models (ICLR ‘19 *oral*)

Foundations

Stealing ML models (USENIX ‘16)
Microsoft’s top 3 threats to AI systems

Threat models for evasion (ICML ‘20)

My work: measuring and enhancing ML security

Evaluations

Evading ML models (NeurIPS '20) (ACM CCS '19)
Influenced design changes in Adblock Plus

Extracting private data (IEEE S&P '21)

Defenses

Training private models (ICLR '21 *spotlight*)

Training robust models (NeurIPS '19 *spotlight*) (ICLR '18)

Deploying private models (ICLR '19 *oral*)

Foundations

Stealing ML models (USENIX '16)
Microsoft's top 3 threats to AI systems

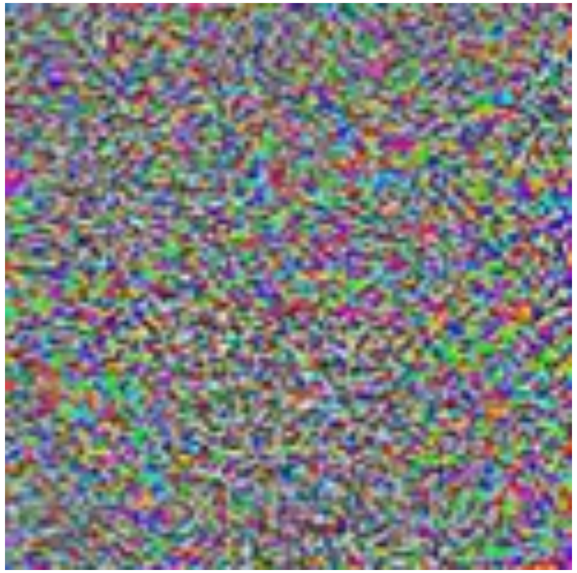
Threat models for evasion (ICML '20)

Adversarial examples: a curious *bug* in ML

[Szegedy et al. ‘13], [Biggio et al. ‘13], [Goodfellow et al. ‘14], ...

88% Tabby Cat

+



Adversarial noise

=

100% Guacamole

We must treat adversarial examples as a computer security problem.

In our threat analysis.

Identify ***deployed systems*** where adversarial examples can cause ***harms beyond misclassification***

In our defense evaluations.

Evaluate robustness against ***adaptive adversaries***

We must treat adversarial examples as a computer security problem.

In our threat analysis.

Identify ***deployed systems*** where adversarial examples can cause ***harms beyond misclassification***

In our defense evaluations.

Evaluate robustness against ***adaptive adversaries***

We must treat adversarial examples as a computer security problem.

In our threat analysis.

Identify ***deployed systems*** where adversarial examples can cause ***harms beyond misclassification***

In our defense evaluations.

Evaluate robustness against ***adaptive adversaries***

We must treat adversarial examples as a computer security problem.

In our threat analysis.

T, Dupré, Rusak, Pellegrino, Boneh (ACM CCS 2019)

- adversarial examples are the perfect tool to attack *online content blockers*
- *using ML for ad-blocking can break Web security*
- *this work led to design changes in Adblock Plus*

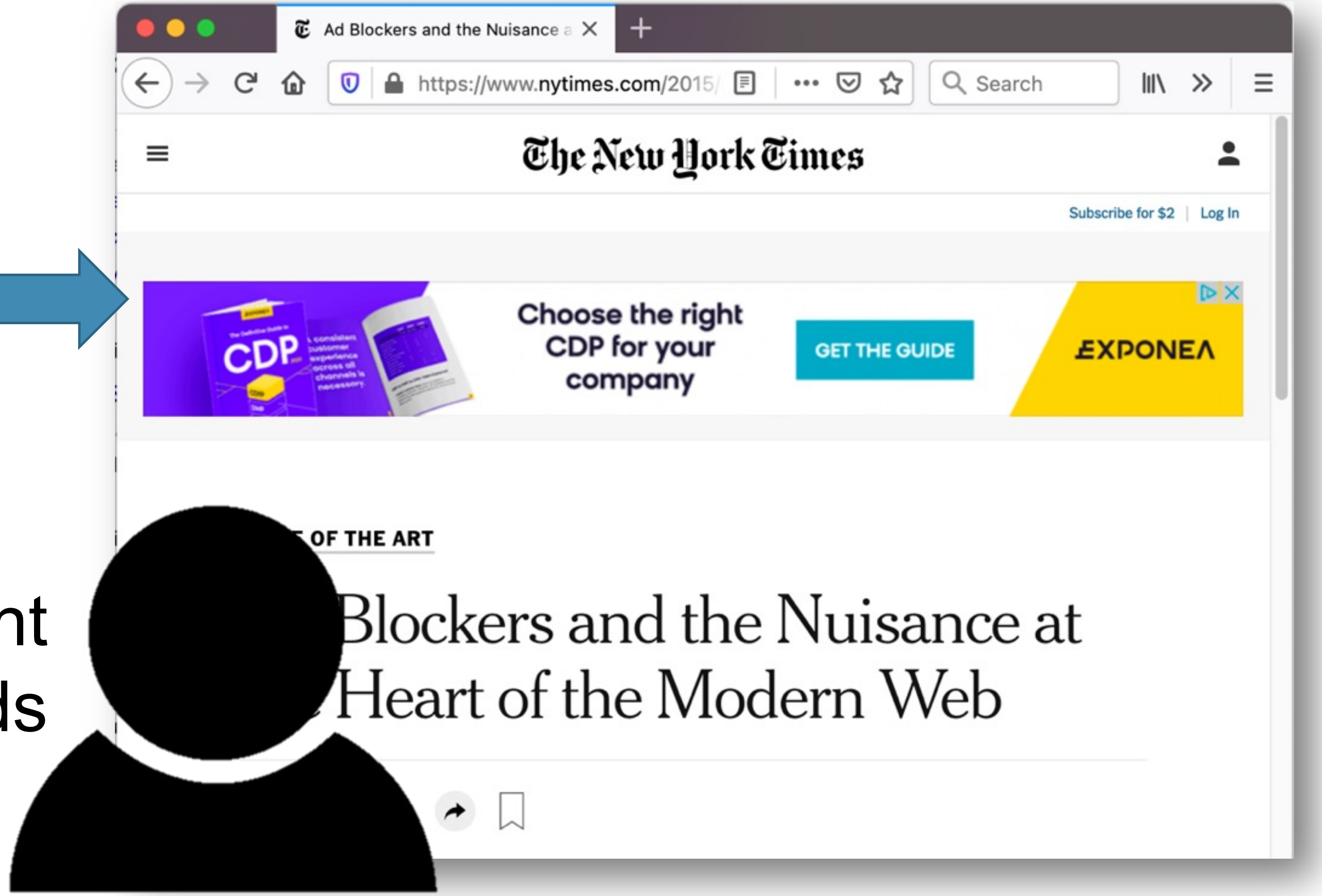
ABP Adblock Plus

100M active users

Adversarial examples are a security threat.

example: online ad-blocking

publishers & advertisers want to show ads to users...



...users don't want to see ads

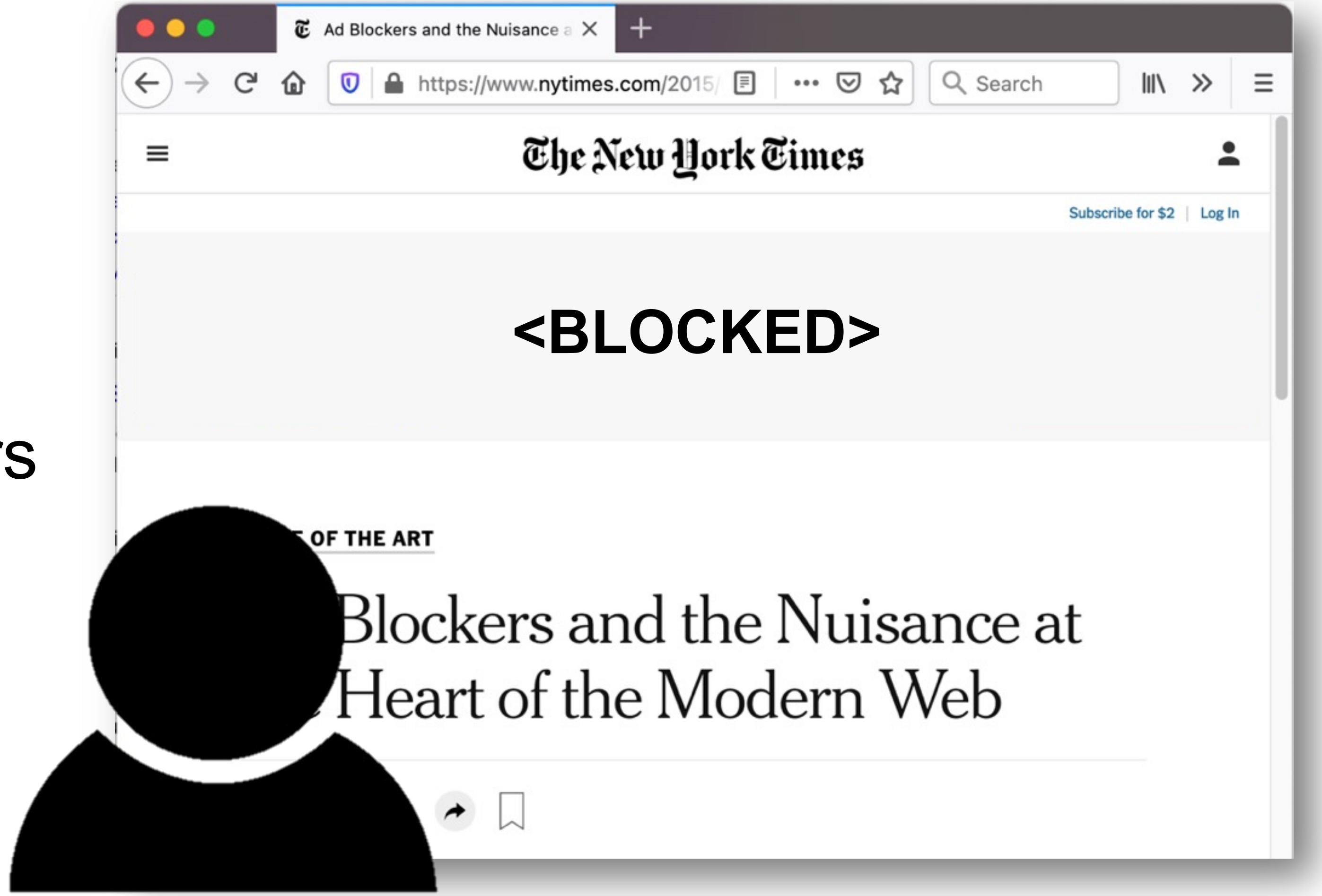
Adversarial examples are a security threat.

example: online ad-blocking

ABP Adblock Plus

brave

users install ad-blockers to remove ads...

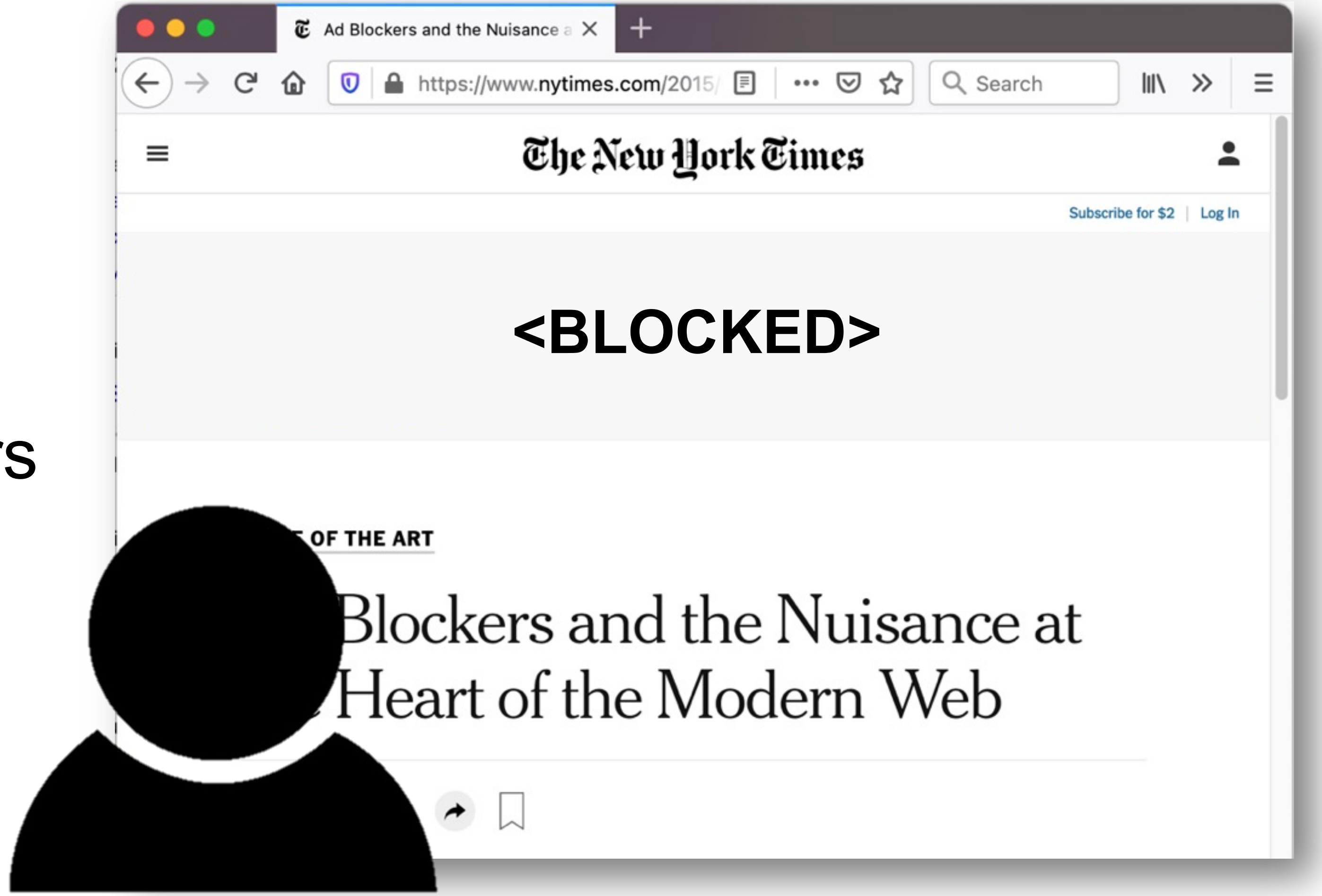


Adversarial examples are a security threat.

example: online ad-blocking

users install ad-blockers to remove ads...

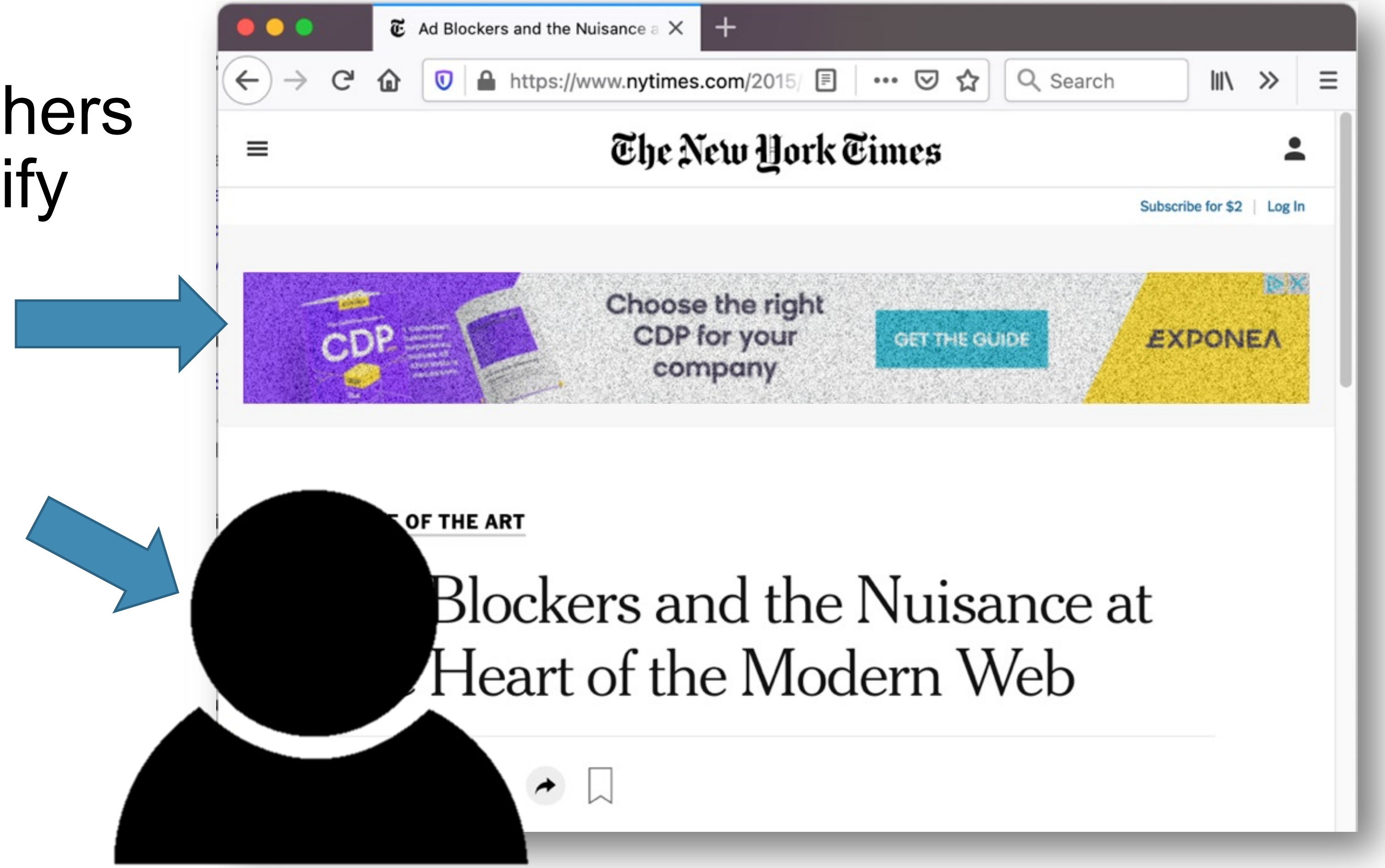
...using machine learning!



An attacker can use adversarial examples to evade content blocking.

adversaries (publishers & advertisers) modify content to evade blocking...

...without changing the user's visual perception of ads



For now, the adversary wins!


MOTHERBOARD
TECH BY VICE

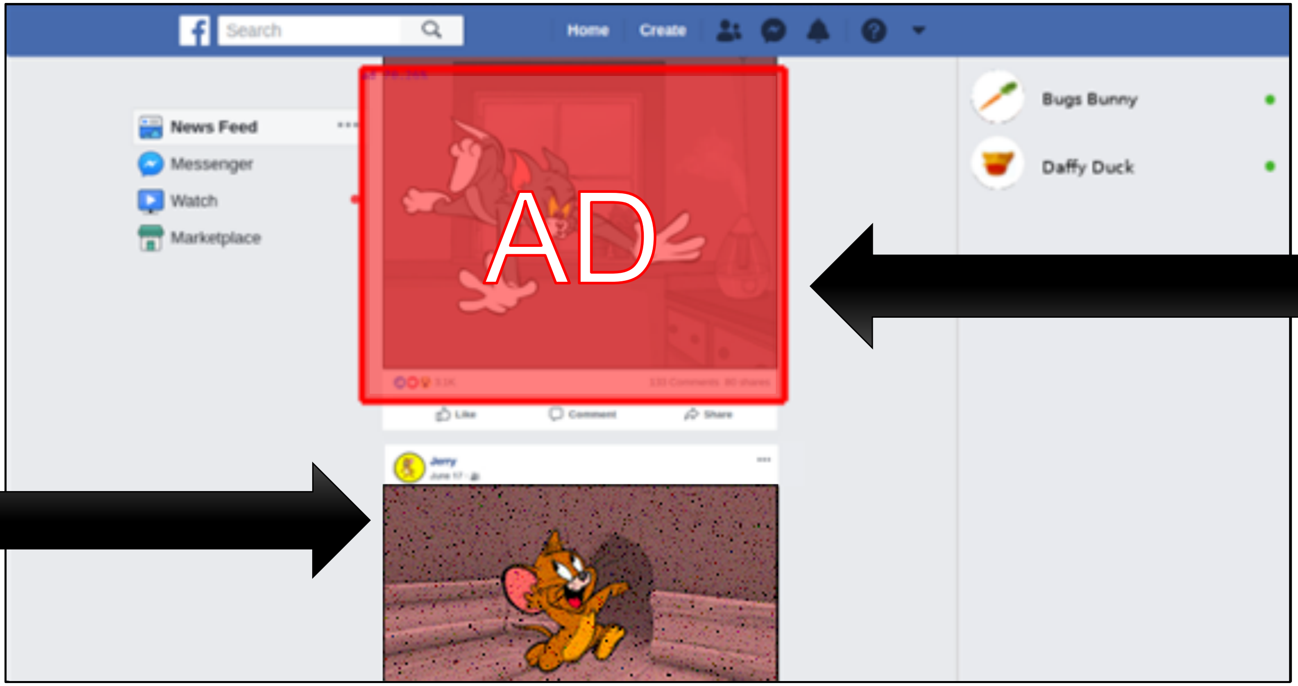
Researchers Defeat Most Powerful Ad Blockers, Declare a 'New Arms Race'

"*AdVersarial: Perceptual Ad Blocking meets Adversarial Machine Learning*", ACM CCS 2019

Adversarial examples can cause harm beyond model evasion.

Adblock Plus wants to run a ML model on *screenshots* of your entire Facebook feed.

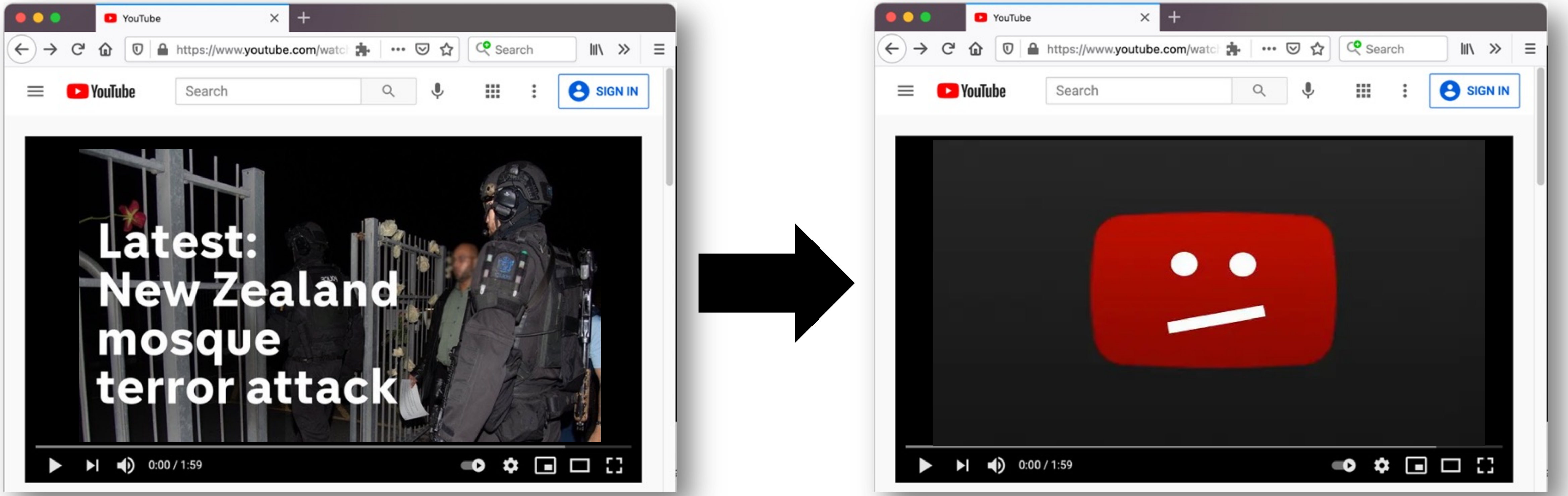
Jerry uploads malicious content …



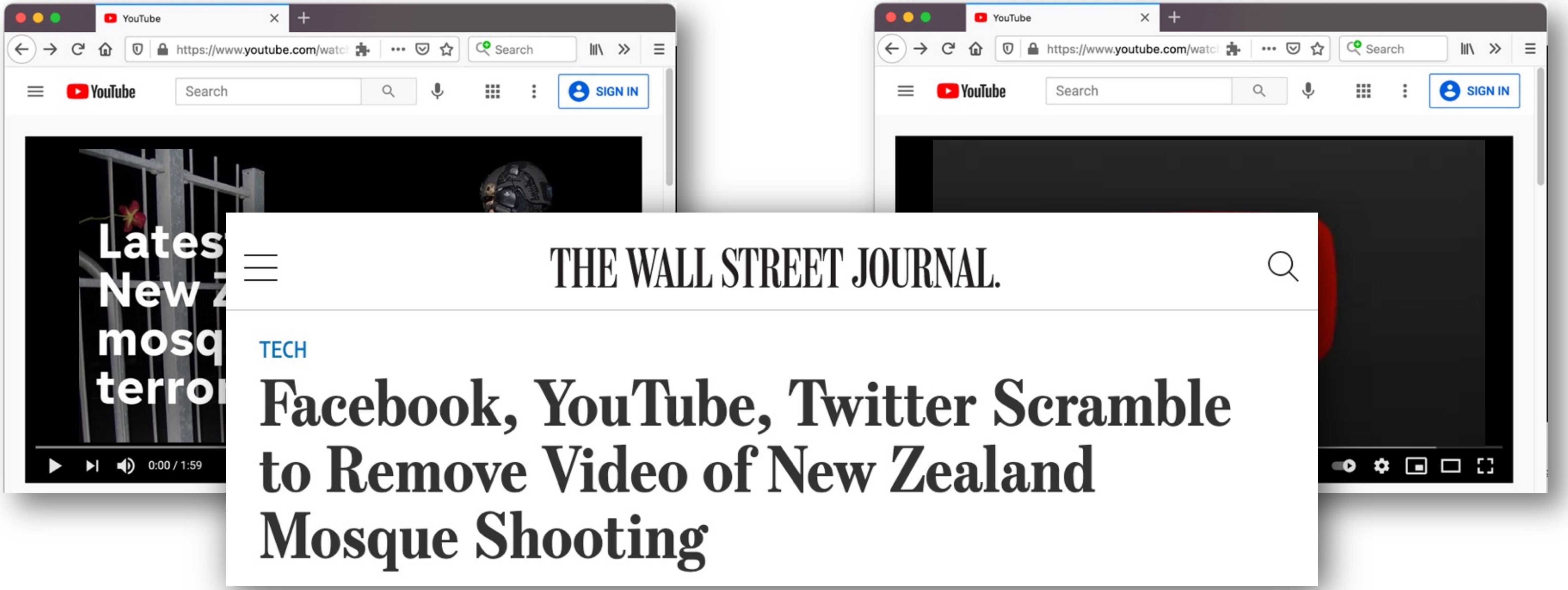
… so that Tom's post gets blocked

"*AdVersarial: Perceptual Ad Blocking meets Adversarial Machine Learning*", ACM CCS 2019

Adversarial examples are a security threat.
example: blocking undesired content



Adversarial examples are a security threat.
example: blocking undesired content



We must treat adversarial examples as a computer security problem.

In our threat analysis.

Identify ***deployed systems*** where adversarial examples can cause ***harms beyond misclassification***

In our defense evaluations.

Evaluate robustness against ***adaptive adversaries***

We must treat adversarial examples as a computer security problem.

In our defense evaluations.

T, Carlini, Brendel, Madry (NeurIPS 2020)

- ➢ empirical study of 13 peer-reviewed defenses (from NeurIPS, ICML, ICLR)
- ➢ evaluations are *overly complex*. *Simpler* attacks break each defense!
- ➢ *new crypto-inspired attack:* **feature collisions**

A formal model for evaluating robustness.

- Train a model $f(\cdot)$ on a distribution $\mathfrak{D}$ of labelled inputs (x, y)
- The adversary *perturbs* <u>test</u> inputs x sampled from $\mathfrak{D}$ with noise δ

Which perturbations δ do we allow?

- Ideal: any "semantically small" perturbation

ambiguous, hard to formalize

A formal model for evaluating robustness.

- Train a model $f(\cdot)$ on a distribution $\mathfrak{D}$ of labelled inputs (x, y)
- The adversary *perturbs* test inputs x sampled from $\mathfrak{D}$ with noise δ

Which perturbations $\boldsymbol{\delta}$ do we allow?

- Ideal: any "semantically small" perturbation
- Relaxation: perturbations δ from a ***fixed*** set S

ambiguous, hard to formalize

Example: $S = \{\delta : \|\delta\|_2 \leq \varepsilon\}$

necessary but not sufficient

A formal model for evaluating robustness.

- Train a model $f(\cdot)$ on a distribution $\mathfrak{D}$ of labelled inputs (x, y)
- The adversary *perturbs* test inputs x sampled from $\mathfrak{D}$ with noise δ

Which perturbations δ do we allow?

- Ideal: any "semantically small" perturbation
- Relaxation: perturbations δ from a ***fixed*** set S

Example: $S = \{\delta : \|\delta\|_2 \leq \varepsilon\}$

Ultimate goal:

- discover defensive techniques that *generalize* across perturbation sets
- learn *something new* about ML

A formal model for evaluating robustness.

- Train a model $f(\cdot)$ on a distribution $\mathfrak{D}$ of labelled inputs (x, y)
- The adversary *perturbs* test inputs x sampled from $\mathfrak{D}$ with noise δ

Which perturbations δ do we allow?

- Ideal: any "semantically small" perturbation
- Relaxation: perturbations δ from a ***fixed*** set S

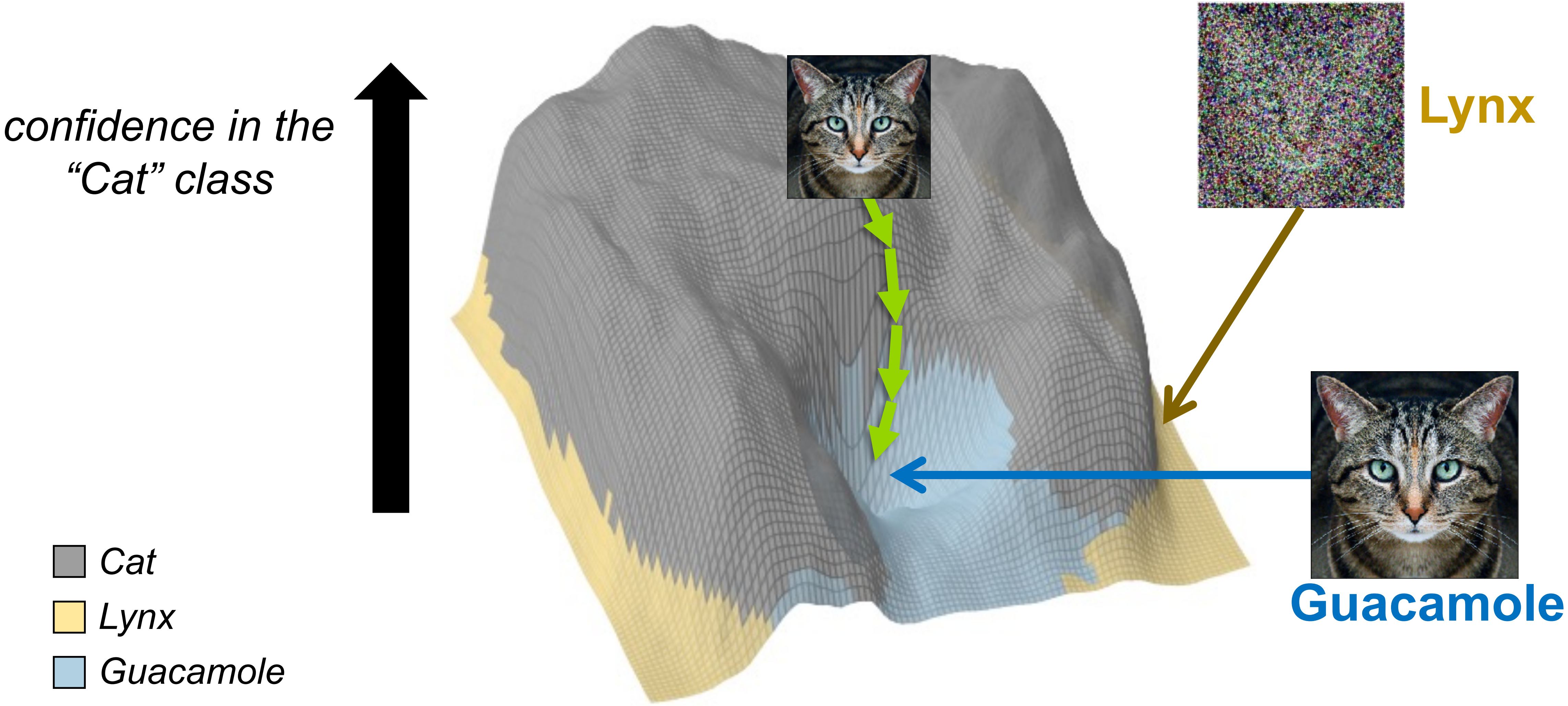
Example: $S = \{\delta: \|\delta\|_2 \leq \varepsilon\}$

evaluating robustness is an *optimization problem*

for an input (x, y), find $\delta \in S$ that minimizes $f(x + \delta)_y$

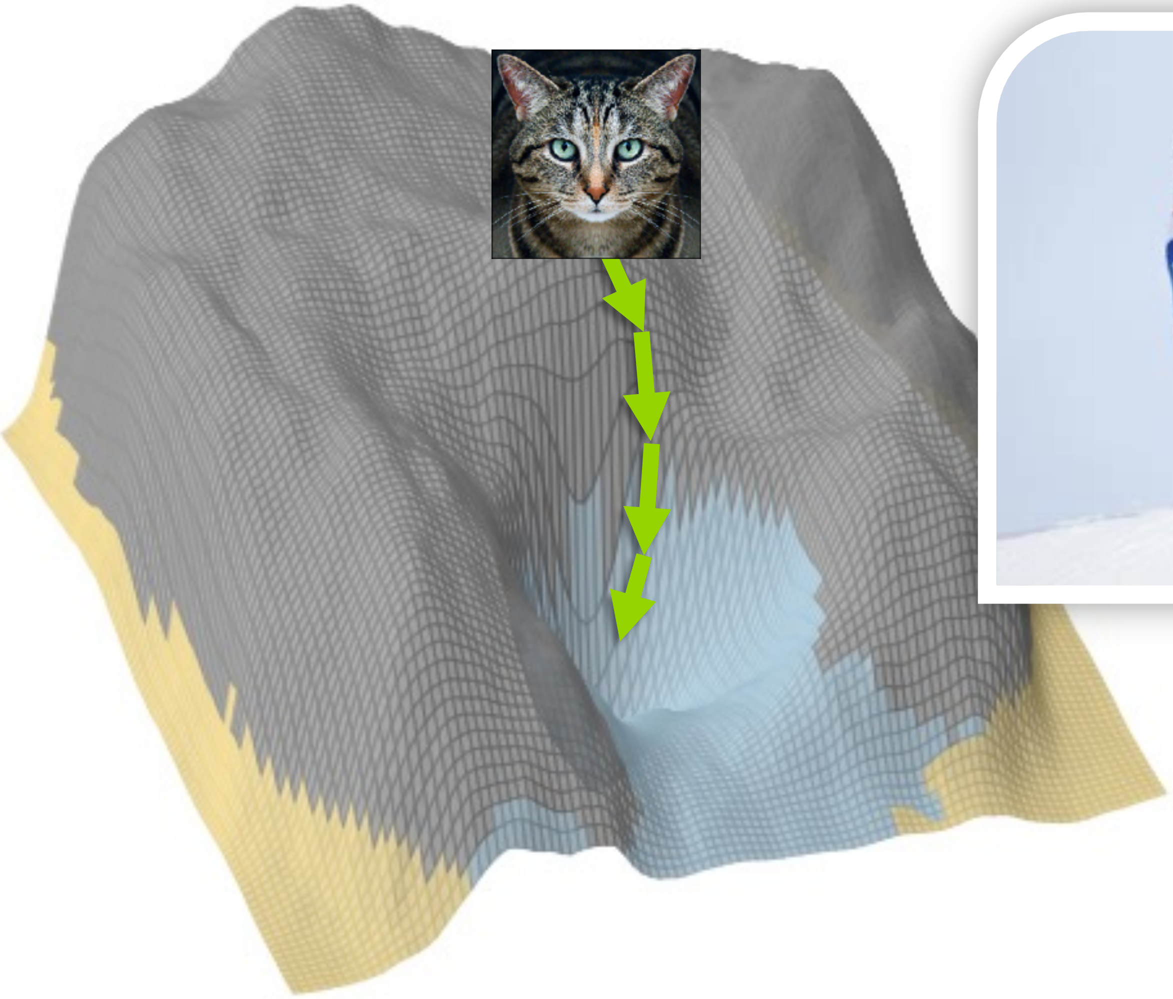
model's confidence in class y

Adversarial examples can be found with gradient descent.



Adversarial examples can be found with gradient descent.

confidence in the "Cat" class

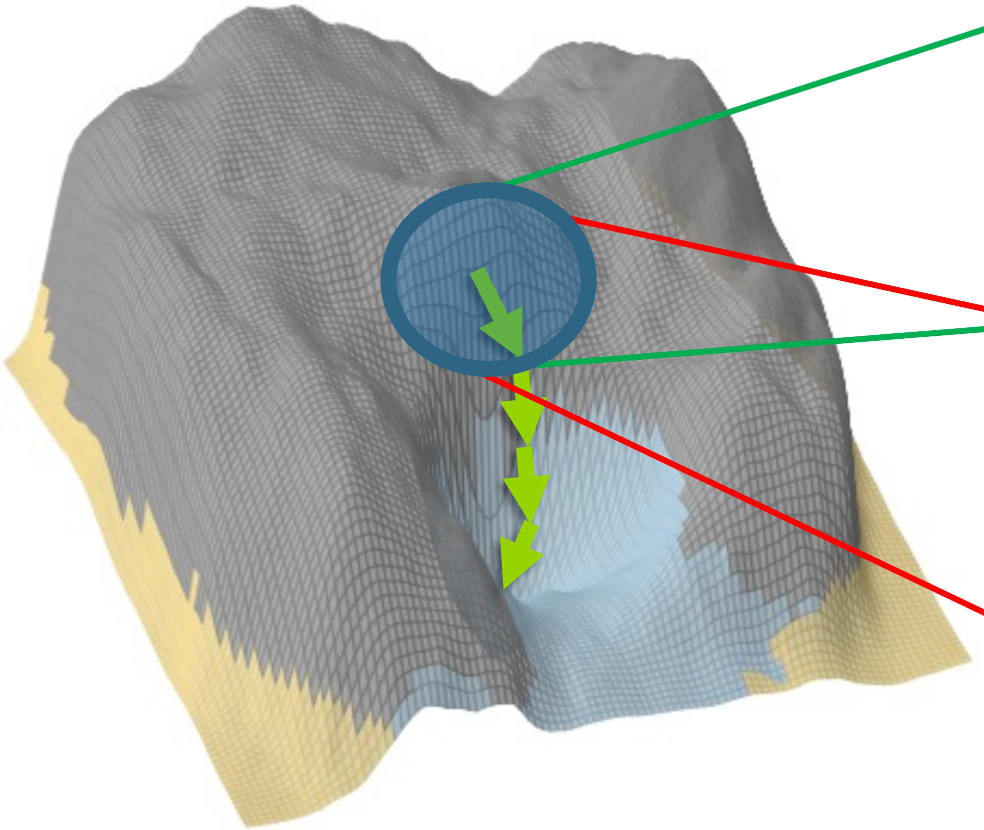


- *Cat*
- *Lynx*
- *Guacamole*

Many defenses *break* gradient descent.

T et al. (ICLR 2018): *defenses can break function smoothness*

Other causes of masked gradients:
- *numerical instability:* [Papernot et al. ‘17], [Carlini & Wagner ‘17]
- *stochasticity:* [Athalye et al. ‘18]



for most ML models, the optimization problem is *easy* (the function is *smooth*)

many defenses against adversarial examples break the smoothness of the function

this doesn’t make the model more robust!

Strong robustness evaluations are *adaptive*.

the optimization strategy is *tailored* to the defense

[Carlini & Wagner ‘17], [Athalye et al. ‘18], [T et al. ‘20]

defense 1

defense 2

defense 3

Strong robustness evaluations are *adaptive*.

the optimization strategy is *tailored* to the defense

[Carlini & Wagner '17], [Athalye et al. '18], [T et al. '20]

defense 1

defense 2

defense 3

Strong robustness evaluations are *adaptive*.

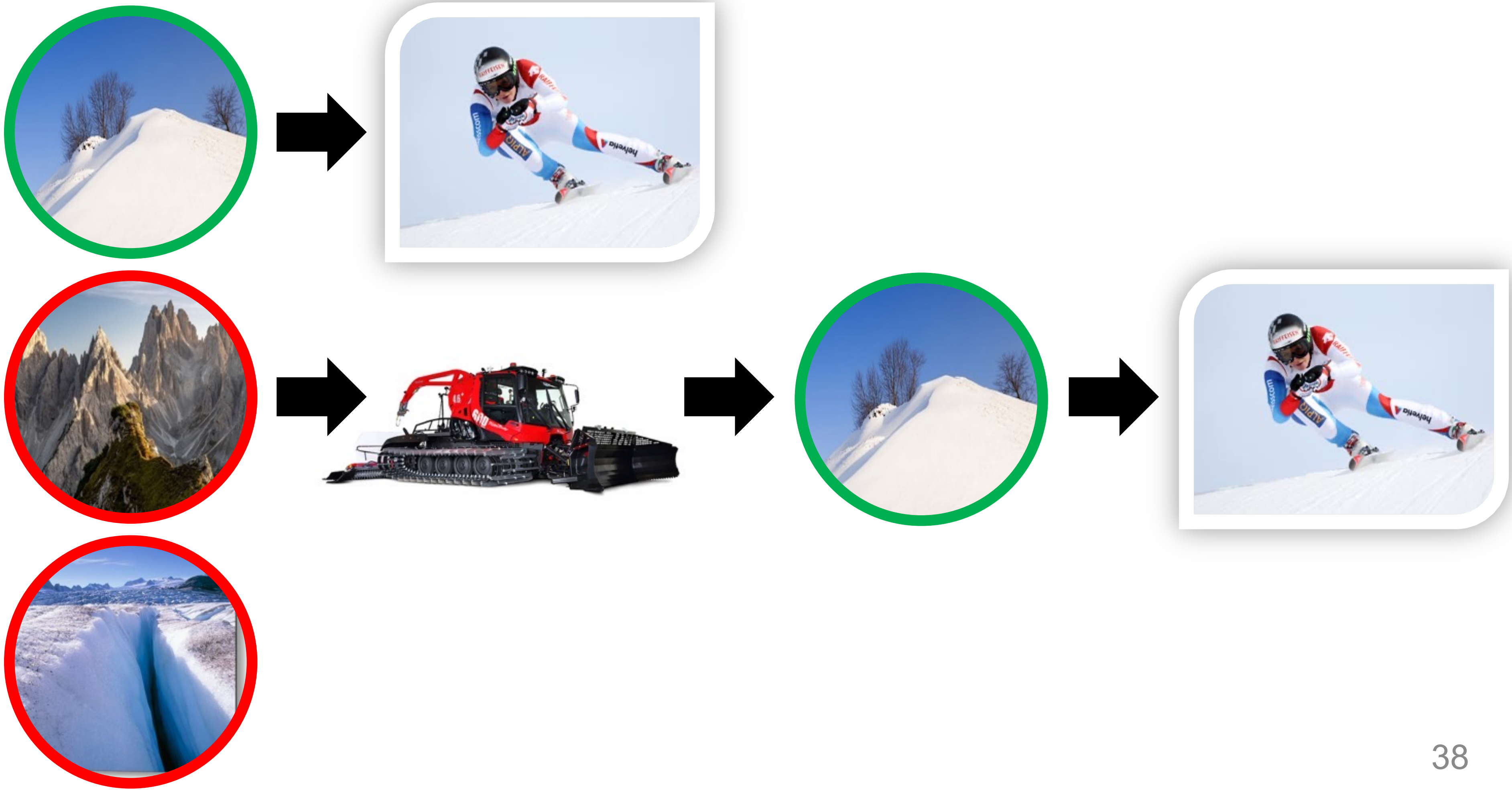
the optimization strategy is *tailored* to the defense

[Carlini & Wagner ‘17], [Athalye et al. ‘18], [T et al. ‘20]

defense 1

defense 2

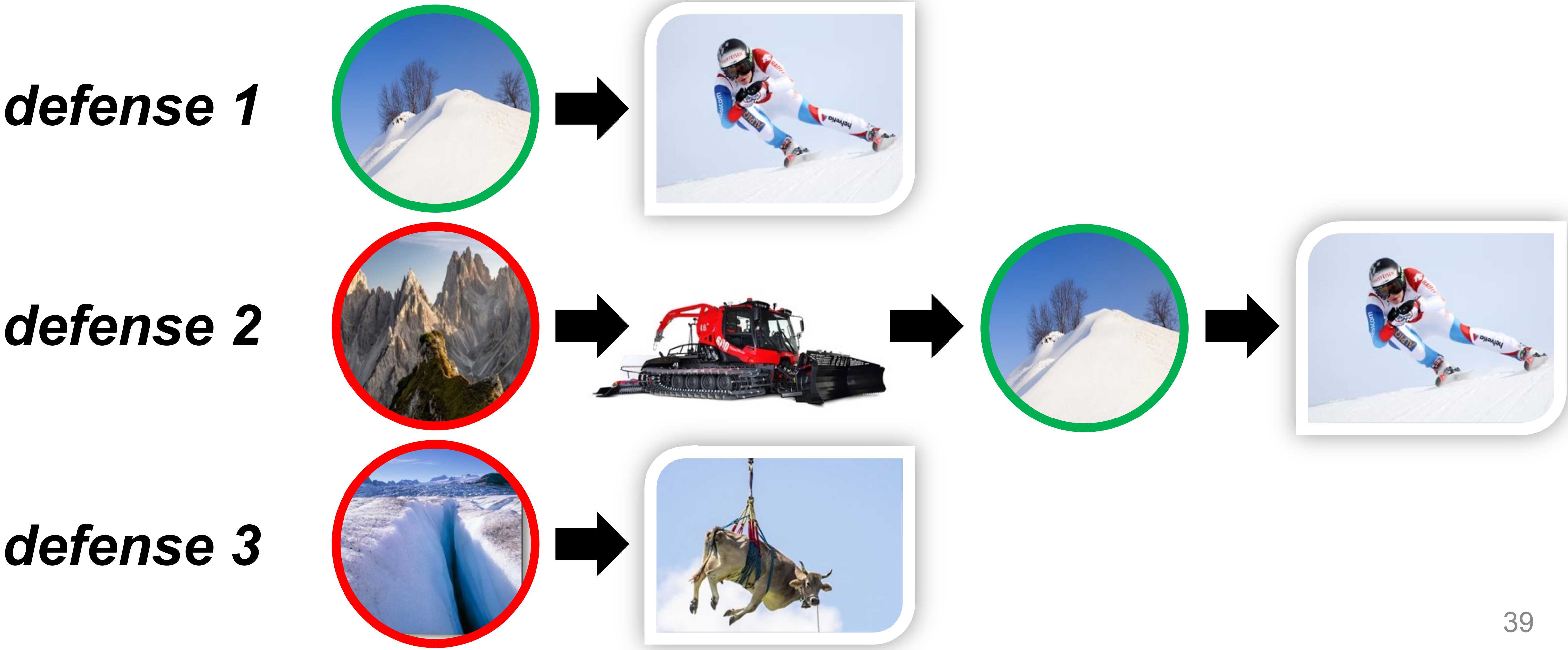
defense 3



Strong robustness evaluations are *adaptive*.

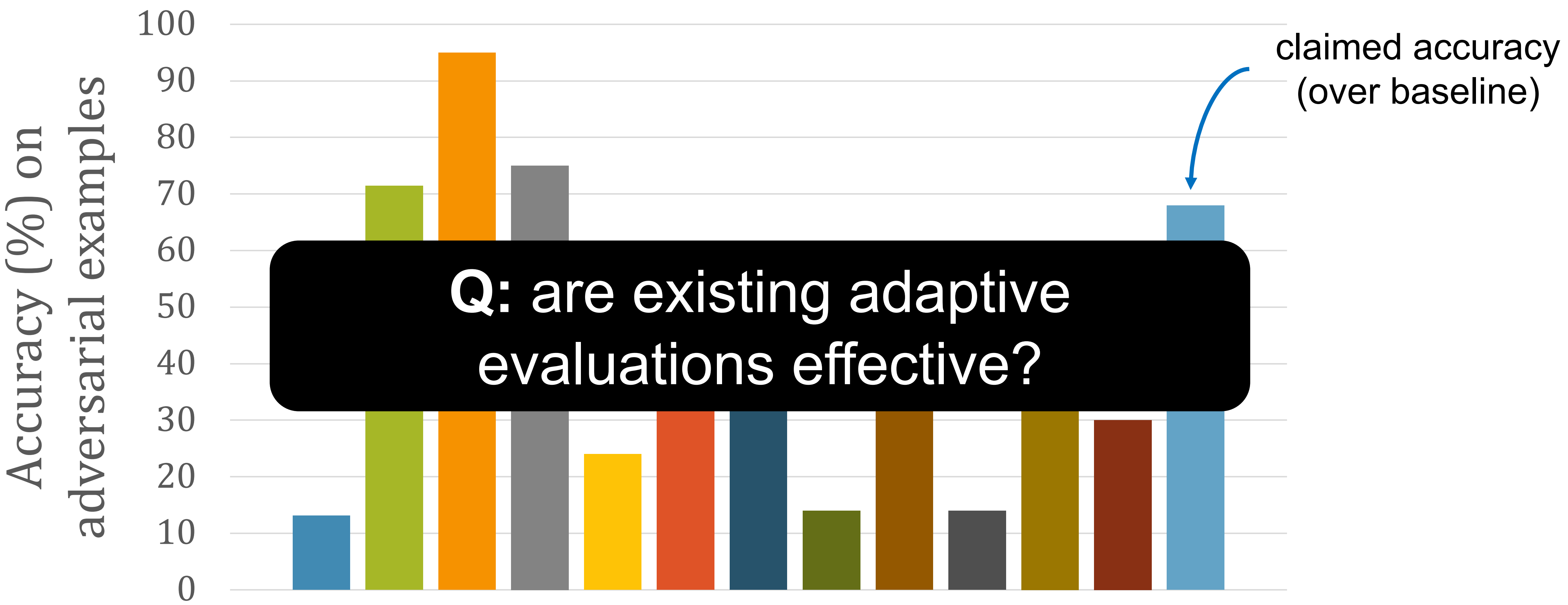
the optimization strategy is *tailored* to the defense

[Carlini & Wagner ‘17], [Athalye et al. ‘18], [T et al. ‘20]



Defenses *try* adaptive evaluations.

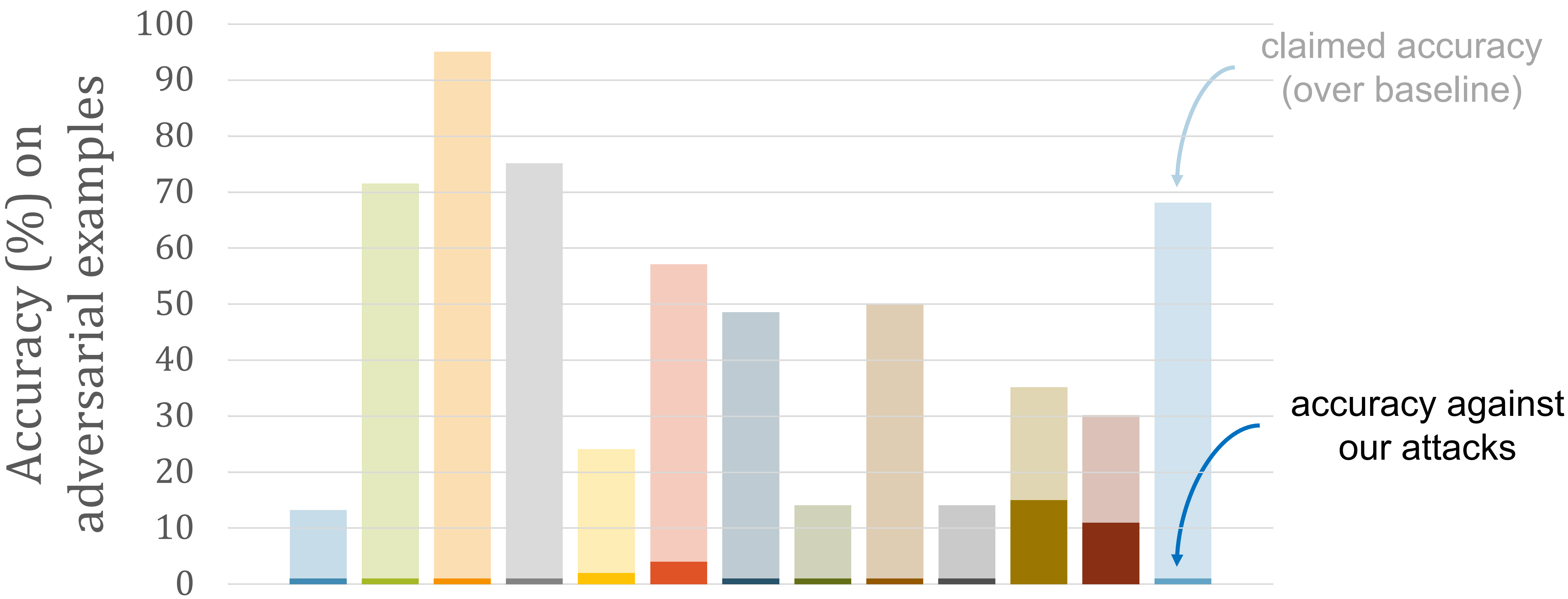
T, Carlini, Brendel, Mądry (NeurIPS 2020): **evaluation of 13 defenses**



Q: are existing adaptive evaluations effective?

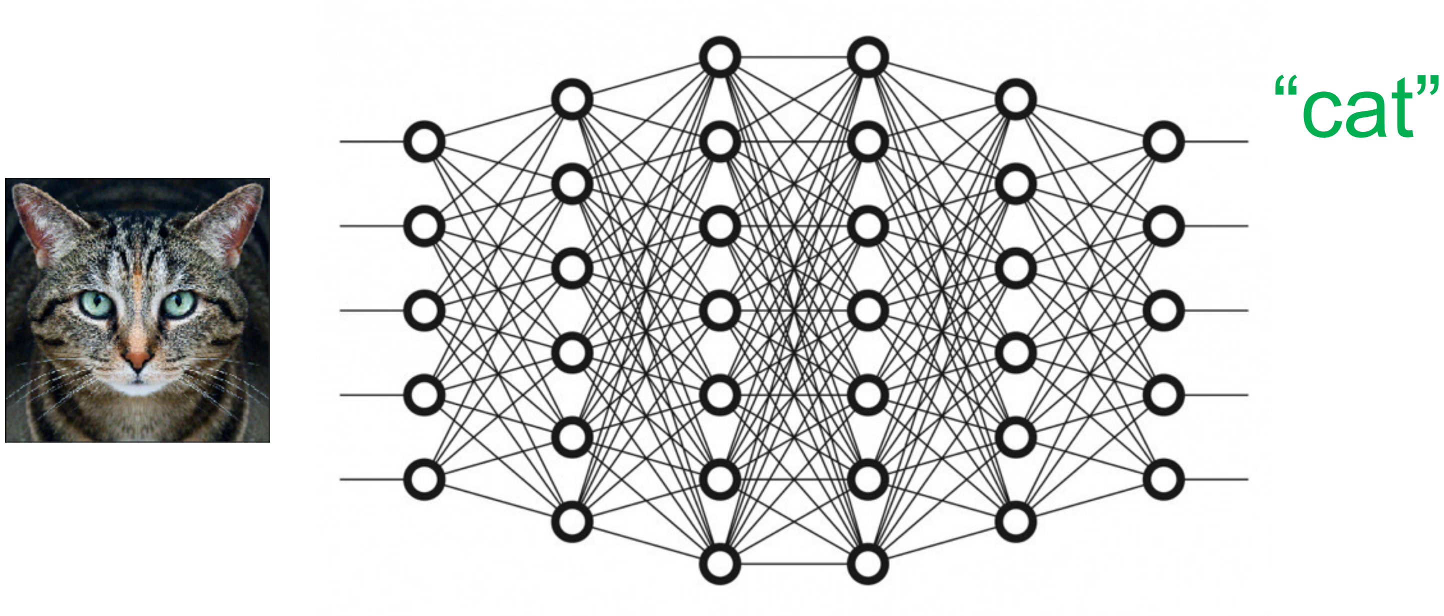
All defenses *over-estimate* robustness.

T, Carlini, Brendel, Mądry (NeurIPS 2020): **evaluation of 13 defenses**



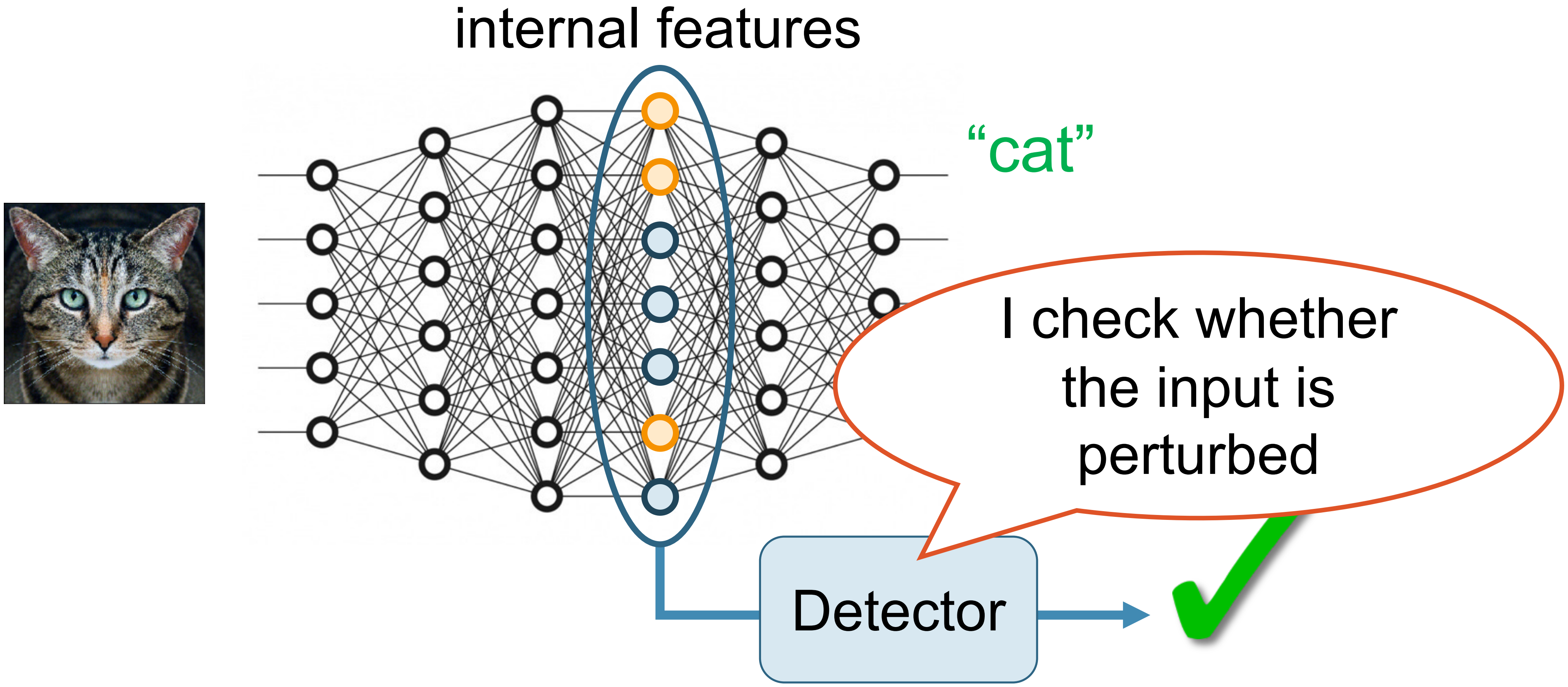
Building stronger adaptive attacks.

our target: adversarial examples ***detectors***



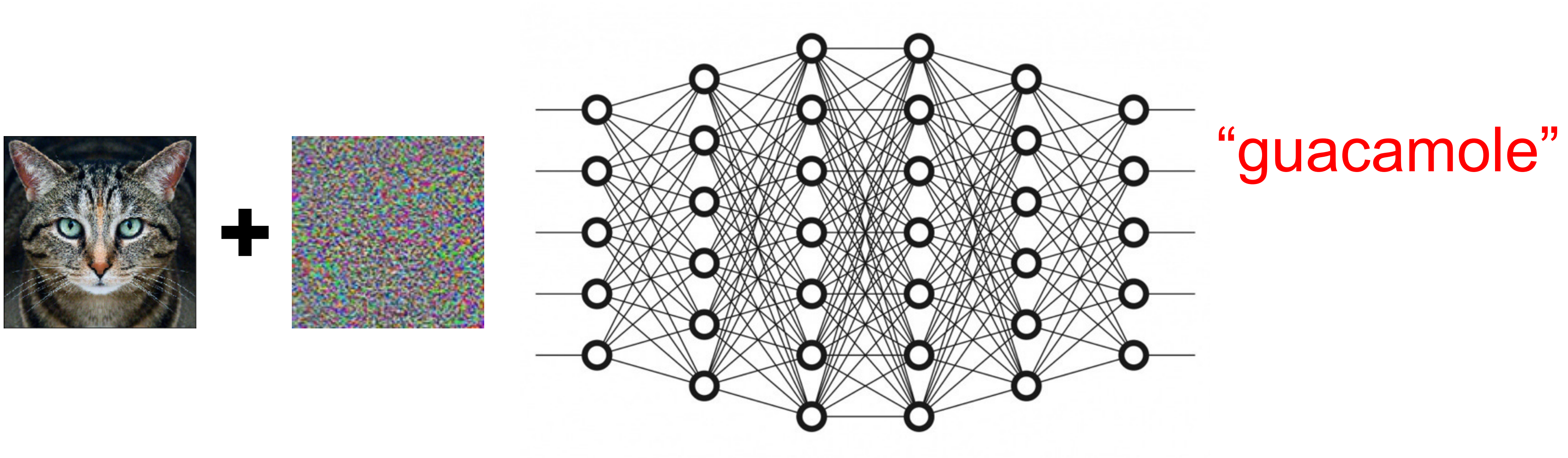
Building stronger adaptive attacks.

our target: adversarial examples ***detectors***



Building stronger adaptive attacks.

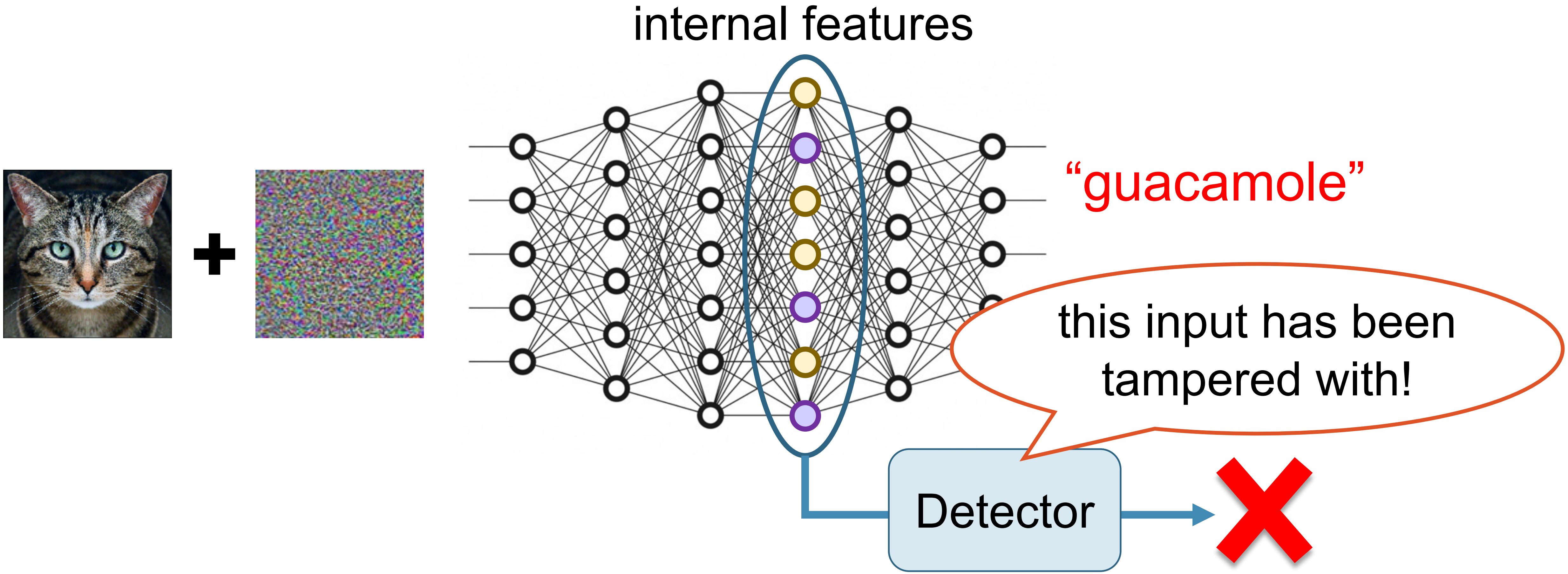
our target: adversarial examples ***detectors***



"*On Adaptive Attacks to Adversarial Example Defenses*", NeurIPS 2020

Building stronger adaptive attacks.

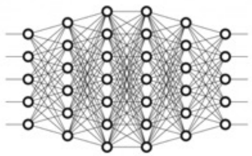
our target: adversarial examples ***detectors***



An overly *complex* adaptive attack.

minimize $\underbrace{f(x+\delta)}_{}{}_y + \lambda \cdot \underbrace{g(x+\delta)}_{\text{Detector}}$

such that $\delta \in S$

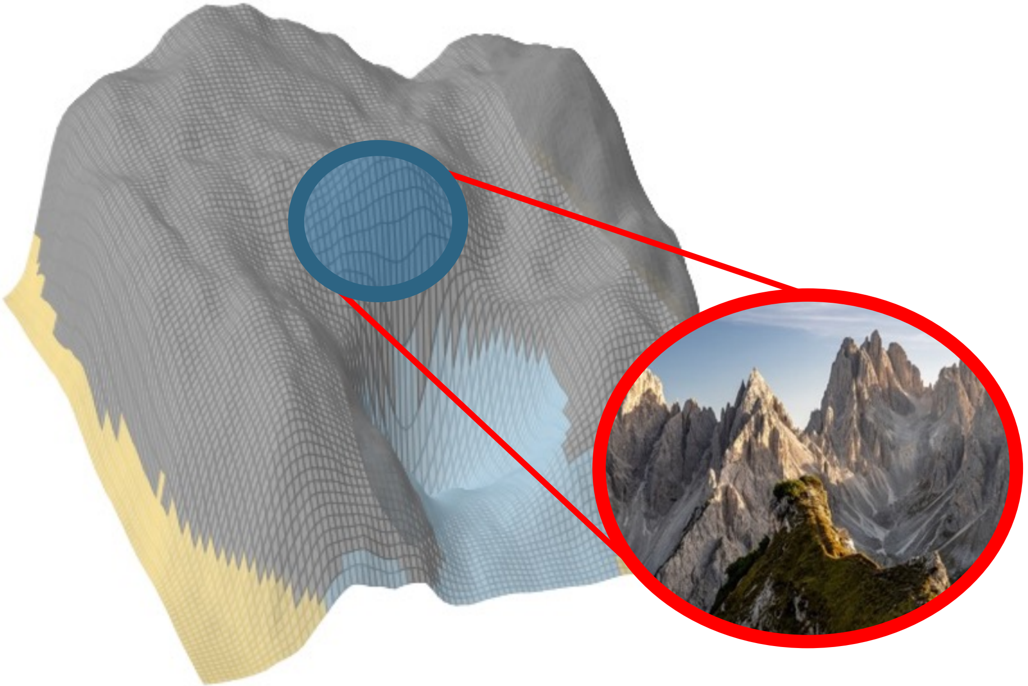


confidence that input is invalid

An overly *complex* adaptive attack.

minimize $\quad f(x+\delta)_y \quad + \; \lambda \cdot \underbrace{g(x+\delta)}_{\text{Detector}}$

such that $\delta \in S$



Issue: detectors are often

- *stochastic*
- *discontinuous*
- *numerically unstable*

"*On Adaptive Attacks to Adversarial Example Defenses*", NeurIPS 2020

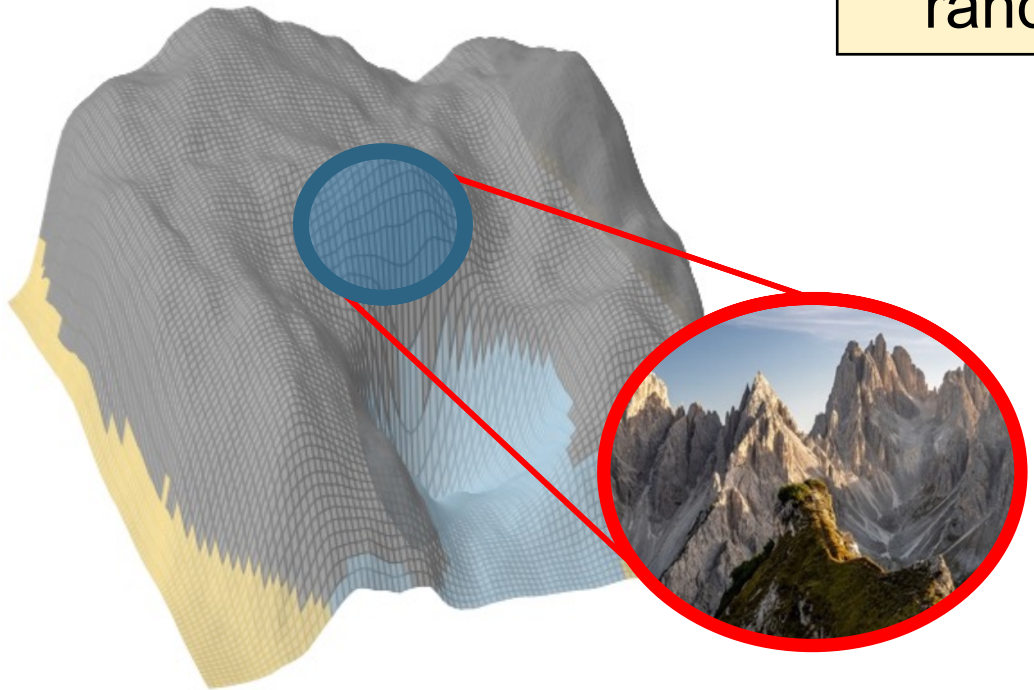
An overly *complex* adaptive attack.

minimize $\quad f(x+\delta)_y \quad + \ \lambda \ \cdot \ g(x+\delta)$

such that $\delta \in S$

take expectation over randomness of g...

replace g by smooth approximation $\hat{g}$...



Issue: detectors are often

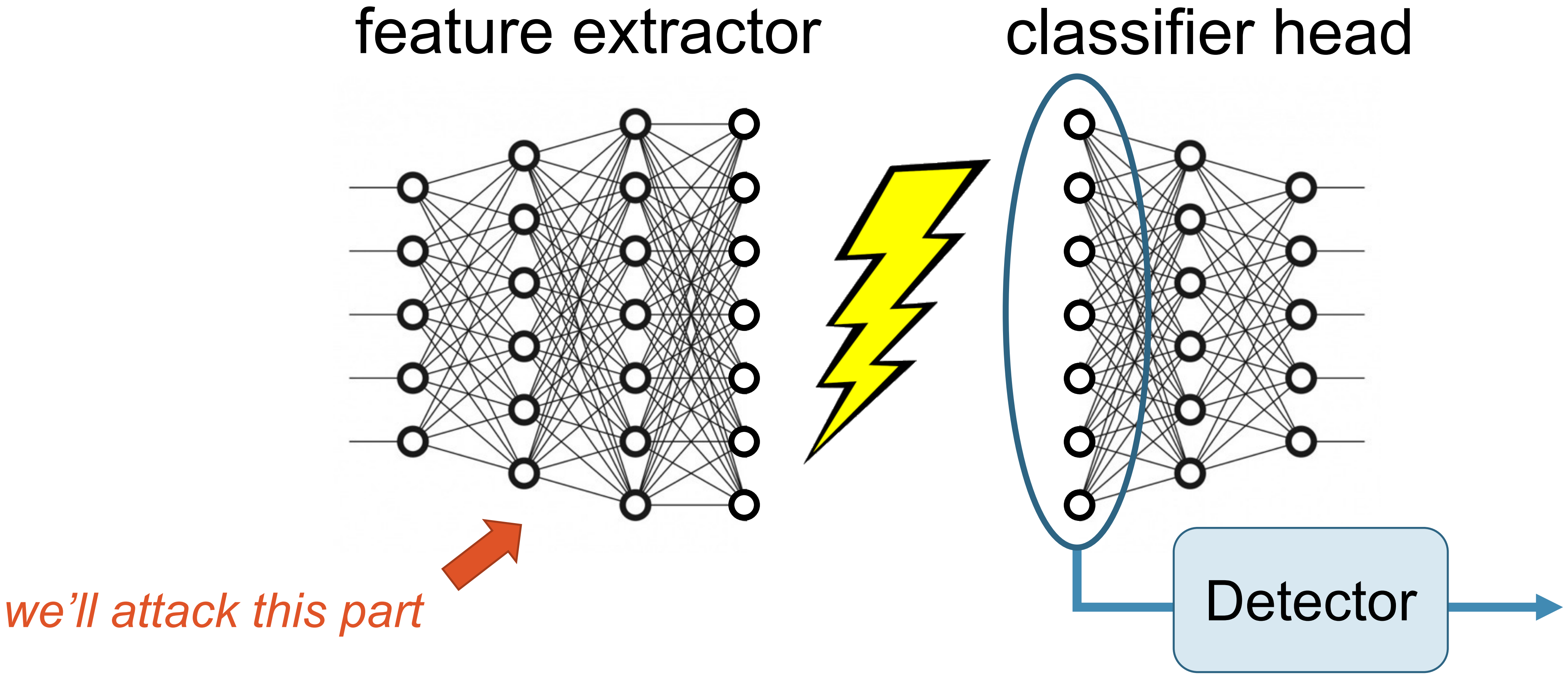
- *stochastic*
- *discontinuous*
- *numerically unstable*

"*On Adaptive Attacks to Adversarial Example Defenses*", NeurIPS 2020

A *simpler* & *stronger* attack: feature collisions.

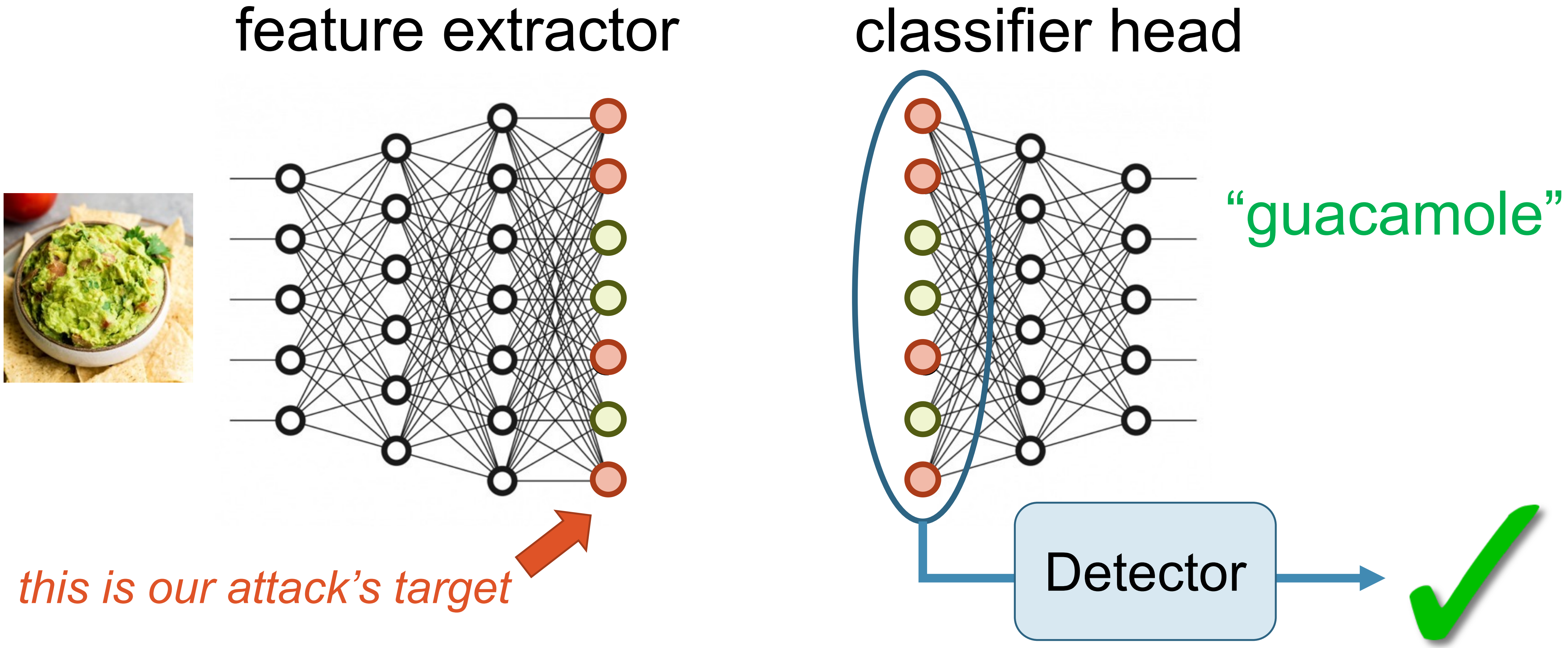
A *simpler* & *stronger* attack: feature collisions.

insight #1: decompose the system



A *simpler* & *stronger* attack: feature collisions.

insight #2: target a natural input



“*On Adaptive Attacks to Adversarial Example Defenses*”, NeurIPS 2020

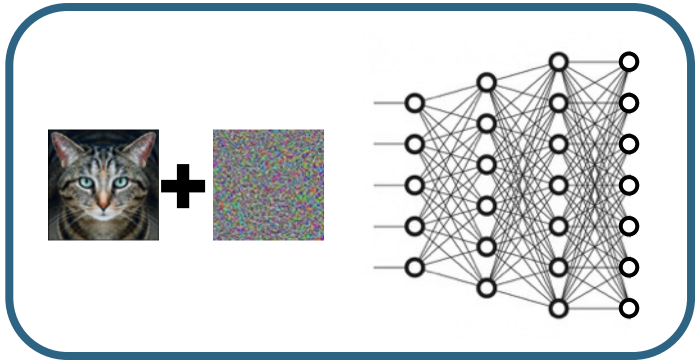
Goal: collide with features of the target input.

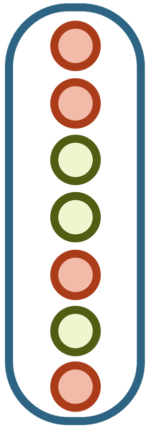
[Sabour et al. ‘15]

minimize

such that $\color{red}{\delta} \in \color{red}{S}$

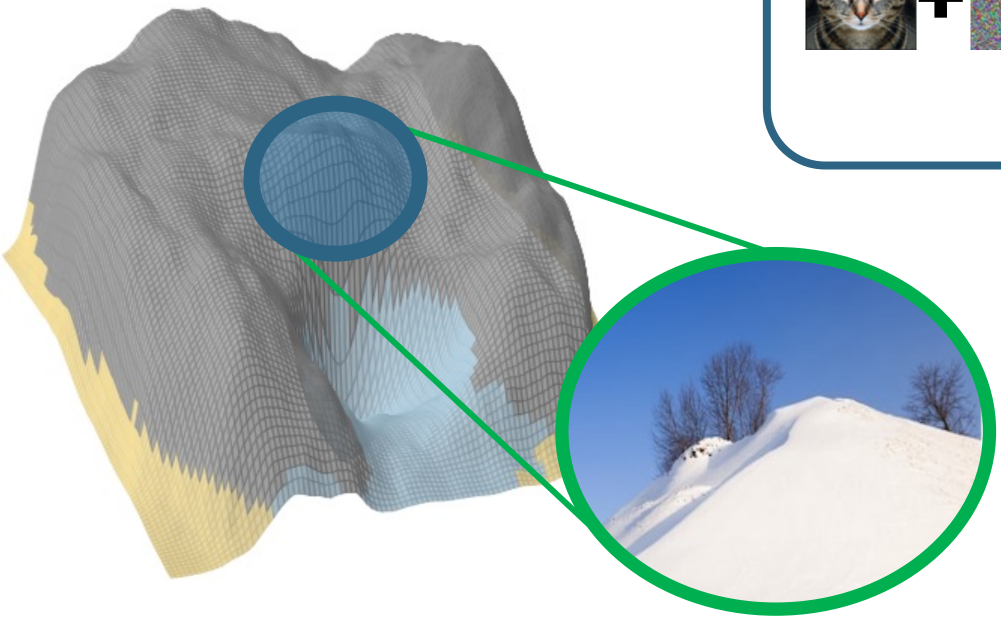
$$\|\color{blue}{f_{\text{extractor}}}(\color{green}{x} + \color{red}{\delta}) - \color{green}{z}\|_2$$





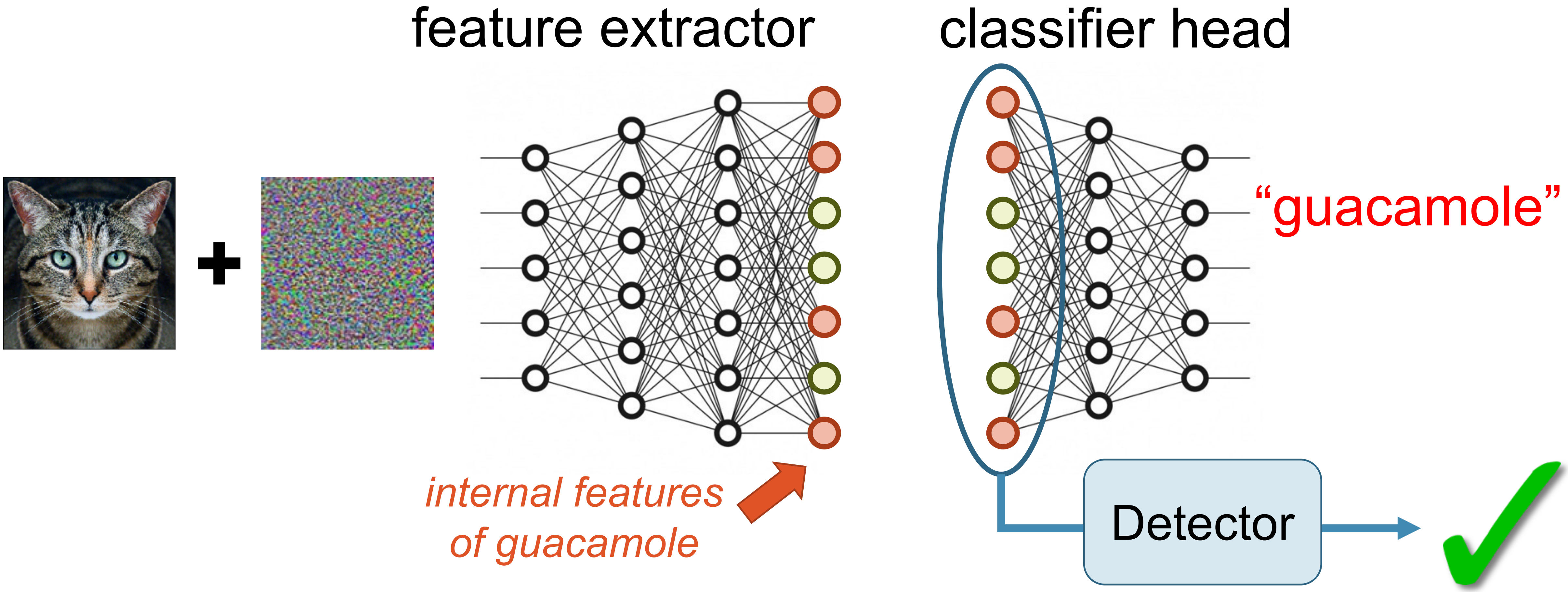
internal features of guacamole

feature extractors are not collision resistant!

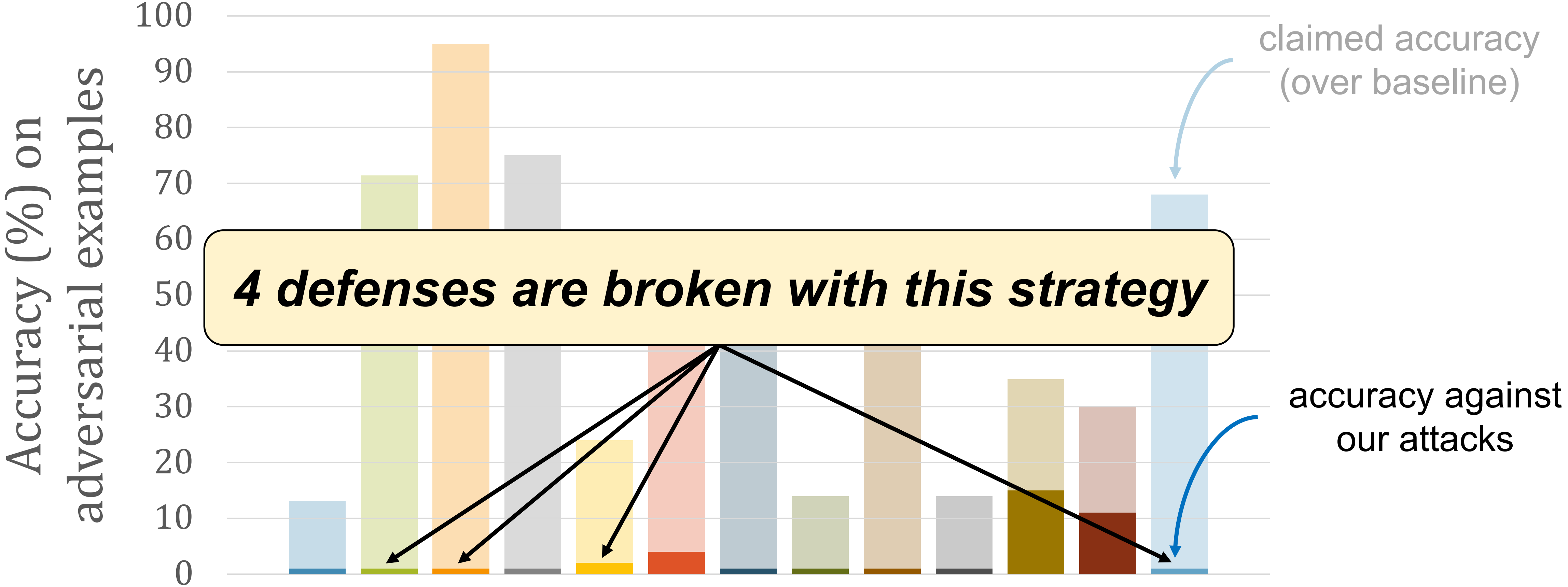


The **feature collision attack.**

or “garbage-in, garbage-out”



Feature collision is a **strong** adaptive attack.



4 defenses are broken with this strategy

"*On Adaptive Attacks to Adversarial Example Defenses*", NeurIPS 2020

Some defenses work.

- Adversarial training [Szegedy et al. ‘13], [Goodfellow et al. ‘14], [Kurakin et al. ‘16], [T et al. ‘17], [Madry et al. ‘18], [Zhang et al. ‘19], [Carmon et al. ‘19], [Uesato et al. ‘19], [Zhai et al. ‘19], [Shafahi et al. ‘19], [Yang et al. ‘19], [Li et al. ‘20], ...
- Certified defenses [Katz et al. ‘17], [Wong et al. ‘17], [Raghunathan et al. ‘18], [Gehr et al. ‘18], [Lecuyer et al. ‘18], [Zhang et al. ‘18], [Mirman et al. ‘18], [Weng et al. ‘19], [Baluta et al. ‘19], [Cohen et al. ‘19], [Singh et al. ‘19], [Gluch et al. ‘20], ...

Some defenses work, **but don't generalize...**

- Adversarial training [Szegedy et al. '13], [Goodfellow et al. '14], [Kurakin et al. '16], [T et al. '17], [Madry et al. '18], [Zhang et al. '19], [Carmon et al. '19], [Uesato et al. '19], [Zhai et al. '19], [Shafahi et al. '19], [Yang et al. '19], [Li et al. '20], ...
- Certified defenses [Katz et al. '17], [Wong et al. '17], [Raghunathan et al. '18], [Gehr et al. '18], [Lecuyer et al. '18], [Zhang et al. '18], [Mirman et al. '18], [Weng et al. '19], [Baluta et al. '19], [Cohen et al. '19], [Singh et al. '19], [Gluch et al. '20], ...

recall: we only consider perturbations δ from a *fixed* set S

issue: all defenses above are *explicitly tailored to a chosen set* S

defenses overfit to the chosen set

T, Behrmann, Carlini, Papernot, Jakobsen (ICML 2020)

generalizing to richer sets hurts robustness

T & Boneh (NeurIPS 2019 *spotlight*)

Take away: we **don't** have robust machine learning in adversarial settings.

THE WALL STREET JOURNAL.

TECH

Facebook, YouTube, Twitter Scramble to Remove Video of New Zealand Mosque Shooting

MOTHERBOARD

TECH BY VICE

Researchers Defeat Most Powerful Ad Blockers, Declare a 'New Arms Race'

Take away: we **don't** have robust machine learning in adversarial settings.

But, we now have:

1. *industry awareness of security risks*

ABP Adblock Plus

brave

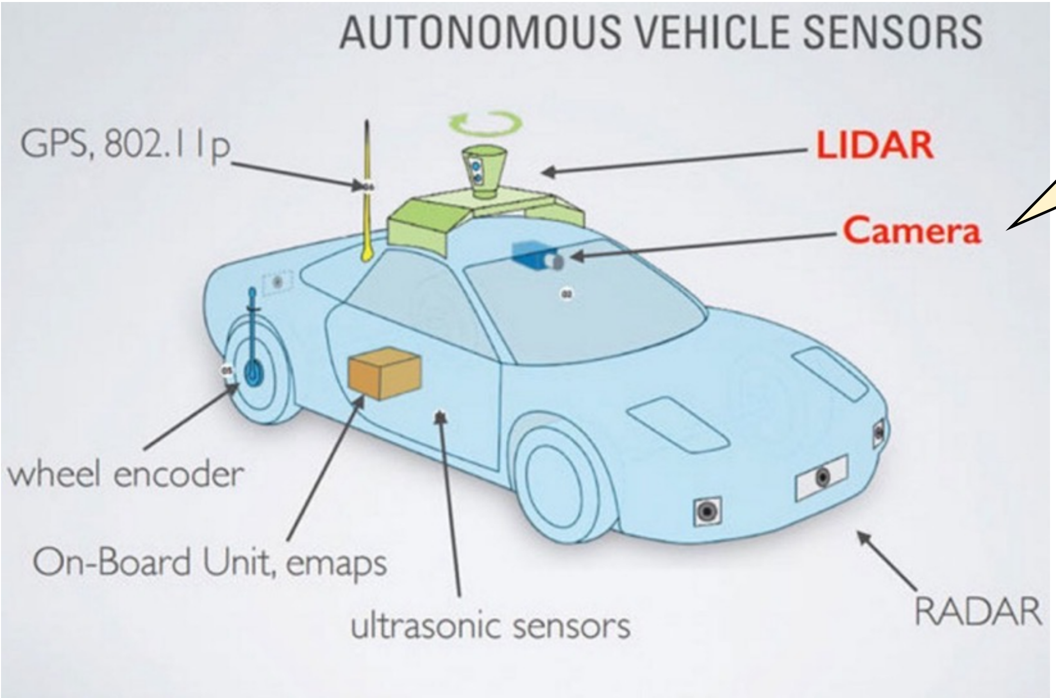
2. *adoption of principled security evaluations*

On adaptive attacks to adversarial example defenses
F Tramer, N Carlini, W Brendel, A Madry
Cited by 101 Related articles

The future: evasion attacks as ***safety*** evaluation.

[Pei et al. ‘17], [Tian et al. ‘17], [Gehr et al. ‘18], [Bansal et al. ‘18], [Ma et al. ‘18], [Sun et al. ‘18], ...

use attacks to *stress-test* ML in safety-critical systems.



our WOOT’18 paper

My work: measuring and enhancing ML security

Evaluations

Evading ML models (NeurIPS ‘20) (ACM CCS ‘19)
Influenced design changes in Adblock Plus

Extracting private data (IEEE S&P ‘21)

Defenses

Training private models (ICLR ‘21 *spotlight*)

Training robust models (NeurIPS ‘19 *spotlight*) (ICLR ‘18)

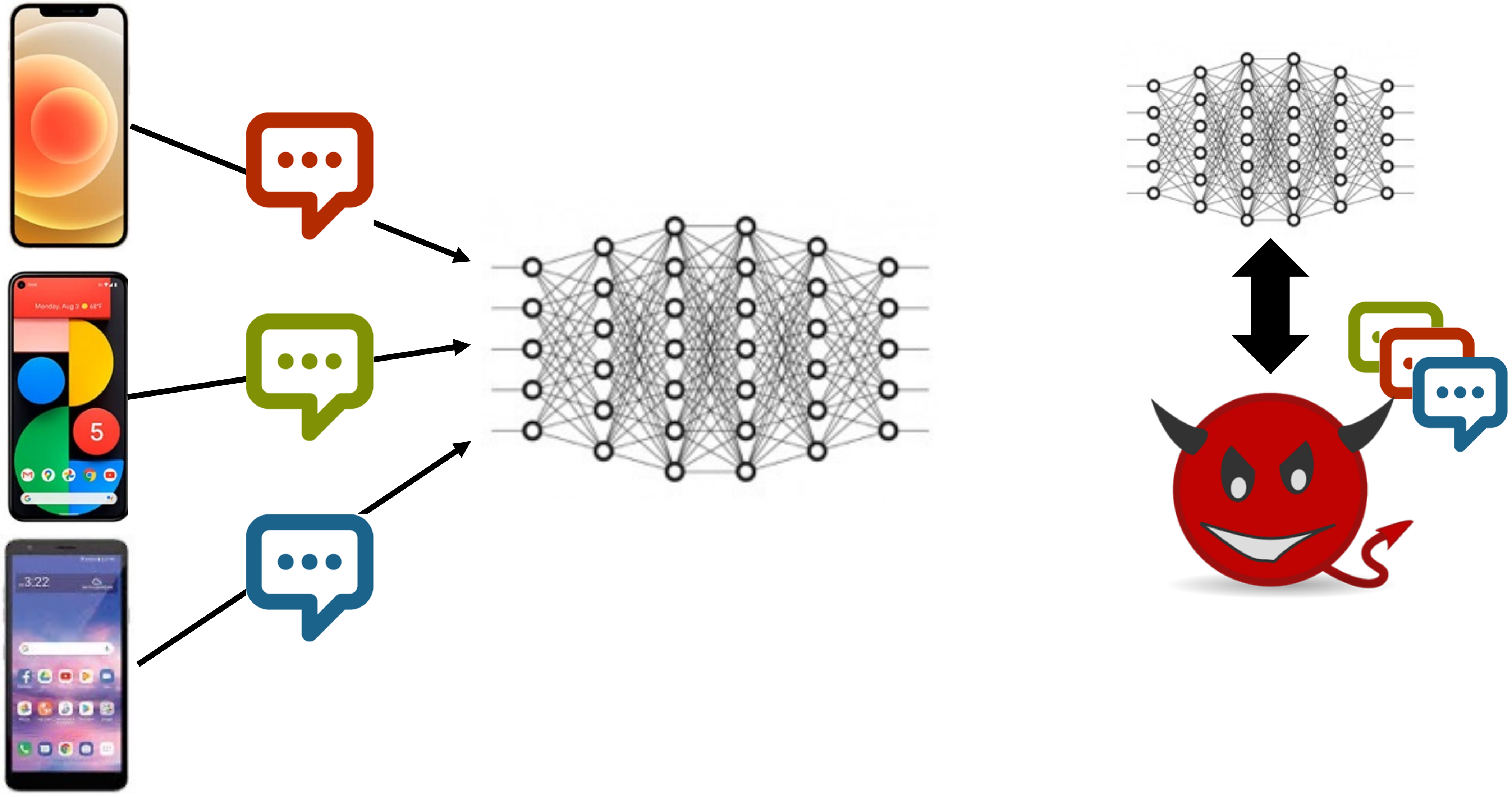
Deploying private models (ICLR ‘19 *oral*)

Foundations

Stealing ML models (USENIX ‘16)
Microsoft’s top 3 threats to AI systems

Threat models for evasion (ICML ‘20)

ML models are trained on private data.



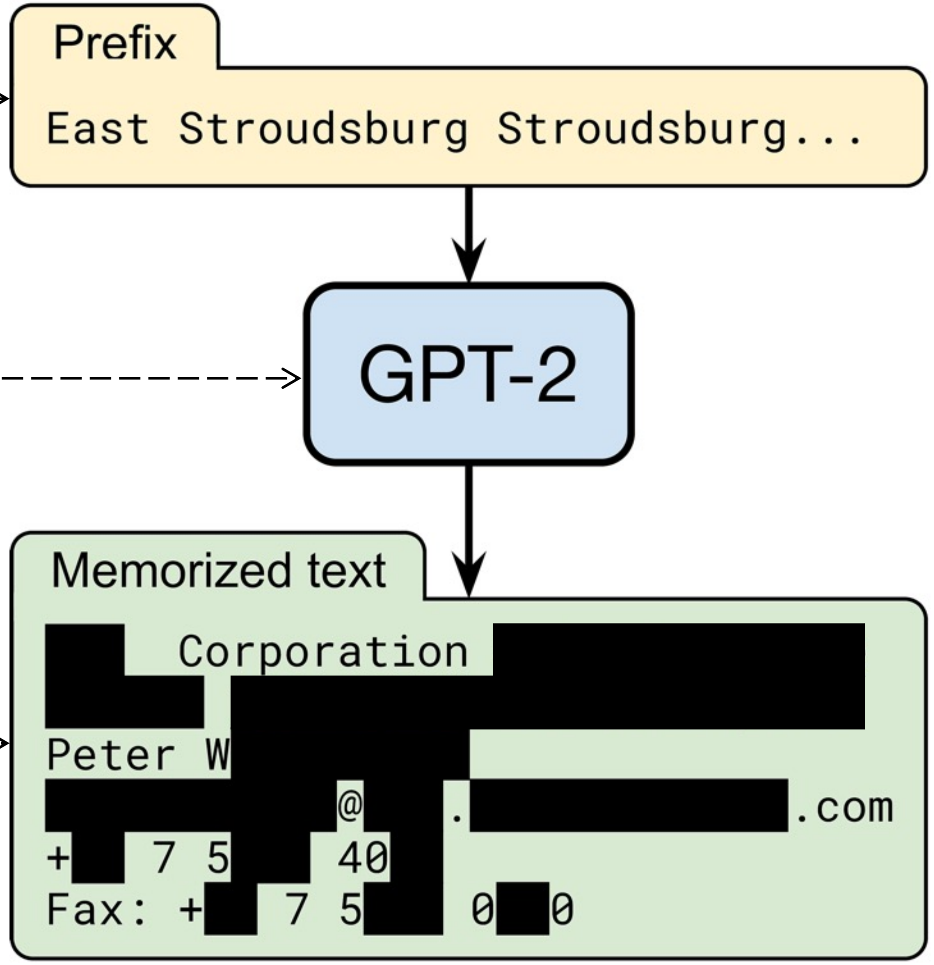
Challenge: models leak their training data.

Carlini, **T**, Wallace, Jagielski, Herbert-Voss, Lee et al. (preprint 2020)

random input

OpenAI's language model trained on text from 8 million web pages

someone's contact information output by the model (***redacted for privacy***)



Data leaks have dramatic ***consequences!***

for users...

The New York Times

Data Breach Victims Talk of Initial Terror, Then Vigilance

for companies...

ZDNet

Facebook could face $1.63bn fine under GDPR over latest data breach

TechCrunch

FTC settlement with Ever orders data and AIs deleted after facial recognition pivot

Preventing data leakage with decade-old ML

T & Boneh (ICLR 2021 *spotlight*)

- *provably* prevent leakage of training data.
  using *differential privacy*

  Extensions: distributed or federated learning
  [Dean et al. '12], [McMahan et al. '16], [Lian et al. '17]

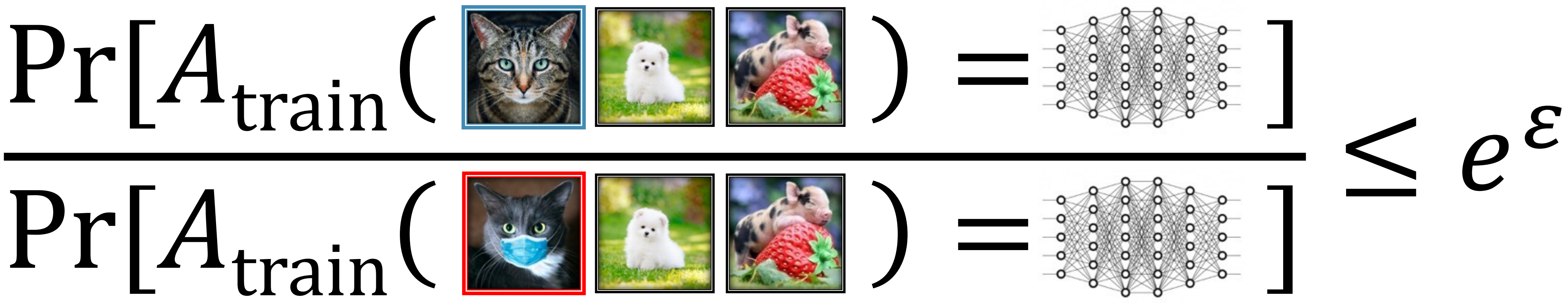
- *better accuracy* than with deep learning methods.
  using domain-specific *feature engineering*

Differential privacy prevents data leakage.

[Dwork et al. ‘06]

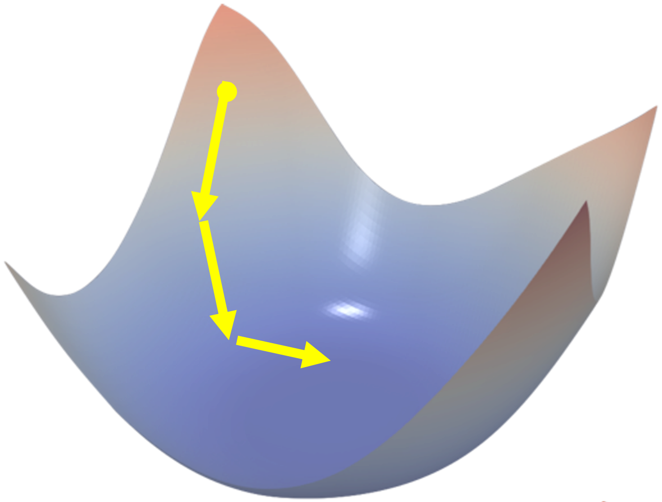
intuition: *randomized* training algorithm is not influenced (too much) by any individual data point

for any two datasets that differ in a single element



Differentially private learning is possible with *noisy* gradient descent.

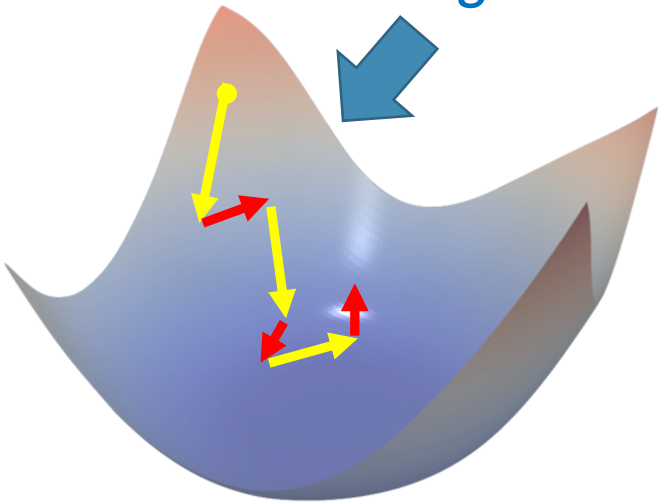
Gradient descent



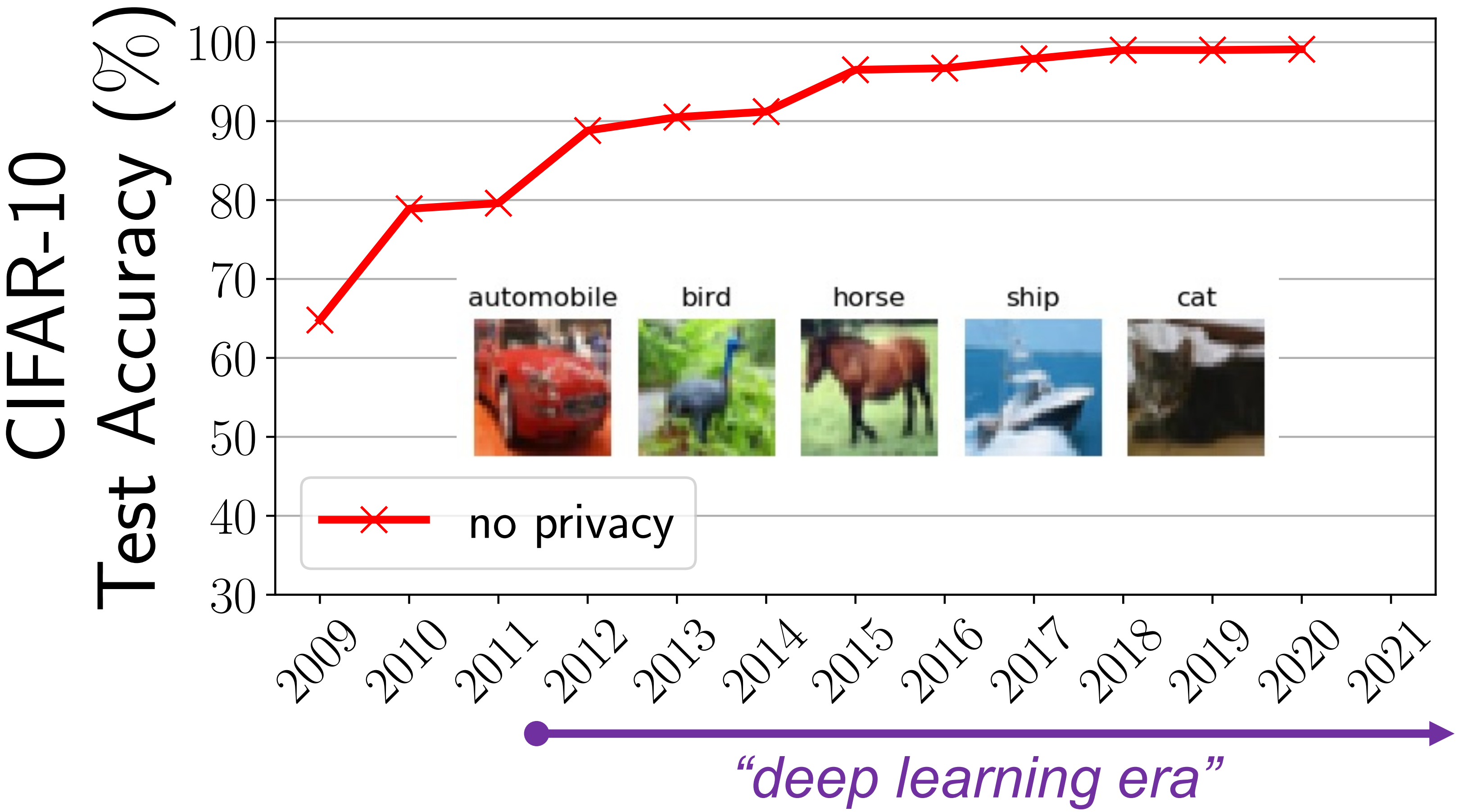
add noise to each step
to guarantee privacy

Private gradient descent

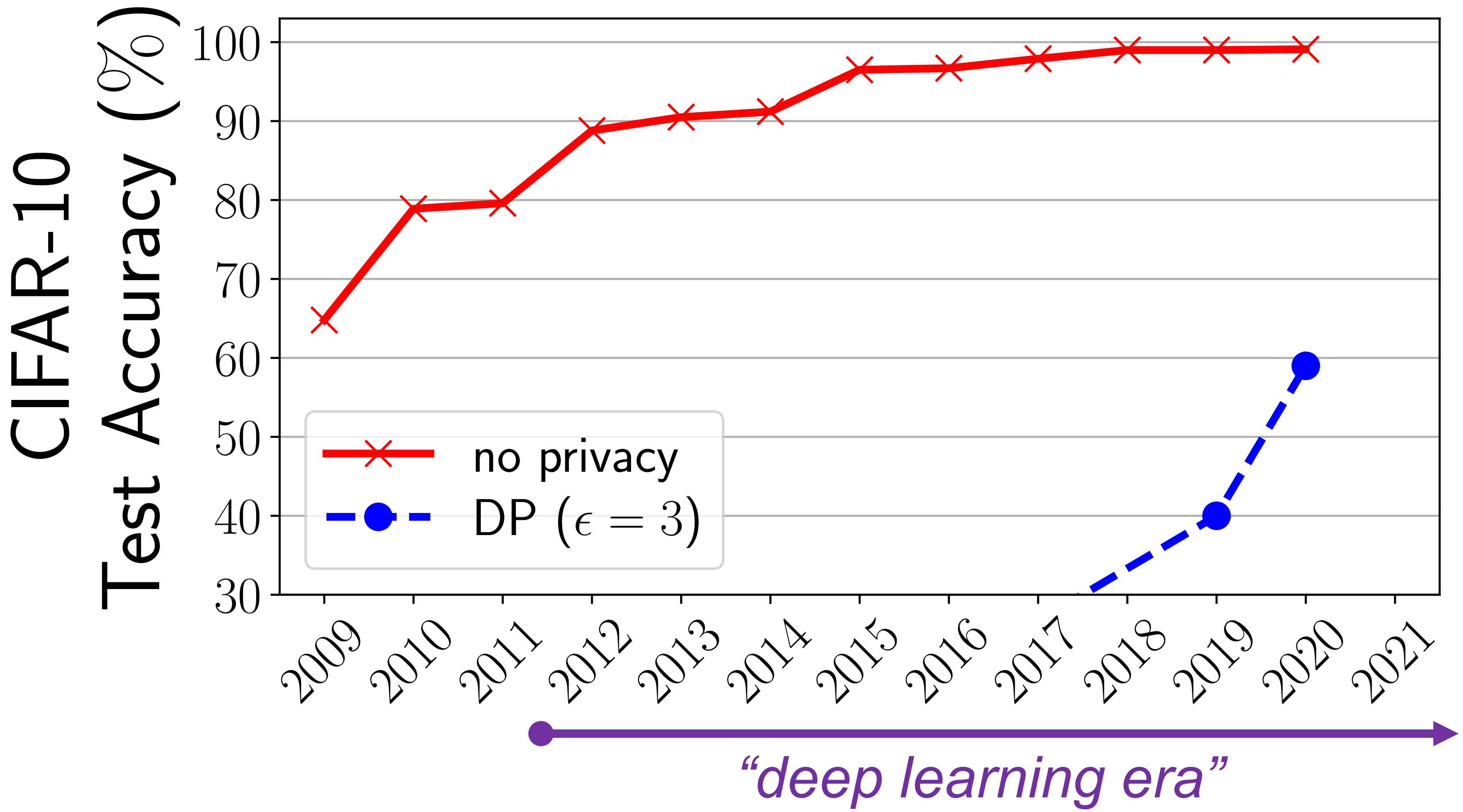
[Chaudhuri et al., ‘11], [Bassily et al. ‘14], [Shokri & Shmatikov ‘15], [Abadi et al. ‘16], ...



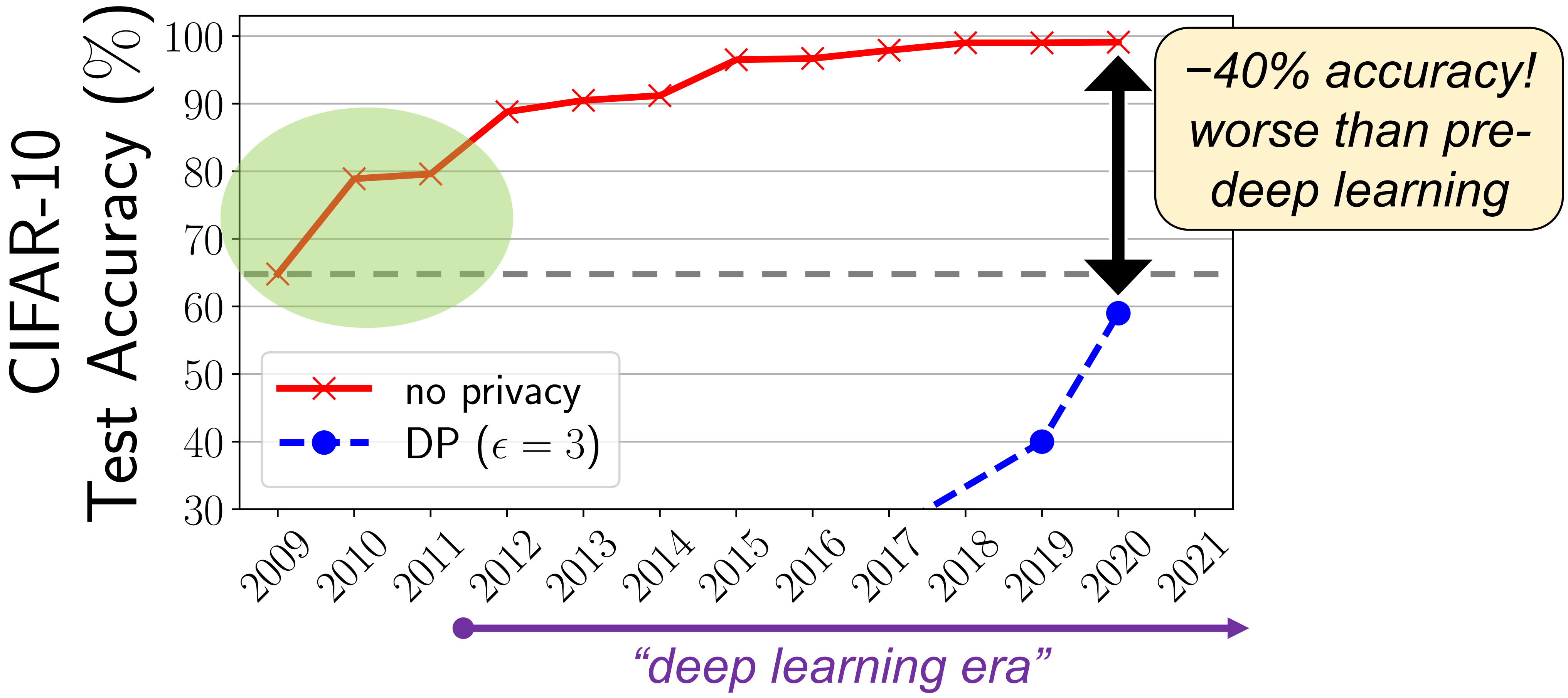
Non-private deep learning can achieve near-perfect accuracy.



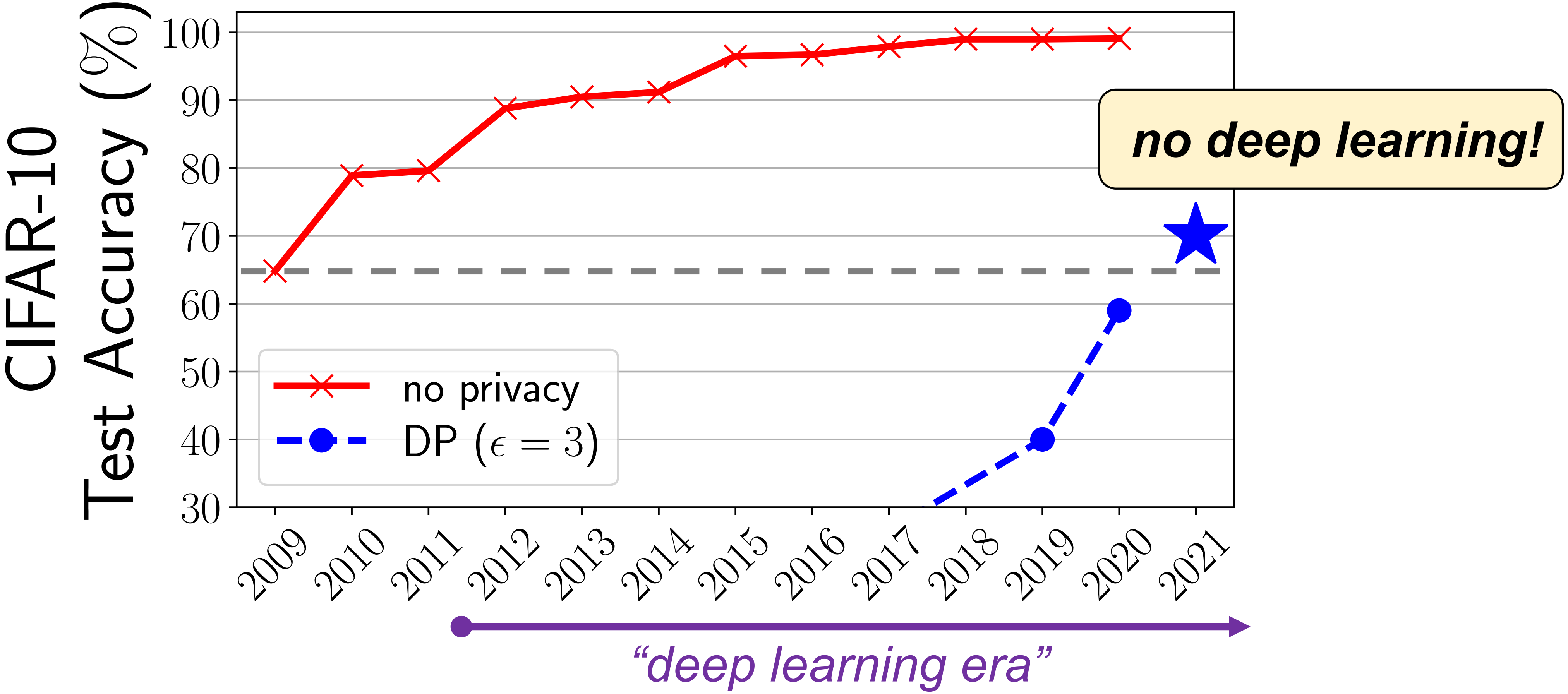
Differentially private deep learning lowers accuracy significantly.



Differentially private deep learning lowers accuracy significantly.

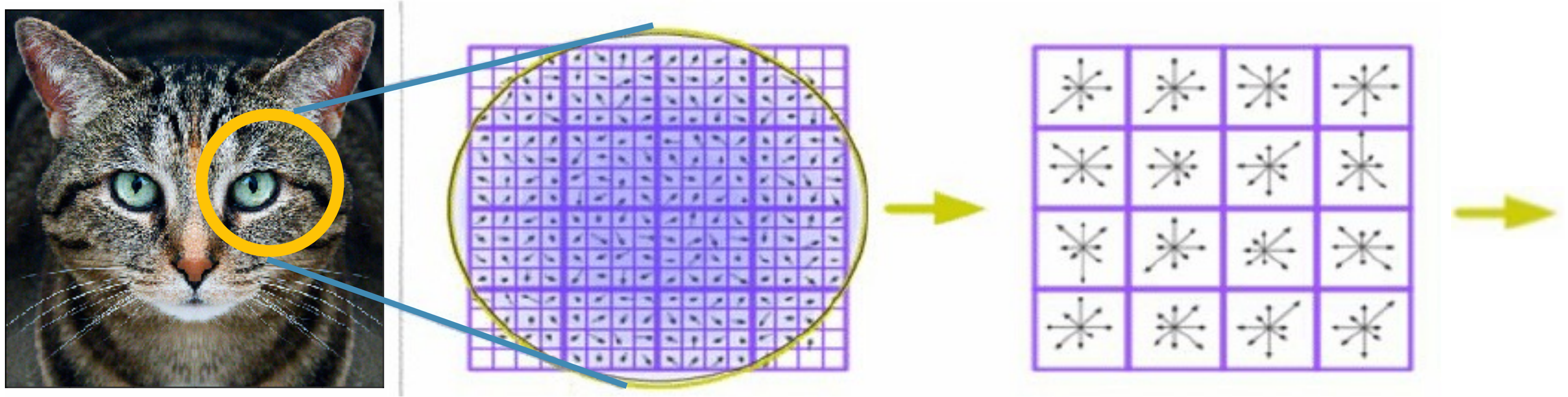


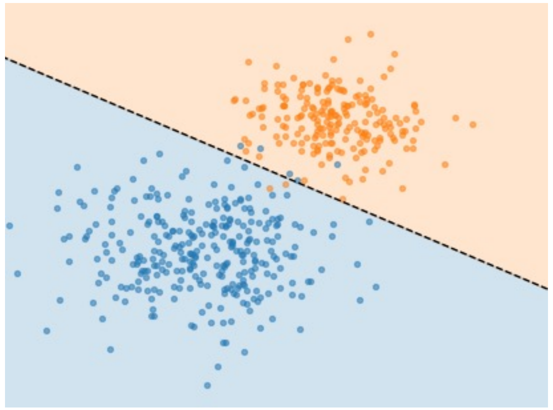
Differential privacy *without deep learning* improves accuracy.



Privacy-free features from “old-school” image recognition.

SIFT [Lowe ‘99, ‘04], HOG [Dalal & Triggs ‘05], SURF [Bay et al. ‘06], ORB [Rublee et al. ‘11], ...
Scattering transforms: [Bruna & Mallat ‘11], [Oyallon & Mallat ‘14], ...





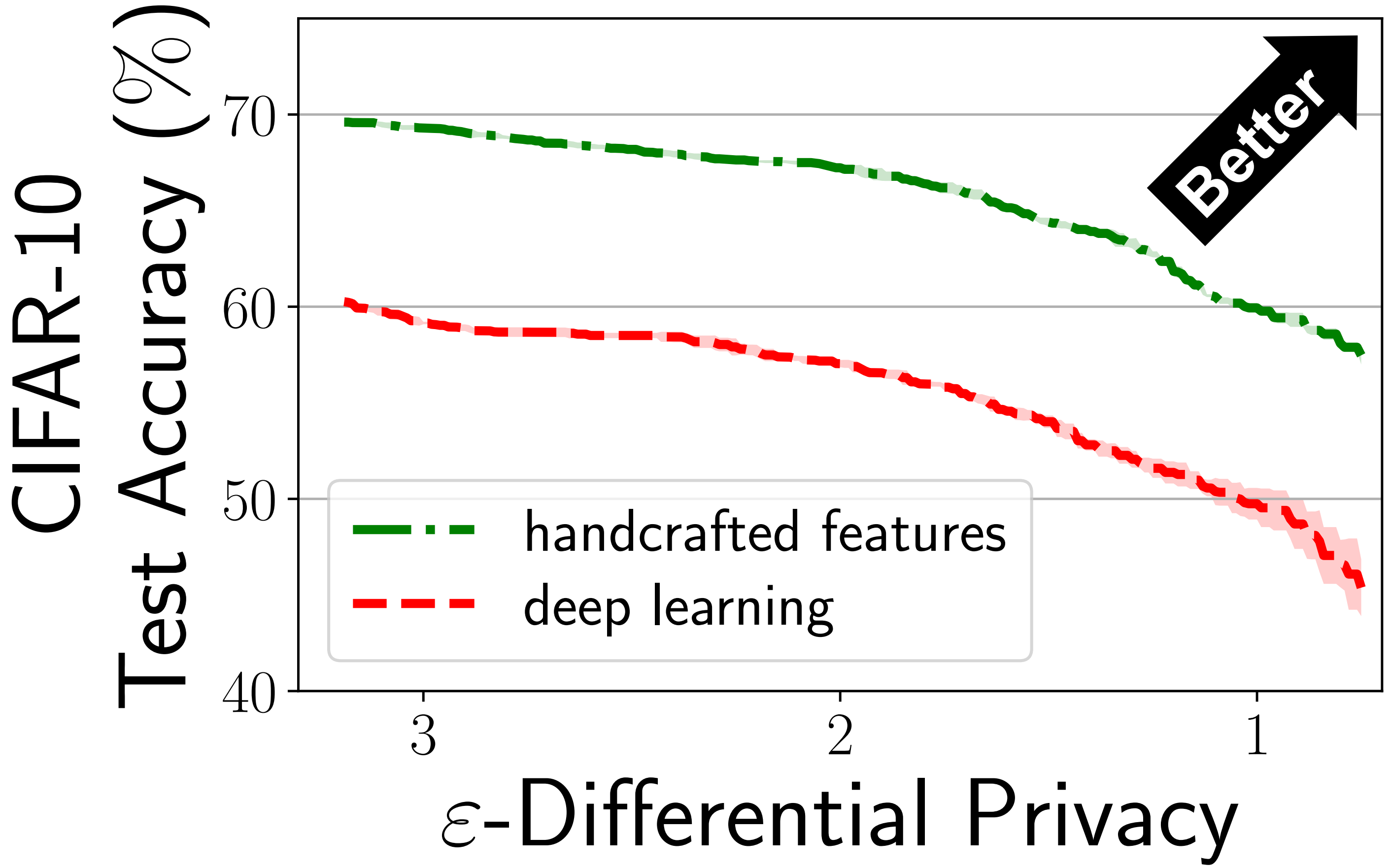
“handcrafted features”
(no learning involved)

simple classifier
(e.g., logistic regression)

privacy free

captures some *prior* about the domain: e.g., invariance under rotation & scaling

Handcrafted features lead to a better tradeoff between accuracy and privacy.

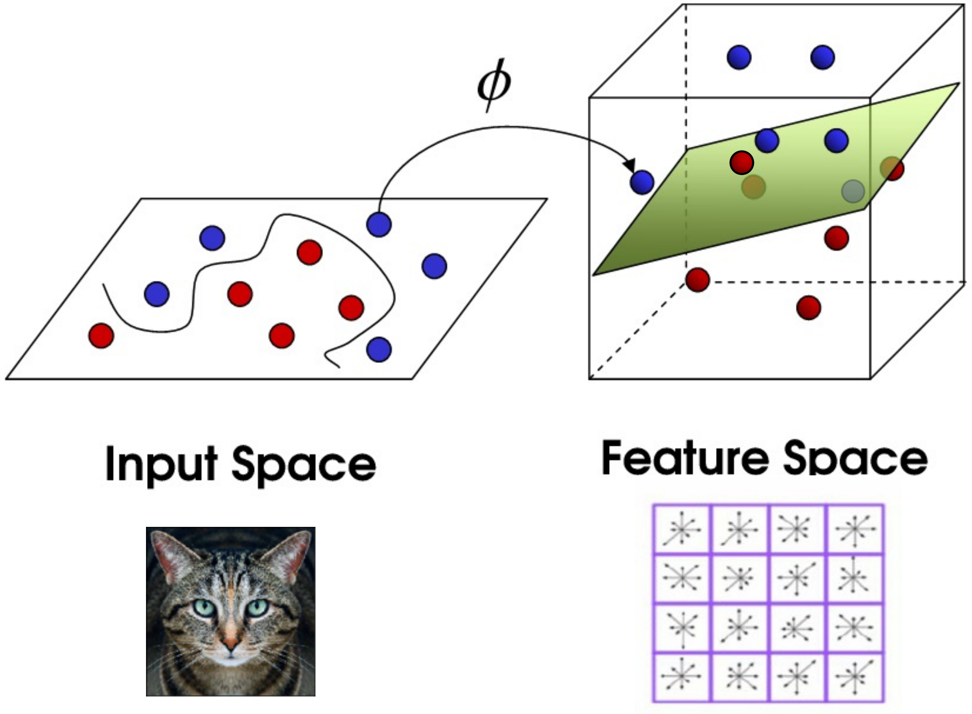


"*Differentially private learning needs better features*", ICLR 2021 *spotlight*

Handcrafted features lead to an *easier* learning task (for noisy gradient descent).

high accuracy classifier exists but learning takes *many gradient steps*

bad for privacy



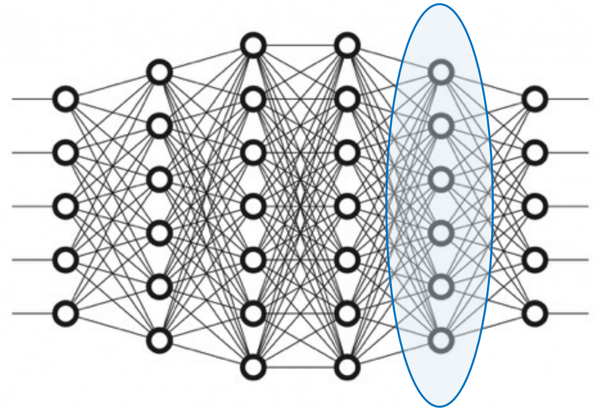
in feature space, maximal accuracy is reduced but *learning progresses faster*

good for privacy

"*Differentially private learning needs better features*", ICLR 2021 *spotlight*

Learning better privacy-free features from public data.

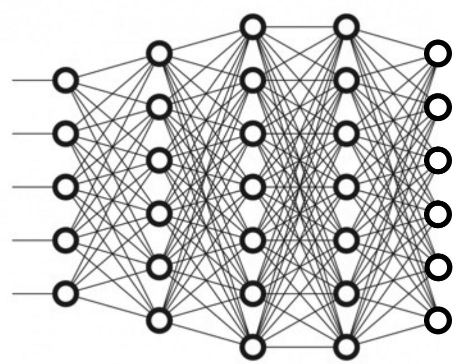
public data



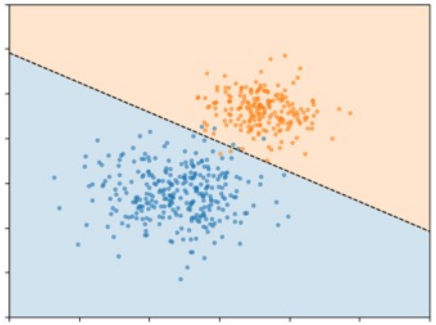
train a feature extractor on public data...

...transfer and fine-tune on private data

private data

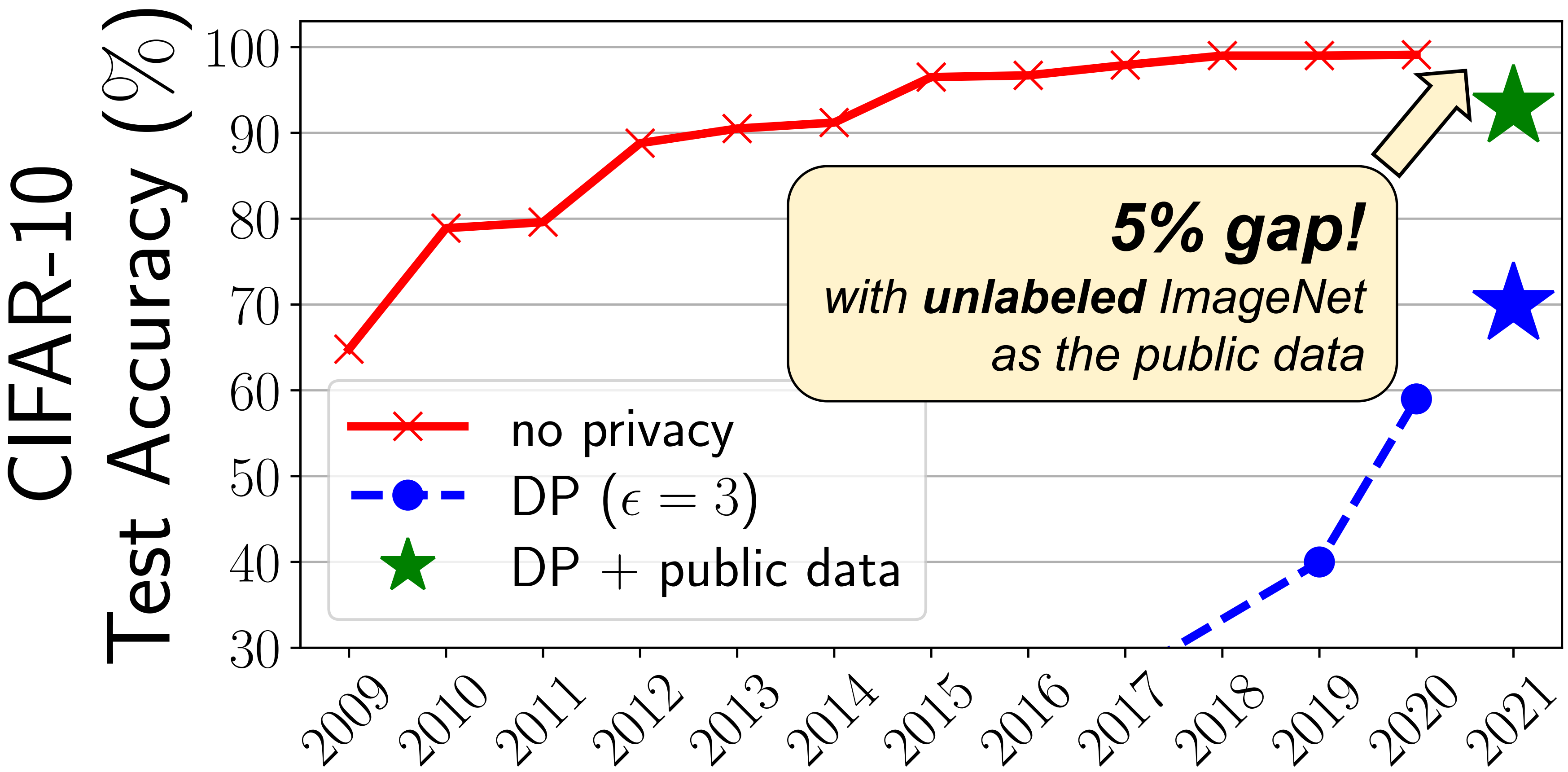


privacy free



"*Differentially private learning needs better features*", ICLR 2021 *spotlight*

With access to a public dataset, privacy comes almost for free!



"*Differentially private learning needs better features*", ICLR 2021 *spotlight*

Differential private learning in industry.

Opacus

tensorflow / privacy

IBM / differential-privacy-library

I added batching support for private gradient descent

I identified and fixed incorrect privacy analyzes

My work: measuring and enhancing ML security

Evaluations

Evading ML models (NeurIPS '20) (ACM CCS '19)

Influenced design changes in Adblock Plus

Extracting private data (IEEE S&P '21)

Defenses

Training private models (ICLR '21 *spotlight*)

Training robust models (NeurIPS '19 *spotlight*) (ICLR '18)

Deploying private models (ICLR '19 *oral*)

Foundations

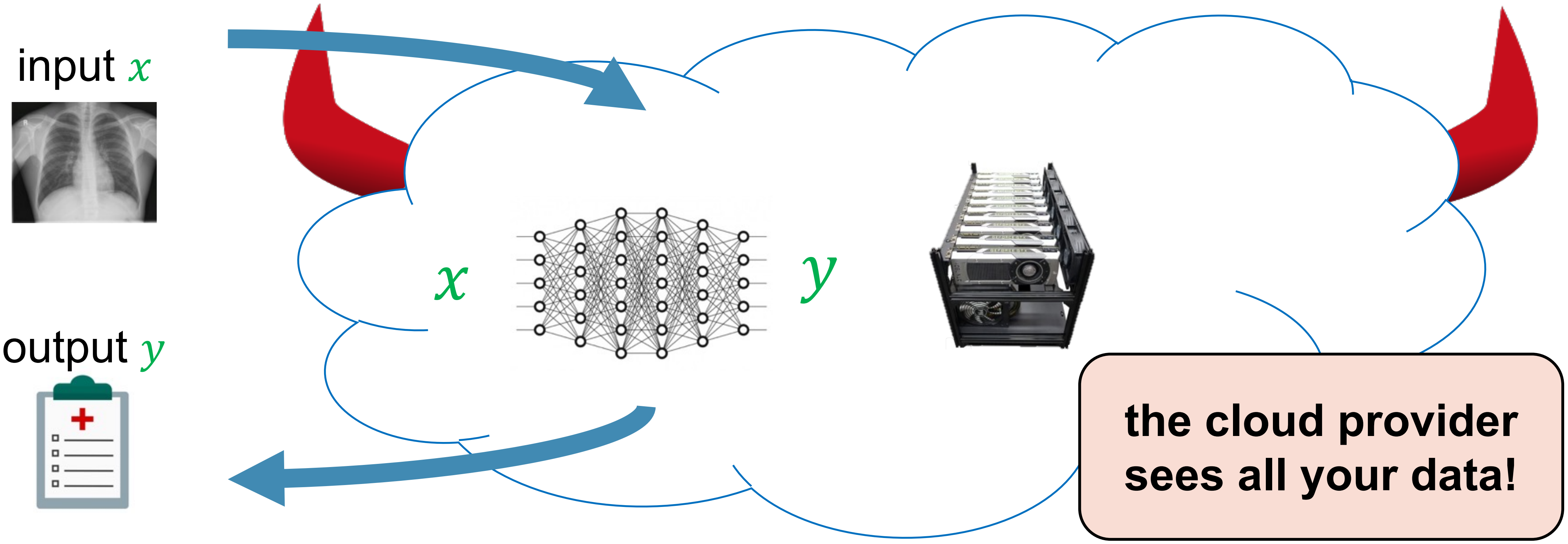
Stealing ML models (USENIX '16)

Microsoft's top 3 threats to AI systems

Threat models for evasion (ICML '20)

Can we *evaluate* neural networks privately?

[Gilad-Bachrach et al. ‘16], [Mohassel et al. ‘17], [Liu et al. ‘17], [Juvekar et al. ‘18], [Hunt et al. ‘18], [Grover et al. ‘18], ...



sensitive applications (e.g., in healthcare) must abide by strict data confidentiality regulations

Slalom: secure cloud deployment of ML

T & Boneh (ICLR 2019 *oral*)

Different from differential privacy!
here, the model is already trained and we want to protect the *test data* of users

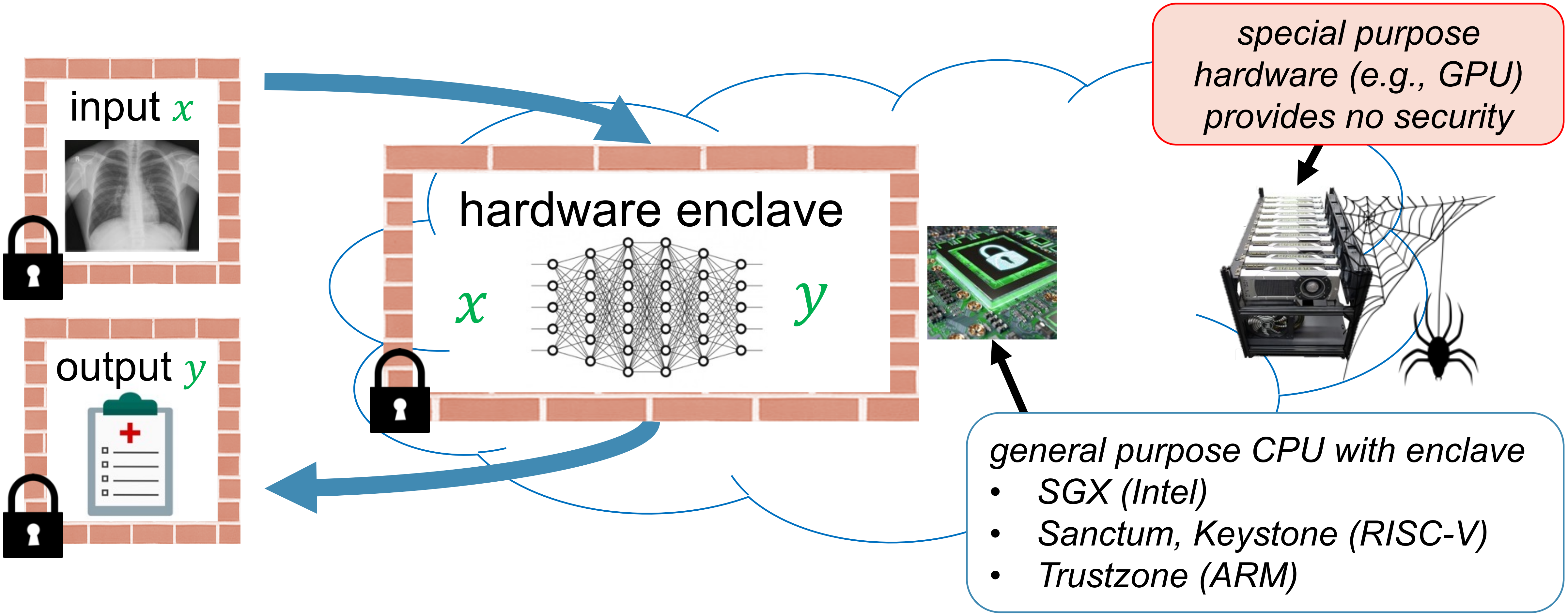
System goals:

- **Confidentiality:** cloud provider does not learn user inputs
- **Integrity:** cloud provider cannot tamper with computation

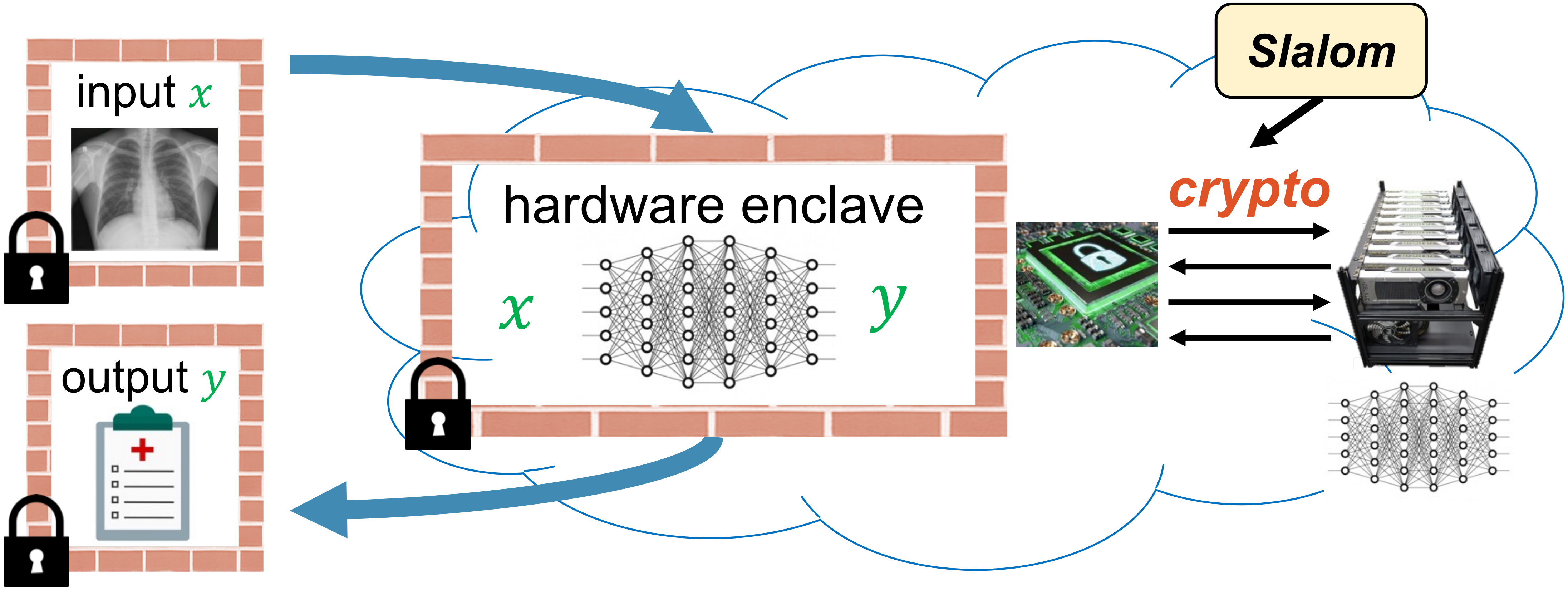
➢ combines ideas from *ML systems*, *hardware security* and *cryptography* to protect user data from a malicious cloud.

➢ maximizes use of cloud's *special-purpose hardware.*

Baseline: security with slow CPU enclaves.



Slalom: security with fast custom hardware



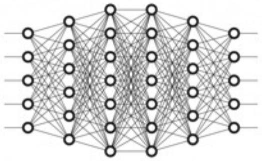
Secure outsourcing of *matrix products.*

user input

X

model weights

W



no confidentiality

$$X \longrightarrow$$

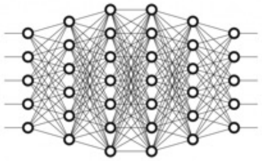
$$WX \longleftarrow$$

Secure outsourcing of *matrix products.*

$X \in \mathbb{Z}_p^{n\times n}$

quantization!
(compute on integers modulo prime)

$W \in \mathbb{Z}_p^{n\times n}$



random mask

$X + \delta$ →

precomputed (independent of input)

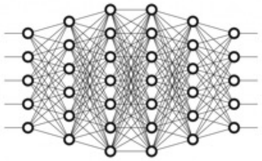
← $WX + W\delta$

“*Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware*”, ICLR 2019 *oral*

Secure outsourcing of *matrix products.*

$X \in \mathbb{Z}_p^{n\times n}$

$W \in \mathbb{Z}_p^{n\times n}$



no integrity!

$\xrightarrow{X + \delta}$

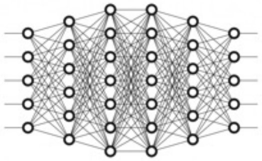
$\xleftarrow{\textit{garbage}}$

“*Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware*”, ICLR 2019 *oral*

Secure outsourcing of *matrix products.*

$X \in \mathbb{Z}_p^{n\times n}$

$W \in \mathbb{Z}_p^{n\times n}$



Probabilistic check

$(Z - W\delta)r \stackrel{?}{=} W(Xr)$

$O(n^2)$ instead of $O(n^3)$

$X + \delta \longrightarrow$

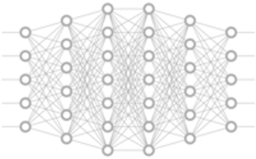
$\longleftarrow Z$

"*Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware*", ICLR 2019 *oral*

Secure outsourcing of *matrix products.*

$X \in \mathbb{Z}_p^{n \times n}$

$W \in \mathbb{Z}_p^{n \times n}$



Probabilistic

$(Z - W\delta)r \stackrel{?}{=}$

$O(n^2)$ instead of $O(n^3)$

Theorem (informal):

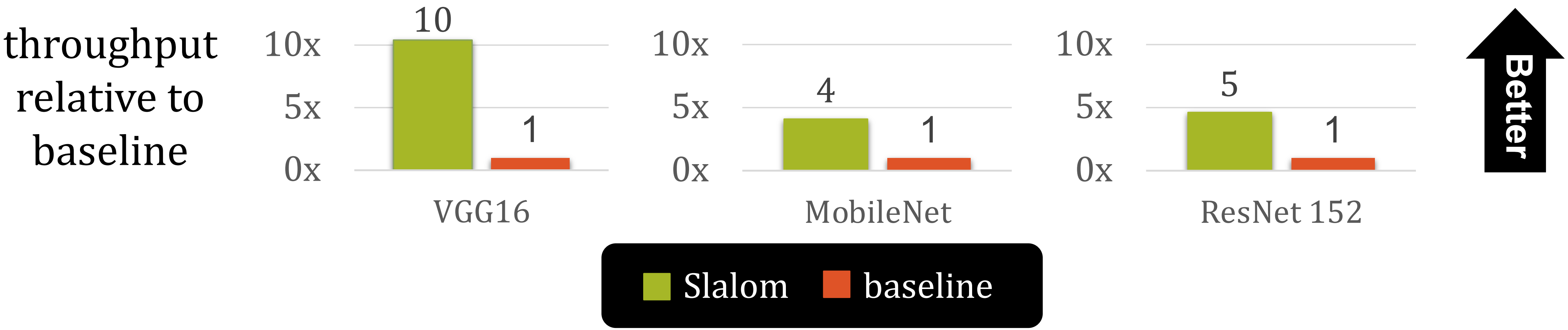
Assuming a secure PRNG, Slalom guarantees confidentiality and integrity (with soundness error k/p for a k-layer neural network).

"*Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware*", ICLR 2019 *oral*

Slalom improves secure inference throughput.

- Intel SGX ↔ Nvidia Titan XP
- ImageNet inference throughput (images per second)
- **Goal: Slalom (Enclave↔GPU) ≫ Enclave$_{\text{baseline}}$**

execute entire model in secure enclave



My work: measuring and enhancing ML security

Evaluations

Evading ML models (NeurIPS ‘20) (ACM CCS ‘19)
Influenced design changes in Adblock Plus

Extracting private data (IEEE S&P ‘21)

Defenses

Training private models (ICLR ‘21 *spotlight*)

Training robust models (NeurIPS ‘19 *spotlight*) (ICLR ‘18)

Deploying private models (ICLR ‘19 *oral*)

Foundations

Stealing ML models (USENIX ‘16)
Microsoft’s top 3 threats to AI systems

Threat models for evasion (ICML ‘20)

Future work

ML security is a critical challenge for our society.

Evaluations

Defenses

Foundations

robustness

fairness

[T et al. '17]

interpretability

Formal foundations for trustworthy ML.

A framework as beautiful as differential privacy for other critical safety properties

Future work

ML security is a critical challenge for our society.

Evaluations

Defenses

Foundations

Cryptography for ML.

Making machine learning secure against ***computationally-bounded*** adversaries

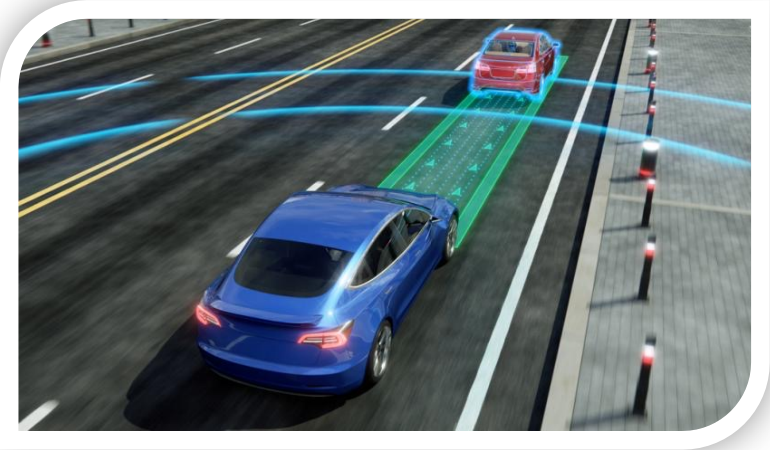
Future work

ML security is a critical challenge for our society.

Evaluations

Defenses

Foundations



Vetting ML safety in critical applications.

Evaluating the failure modes of models once they reach 99.999% accuracy

Conclusion

ML is currently not ***trustworthy***.

- it is not ***robust***.
- it is not ***private***.

We can get better robustness than current ML.

- ➢ *humans are an existence proof.*
- ➢ *we must approach this as a security problem.*

We can get better privacy than current ML.

- ➢ *with differential privacy and cryptography.*

Conclusion

ML is currently not ***trustworthy***.

- it is not ***robust***.
- it is not ***private***.

We can get better robustness than current ML.

- ➢ *humans are an existence proof.*
- ➢ *we must approach this as a security problem.*

We can get better privacy than current ML.

- ➢ *with differential privacy and cryptography.*

Thank you!