Detecting Adversarial Examples Is (Nearly) As Hard As Classifying Them

Florian Tramèr

Stanford University

Thanks: Wieland Brendel, Nicholas Carlini, Alex Ozdemir

This robust classifier sounds implausible.

- Dataset: CIFAR-10
- Norm: ℓ_∞
- Bound: $\varepsilon = {}^{8}/_{255}$
- Robust accuracy: 80%

(current SOTA is ~65%)

What about this robust detector?

- Dataset: CIFAR-10
- Norm: ℓ_∞
- Bound: $\varepsilon = {}^{16}/_{255}$
- Robust **detection** accuracy: 80%

(current SOTA ???)

Defense is allowed to **abstain** if it detects an adversarial example

These two claims are equivalent!

(if we disregard computational complexity)

main idea:
"minimum distance decoding"

Theorem (informal):

detector with robust accuracy α for attacks of size $\boldsymbol{\varepsilon}$

$\Leftrightarrow$

classifier with robust accuracy α for attacks of size $\boldsymbol{\varepsilon/2}$

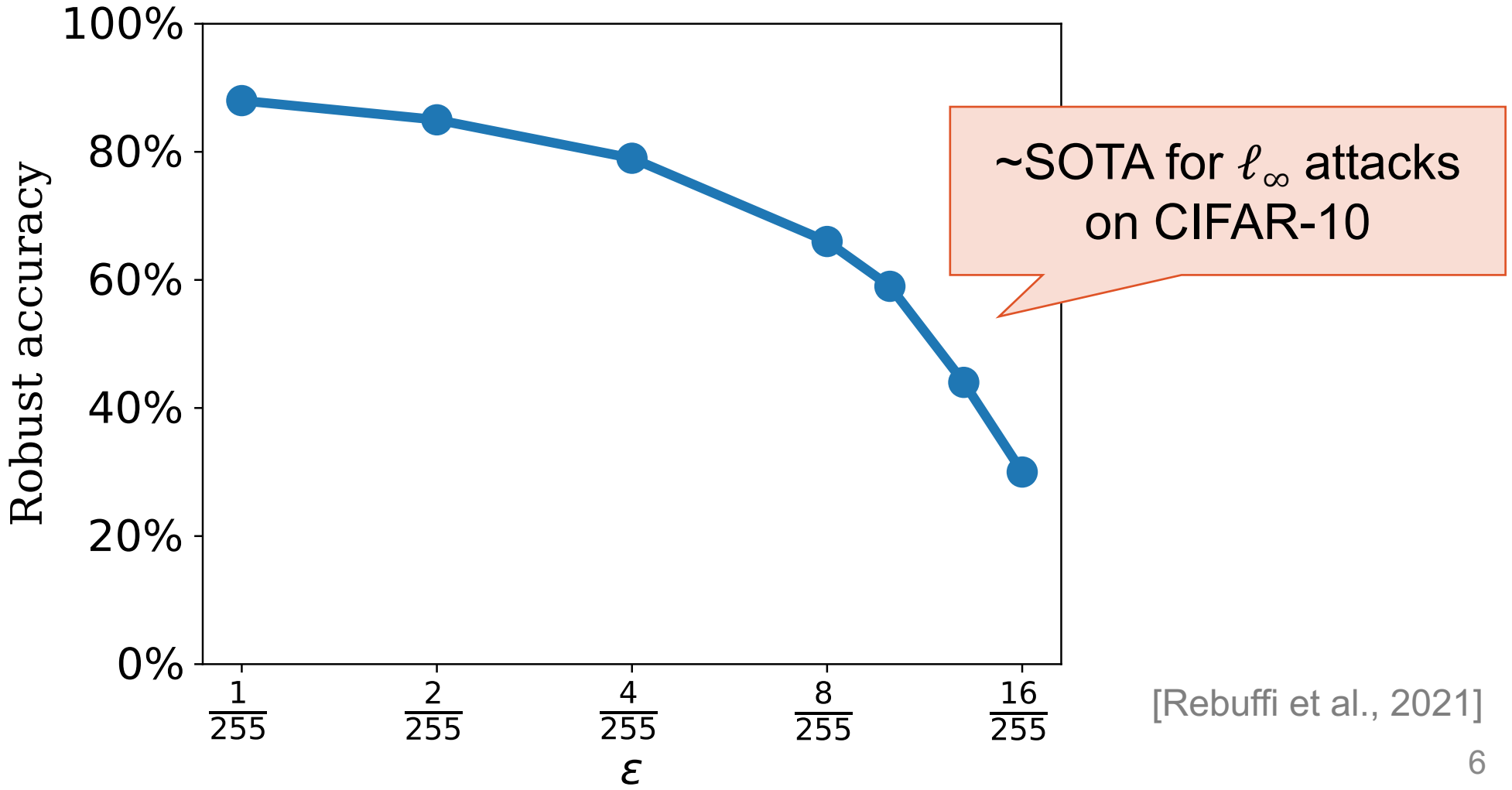
Caveat: the reduction is *computationally inefficient*

What is this reduction useful for?
Theory: port *unconditional* results to detectors

- Robust generalization [Schmidt et al., 2018]
- Accuracy-robustness tradeoff [Tsipras et al., 2019, Zhang et al., 2019]
- Multi-robustness tradeoff [T & Boneh, 2019, Maini et al., 2020]
- Robustness vs error on noise [Ford et al., 2020]
- ...

What is this reduction useful for?

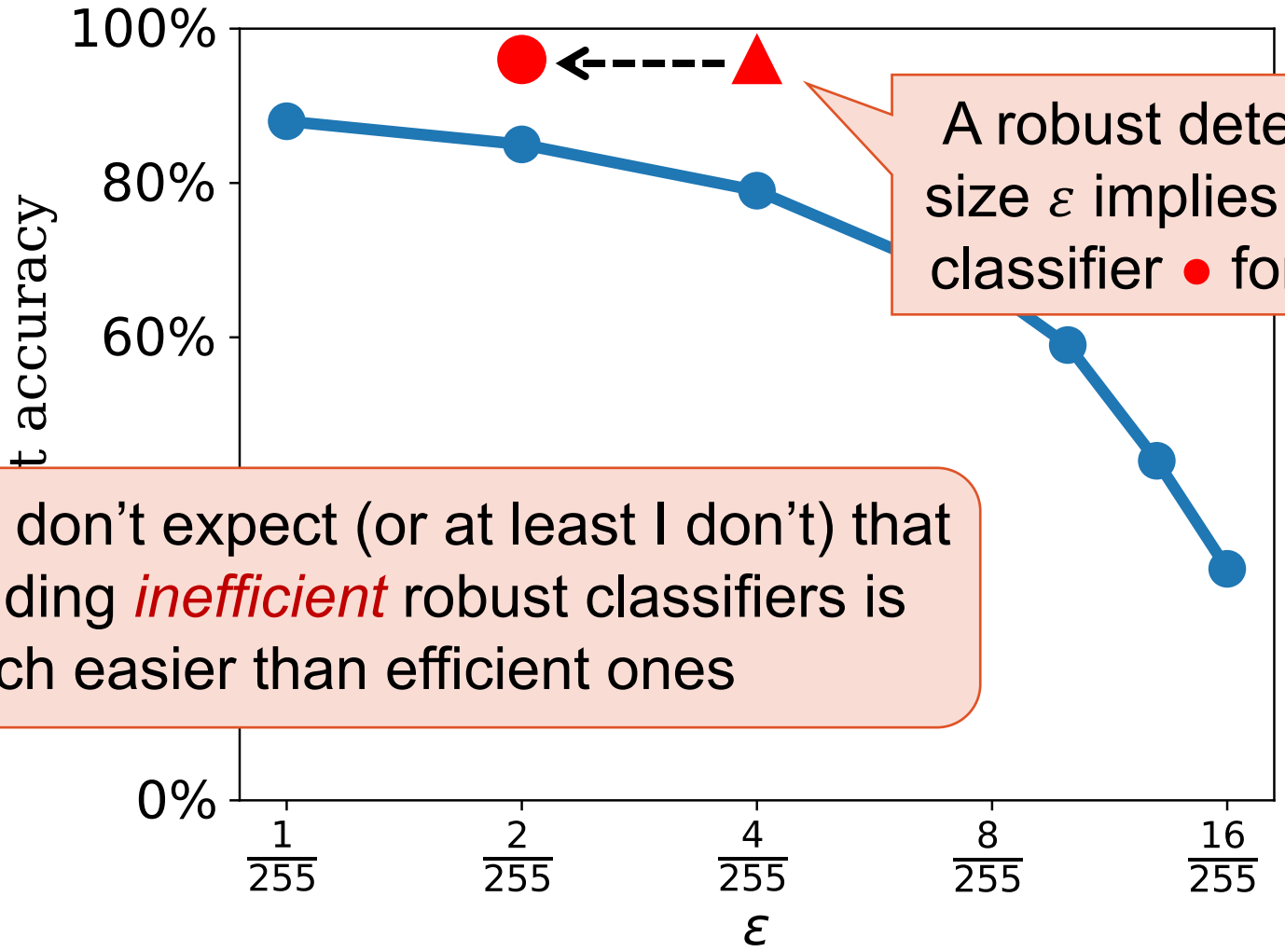
Practice: a sanity check for detector defenses.



[Rebuffi et al., 2021]

What is this reduction useful for?

Practice: a sanity check for detector defenses.

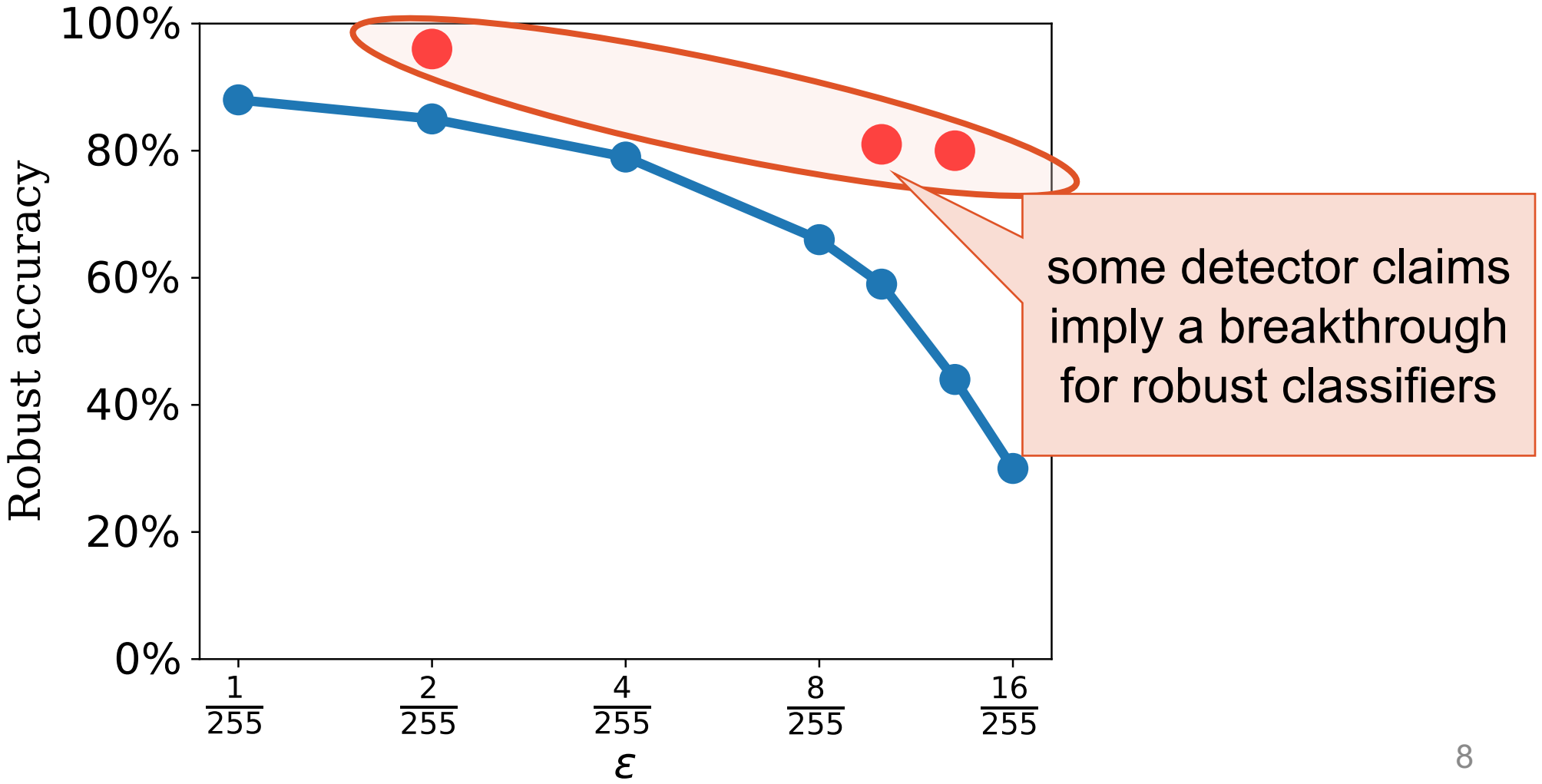


A robust detector ▲ for attacks of size ε implies an inefficient robust classifier ● for attacks of size $\varepsilon/2$

We don't expect (or at least I don't) that building *inefficient* robust classifiers is much easier than efficient ones

What is this reduction useful for?

Practice: a sanity check for detector defenses.



Take-away:

if robust classification is hard,
so is robust detection!

Open problems:

- Is there an efficient detector ⇔ classifier reduction?
- What about detectors that claim conditional robustness?

Thanks!