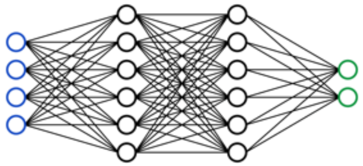
Why you should treat your ML defense like a *theorem*

Florian Tramèr

Stanford → Google → **Vacation** → ETHZ

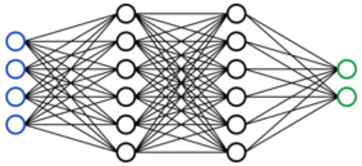
Defense ⇔ Theorem

Theorem 7.7. $\mathrm{P} \neq \mathrm{NP}$.

My  is robust.

Defense ⇔ Theorem, Evaluation ⇔ Proof.

Theorem 7.7. P $\neq$ NP.

My  is robust.

Proof. Consider the solution space of k-SAT in the d1RSB phase for $k > 8$ as recalled in Section.5.2.1. We know that for high enough values of the clause density α, we have $O(n)$ frozen variables in almost all of the exponentially many clusters. Let us consider the situation where these clusters were generated by a purported LFP algorithm for k-SAT. However, when exponentially many solutions have been generated from distributions having the parametrization of the ENSP model, we will see the effect of conditional independencies beyond range $\text{poly}(\log n)$. Let $\alpha\beta\gamma$ be a representation of the variables in cliques α, β and γ, then given a value of β, we will see independent variation over all their possible conditional values in the variables of α and γ. If each set of such variables has scope at most $\text{poly}(\log n)$, then this means that once more than $c^{\text{poly}(\log n)}$, $c > 1$ many distinct solutions are generated, we have non-trivial conditional distributions conditioned upon values of β variables (this factor accounts for the possible orderings within the $\text{poly}(\log n)$ variables as well). At this point, the conditional independence ensure that we will see cross terms of the form

$$\alpha_1\beta\gamma_1 \qquad \alpha_2\beta\gamma_2 \qquad \alpha_1\beta\gamma_2 \qquad \alpha_2\beta\gamma_1.$$

Note that since $O(n)$ variables have to be changed when jumping from one cluster to another, we may even chose our $\text{poly}(\log n)$ blocks to be in overlaps of these variables. This would mean that with a $\text{poly}(\log n)$ change in frozen variables of one cluster, we would get a solution in another cluster. But we know that in the highly constrained phases of d1RSB, we need $O(n)$ variable flips to get from one cluster to the next. This gives us the contradiction that we seek. ∎

6 ADAPTIVE ATTACK EVALUATION

Given that our proposed defense effectively prevents existing black-box attacks, we now study whether or not it can prevent more sophisticated attacks. We find that while it is possible to degrade the effectiveness of the defense, we can not defeat it completely. We study three categories of attacks: gradient attacks (specifically, variants of NES with various kinds of query blinding), boundary-following attacks (specifically, variants of the boundary attack with query blinding), and hybrid attacks (which combine gradient attacks with a surrogate model).

6.1 The NES Attack

We use the NES attack as one starting point for attacking our scheme. To generate a targeted adversarial example a given input x, NES generates an adversarial example by seeding it with an image x' of the target class (i.e., that is already adversarial). NES then uses projected gradient descent to slowly reduce the distortion between this image (which is already the target class) and the original example x until it is within ϵ of the original image, while still being classified as the target class.

In order to estimate the gradient at any given location x, NES uses finite differences on a random Gaussian basis. Full details can be found in [22], but simplified, the gradient is estimated by: (1) sampling n instances of Gaussian noise $\delta_1, \ldots, \delta_n \sim \mathcal{N}(0,1)$ and adding them each to x as $\theta_i = x + \sigma\delta_i$ to generate n basis points, (2) for each basis point θ_i, estimating the confidence scores at θ_i, (3) estimating the gradient at x using these estimated confidence scores and then taking a step in the direction of the estimated gradient. The confidence score at θ_i is estimated by querying the labels for s points near θ_i chosen randomly from a sampling ball of l_∞ radius μ and computing the proportion of each class as the estimate for that class's confidence score. The default attack parameters for NES are $\sigma = 0.001$, $n = 4$, $s = 50$, $\mu = 0.001$, and learning rate = 0.01 [22]; we consider below how to adjust them.

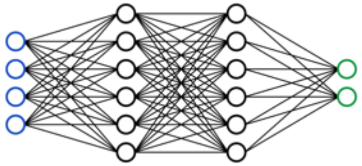
6.1.2 Query Blinding. The second natural modification is for an attacker to use query blinding to transform each query, as described previously. In particular, we modify the confidence score estimation procedure (step 2 from the previous attack description) to sample s points using the different transformations listed in Section 5.1 instead of sampling from a ℓ_∞ ball of uniform radius. The parameters for each transformation are normalized so that the expected ℓ_2 distortion from each transformation is equal to 2.32; exact values are given in Appendix B. We selected a constant of 2.32 to match the ℓ_2 distortion of setting $\mu = 0.064$. (Note that μ is now only applicable when using the original strategy of sampling from a ball of ℓ_∞ radius μ). We used the same parameters when training the similarity encoder.

When running the NES attack, each time we query the classifier we preprocess the image with one strategy. We use default parameter values for the NES attack, except we set $s = 2$ to make it harder to detect. Table 2 shows the effectiveness of different transformations. For some transformations, like uniform and Gaussian noise, the NES attack fails completely. However, the NES attack works even better with brightness and pixel-scale transformations than the original confidence estimation procedure of uniform noise. This suggests that estimating the confidence score for an image may be more accurate with certain image transformations than others.

For all transformations, each attack will trigger at least one hundred detections (on average), so our defense is effective at detecting these query blinding attacks. The exact attacker cost corresponding to this number of detections is quantified further in Section 7.

For this level of transformation distortion, the similarity encoder offers little benefit over ℓ_2 distance on images. This is understandable, as for $k = 50$, the ℓ_2 detection threshold is $\delta = 5.069$ when using ℓ_2 distance on images, which is greater than the ℓ_2 distortion of 2.32 introduced by these transformations. We also evaluated against attacks that use transformations that introduce a greater distortion. Specifically, we increase σ to $\sigma = 0.01$ (the highest value

Today: refuting a defense = building an attack.

My  is robust.

6 ADAPTIVE ATTACK EVALUATION

Given that our proposed defense effectively prevents existing black-box attacks, we now study whether or not it can prevent more sophisticated attacks. We find that while it is possible to degrade the effectiveness of the defense, we can not defeat it completely. We study three categories of attacks: gradient attacks (specifically, variants of NES with various kinds of query blinding), boundary-following attacks (specifically, variants of the boundary attack with query blinding), and hybrid attacks (which combine gradient attacks with a surrogate model).

6.1 The NES Attack

We use the NES attack as one starting point for attacking our scheme. To generate a targeted adversarial example a given input x, NES generates an adversarial example by seeding it with an image x' of the target class (i.e., that is already adversarial). NES then uses projected gradient descent to slowly reduce the distortion between this image (which is already the target class) and the original example x until it is within ϵ of the original image, while still being classified as the target class.

In order to estimate the gradient at any given location x, NES uses finite differences on a random Gaussian basis. Full details can be found in [22], but simplified, the gradient is estimated by: (1) sampling n instances of Gaussian noise $\delta_1, \ldots, \delta_n \sim \mathcal{N}(0,1)$ and adding them each to x as $\theta_i = x + \sigma\delta_i$ to generate n basis points, (2) for each basis point θ_i, estimating the confidence scores at θ_i, (3) estimating the gradient at x using these estimated confidence scores and then taking a step in the direction of the estimated gradient. The confidence score at θ_i is estimated by querying the labels for s points near θ_i chosen randomly from a sampling ball of l_∞ radius μ and computing the proportion of each class as the estimate for that class's confidence score. The default attack parameters for NES are $\sigma = 0.001$, $n = 4$, $s = 50$, $\mu = 0.001$, and learning rate = 0.01 [22]; we consider below how to adjust them.

6.1.2 Query Blinding. The second natural modification is for an attacker to use query blinding to transform each query, as described previously. In particular, we modify the confidence score estimation procedure (step 2 from the previous attack description) to sample s points using the different transformations listed in Section 5.1 instead of sampling from a ℓ_∞ ball of uniform radius. The parameters for each transformation are normalized so that the expected ℓ_2 distortion from each transformation is equal to 2.32; exact values are given in Appendix B. We selected a constant of 2.32 to match the ℓ_2 distortion of setting $\mu = 0.064$. (Note that μ is now only applicable when using the original strategy of sampling from a ball of ℓ_∞ radius μ). We used the same parameters when training the similarity encoder.

When running the NES attack, each time we query the classifier we preprocess the image with one strategy. We use default parameter values for the NES attack, except we set $s = 2$ to make it harder to detect. Table 2 shows the effectiveness of different transformations. For some transformations, like uniform and Gaussian noise, the NES attack fails completely. However, the NES attack works even better with brightness and pixel-scale transformations than the original confidence estimation procedure of uniform noise. This suggests that estimating the confidence score for an image may be more accurate with certain image transformations than others.

For all transformations, each attack will trigger at least one hundred detections (on average), so our defense is effective at detecting these query blinding attacks. The exact attacker cost corresponding to this number of detections is quantified further in Section 7.

For this level of transformation distortion, the similarity encoder offers little benefit over ℓ_2 distance on images. This is understandable, as for $k = 50$, the ℓ_2 detection threshold is $\delta = 5.069$ when using ℓ_2 distance on images, which is greater than the ℓ_2 distortion of 2.32 introduced by these transformations. We also evaluated against attacks that use transformations that introduce a greater distortion. Specifically, we increase σ to $\sigma = 0.01$ (the highest value

Today: refuting a defense = building an attack.

SoK: How Robust is Image Classification Deep Neural Network Watermarking? (Extended Version)

Nils Lukas, Edward Jiang, Xinda Li, Florian Kerschbaum

Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods

Nicholas Carlini David Wagner
University of California, Berkeley

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

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

On Adaptive Attacks to Adversarial Example Defenses

Florian Tramèr* Stanford University
Nicholas Carlini* Google
Wieland Brendel* University of Tübin
Aleksander Mądry MIT

Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks

Francesco Croce [1] Matthias Hein [1]

What’s next? Breaking 100 defenses?

Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods

10 defenses

SoK: How Robust is Image Classification Deep Neural Network Watermarking?

11 defenses

Nils Lukas, Edward Jiang, Xinda Li, Florian Kerschbaum

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

7 defenses

On Adaptive Attacks to Adversarial Example Defenses

Florian Tra … d Brendel*
Stanford Uni … of Tübin
MIT

13 defenses

Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks

35 defenses

This is *not* how we refute theorems!

Theorem 7.7. $\mathrm{P} \neq \mathrm{NP}$.

Proof. Consider the solution space of k-SAT in the d1RSB phase for $k > 8$ as recalled in Section.5.2.1. We know that for high enough values of the clause density α, we have $O(n)$ frozen variables in almost all of the exponentially many clusters. Let us consider the situation where these clusters were generated by a purported LFP algorithm for k-SAT. However, when exponentially many solutions have been generated from distributions having the parametrization of the ENSP model, we will see the effect of conditional independencies beyond range $\operatorname{poly}(\log n)$. Let $\alpha\beta\gamma$ be a representation of the variables in cliques α, β and γ, then given a value of β, we will see independent variation over all their possible conditional values in the variables of α and γ. If each set of such variables has scope at most $\operatorname{poly}(\log n)$, then this means that once more than $c^{\operatorname{poly}(\log n)}, c > 1$ many distinct solutions are generated, we have non-trivial conditional distributions conditioned upon values of β variables (this factor accounts for the possible orderings within the $\operatorname{poly}(\log n)$ variables as well). At this point, the conditional independence ensure that we will see cross terms of the form

$$\alpha_1\beta\gamma_1 \qquad \alpha_2\beta\gamma_2 \qquad \alpha_1\beta\gamma_2 \qquad \alpha_2\beta\gamma_1.$$

Note that since $O(n)$ variables have to be changed when jumping from one cluster to another, we may even chose our $\operatorname{poly}(\log n)$ blocks to be in overlaps of these variables. This would mean that with a $\operatorname{poly}(\log n)$ change in frozen variables of one cluster, we would get a solution in another cluster. But we know that in the highly constrained phases of d1RSB, we need $O(n)$ variable flips to get from one cluster to the next. This gives us the contradiction that we seek. ■

Here's an efficient algorithm for 3-SAT!

Instead, we just refute the proof.

Theorem 7.7. P $\neq$ NP.

Proof. Consider the solution space of k-SAT in the d1RSB phase for $k > 8$ as recalled in Section.5.2.1. We know that for high enough values of the clause density α, we have $O(n)$ frozen variables in almost all of the exponentially many clusters. Let us consider the situation where these clusters were generated by a purported LFP algorithm for k-SAT. However, when exponentially many solutions have been generated from distributions having the parametrization of the ENSP model, we will see the effect of conditional independencies beyond range $\text{poly}(\log n)$. Let $\alpha\beta\gamma$ be a representation of the variables in cliques α, β and γ, then given a value of β, we will see independent variation over all their possible conditional values in the variables of α and γ. If each set of such variables has scope at most $\text{poly}(\log n)$, then this means that once more than $c^{\text{poly}(\log n)}, c > 1$ many distinct solutions are generated, we have non-trivial conditional distributions conditioned upon values of β variables (this factor accounts for the possible orderings within the $\text{poly}(\log n)$ variables as well). At this point, the conditional independence ensure that we will see cross terms of the form

$$\alpha_1\beta\gamma_1 \quad \alpha_2\beta\gamma_2 \quad \alpha_1\beta\gamma_2 \quad \alpha_2\beta\gamma_1.$$

Note that since $O(n)$ variables have to be changed when jumping from one cluster to another, we may even chose our $\text{poly}(\log n)$ blocks to be in overlaps of these variables. This would mean that with a $\text{poly}(\log n)$ change in frozen variables of one cluster, we would get a solution in another cluster. But we know that in the highly constrained phases of d1RSB, we need $O(n)$ variable flips to get from one cluster to the next. This gives us the contradiction that we seek. ■

There's a flaw in line 637...

REJECT!

Similarly, we should focus on refuting ML defense *evaluations*.

My is robust.

This evaluation is unconvincing because...
Conclusion: NOT ROBUST

6 ADAPTIVE ATTACK EVALUATION

Given that our proposed defense effectively prevents existing black-box attacks, we now study whether or not it can prevent more sophisticated attacks. We find that while it is possible to degrade the effectiveness of the defense, we can not defeat it completely. We study three categories of attacks: gradient attacks (specifically, variants of NES with various kinds of query blinding), boundary-following attacks (specifically, variants of the boundary attack with query blinding), and hybrid attacks (which combine gradient attacks with a surrogate model).

6.1 The NES Attack

We use the NES attack as one starting point for attacking our scheme. To generate a targeted adversarial example a given input x, NES generates an adversarial example by seeding it with an image x' of the target class (i.e., that is already adversarial). NES then uses projected gradient descent to slowly reduce the distortion between this image (which is already the target class) and the original example x until it is within ϵ of the original image, while still being classified as the target class.

In order to estimate the gradient at any given location x, NES uses finite differences on a random Gaussian basis. Full details can be found in [22], but simplified, the gradient is estimated by: (1) sampling n instances of Gaussian noise $\delta_1, \ldots, \delta_n \sim \mathcal{N}(0,1)$ and adding them each to x as $\theta_i = x + \sigma\delta_i$ to generate n basis points, (2) for each basis point θ_i, estimating the confidence scores at θ_i, (3) estimating the gradient at x using these estimated confidence scores and then taking a step in the direction of the estimated gradient. The confidence score at θ_i is estimated by querying the labels for s points near θ_i chosen randomly from a sampling ball of l_∞ radius μ and computing the proportion of each class as the estimate for that class's confidence score. The default attack parameters for NES are $\sigma = 0.001$, $n = 4$, $s = 50$, $\mu = 0.001$, and learning rate = 0.01 [22]; we consider below how to adjust them.

6.1.2 Query Blinding. The second natural modification is for an attacker to use query blinding to transform each query, as described previously. In particular, we modify the confidence score estimation procedure (step 2 from the previous attack description) to sample s points using the different transformations listed in Section 5.1 instead of sampling from a ℓ_∞ ball of uniform radius. The parameters for each transformation are normalized so that the expected ℓ_2 distortion from each transformation is equal to 2.32; exact values are given in Appendix B. We selected a constant of 2.32 to match the ℓ_2 distortion of setting $\mu = 0.064$. (Note that μ is now only applicable when using the original strategy of sampling from a ball of ℓ_∞ radius μ). We used the same parameters when training the similarity encoder.

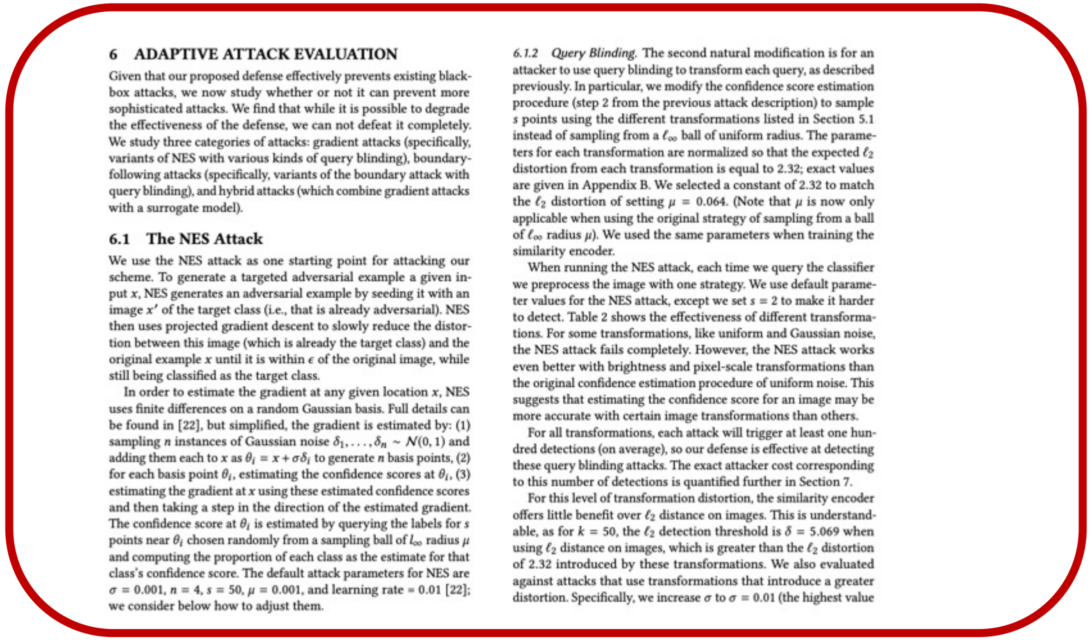
When running the NES attack, each time we query the classifier we preprocess the image with one strategy. We use default parameter values for the NES attack, except we set $s = 2$ to make it harder to detect. Table 2 shows the effectiveness of different transformations. For some transformations, like uniform and Gaussian noise, the NES attack fails completely. However, the NES attack works even better with brightness and pixel-scale transformations than the original confidence estimation procedure of uniform noise. This suggests that estimating the confidence score for an image may be more accurate with certain image transformations than others.

For all transformations, each attack will trigger at least one hundred detections (on average), so our defense is effective at detecting these query blinding attacks. The exact attacker cost corresponding to this number of detections is quantified further in Section 7.

For this level of transformation distortion, the similarity encoder offers little benefit over ℓ_2 distance on images. This is understandable, as for $k = 50$, the ℓ_2 detection threshold is $\delta = 5.069$ when using ℓ_2 distance on images, which is greater than the ℓ_2 distortion of 2.32 introduced by these transformations. We also evaluated against attacks that use transformations that introduce a greater distortion. Specifically, we increase σ to $\sigma = 0.01$ (the highest value

What makes an evaluation *convincing?*

My is robust.



6 ADAPTIVE ATTACK EVALUATION

Given that our proposed defense effectively prevents existing black-box attacks, we now study whether or not it can prevent more sophisticated attacks. We find that while it is possible to degrade the effectiveness of the defense, we can not defeat it completely. We study three categories of attacks: gradient attacks (specifically, variants of NES with various kinds of query blinding), boundary-following attacks (specifically, variants of the boundary attack with query blinding), and hybrid attacks (which combine gradient attacks with a surrogate model).

6.1 The NES Attack

We use the NES attack as one starting point for attacking our scheme. To generate a targeted adversarial example a given input x, NES generates an adversarial example by seeding it with an image x' of the target class (i.e., that is already adversarial). NES then uses projected gradient descent to slowly reduce the distortion between this image (which is already the target class) and the original example x until it is within ϵ of the original image, while still being classified as the target class.

In order to estimate the gradient at any given location x, NES uses finite differences on a random Gaussian basis. Full details can be found in [22], but simplified, the gradient is estimated by: (1) sampling n instances of Gaussian noise $\delta_1, \ldots, \delta_n \sim \mathcal{N}(0, 1)$ and adding them each to x as $\theta_i = x + \sigma\delta_i$ to generate n basis points, (2) for each basis point θ_i, estimating the confidence scores at θ_i, (3) estimating the gradient at x using these estimated confidence scores and then taking a step in the direction of the estimated gradient. The confidence score at θ_i is estimated by querying the labels for s points near θ_i chosen randomly from a sampling ball of l_∞ radius μ and computing the proportion of each class as the estimate for that class's confidence score. The default attack parameters for NES are $\sigma = 0.001$, $n = 4$, $s = 50$, $\mu = 0.001$, and learning rate = 0.01 [22]; we consider below how to adjust them.

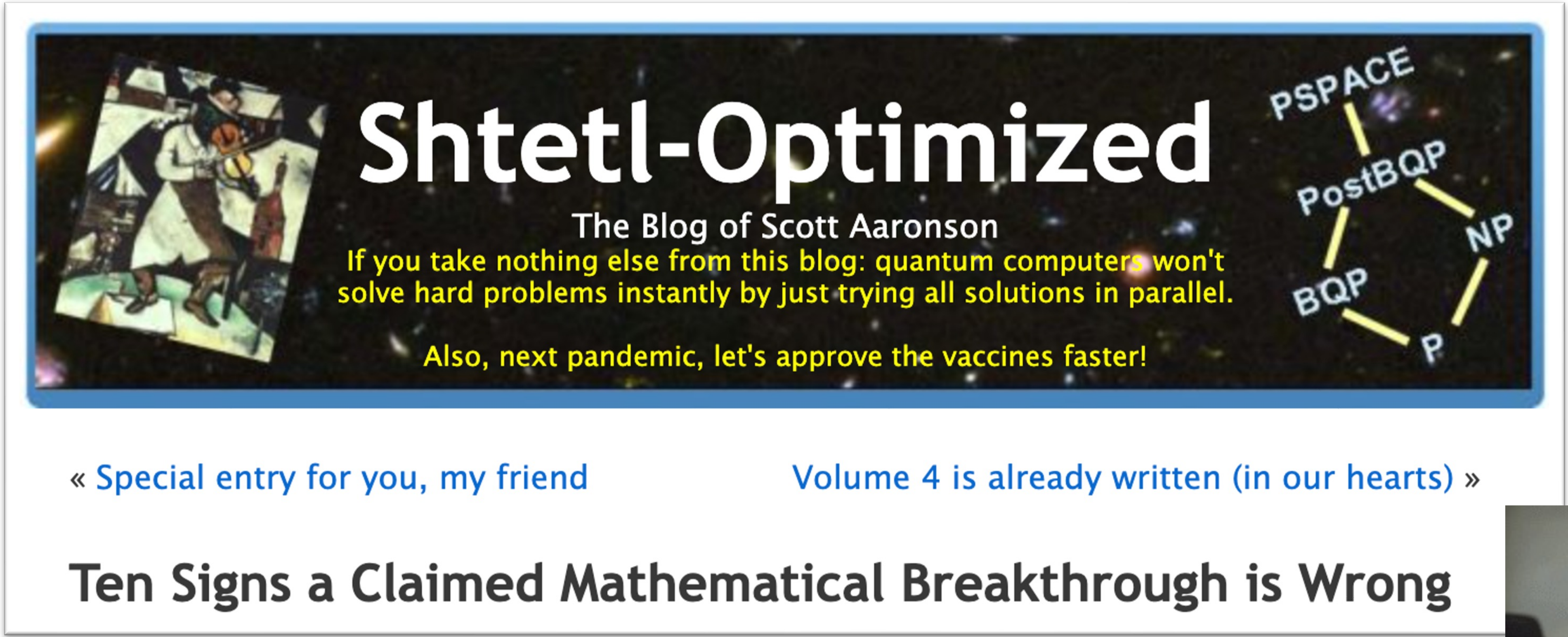
6.1.2 Query Blinding. The second natural modification is for an attacker to use query blinding to transform each query, as described previously. In particular, we modify the confidence score estimation procedure (step 2 from the previous attack description) to sample s points using the different transformations listed in Section 5.1 instead of sampling from a ℓ_∞ ball of uniform radius. The parameters for each transformation are normalized so that the expected ℓ_2 distortion from each transformation is equal to 2.32; exact values are given in Appendix B. We selected a constant of 2.32 to match the ℓ_2 distortion of setting $\mu = 0.064$. (Note that μ is now only applicable when using the original strategy of sampling from a ball of ℓ_∞ radius μ). We used the same parameters when training the similarity encoder.

When running the NES attack, each time we query the classifier we preprocess the image with one strategy. We use default parameter values for the NES attack, except we set $s = 2$ to make it harder to detect. Table 2 shows the effectiveness of different transformations. For some transformations, like uniform and Gaussian noise, the NES attack fails completely. However, the NES attack works even better with brightness and pixel-scale transformations than the original confidence estimation procedure of uniform noise. This suggests that estimating the confidence score for an image may be more accurate with certain image transformations than others.

For all transformations, each attack will trigger at least one hundred detections (on average), so our defense is effective at detecting these query blinding attacks. The exact attacker cost corresponding to this number of detections is quantified further in Section 7.

For this level of transformation distortion, the similarity encoder offers little benefit over ℓ_2 distance on images. This is understandable, as for $k = 50$, the ℓ_2 detection threshold is $\delta = 5.069$ when using ℓ_2 distance on images, which is greater than the ℓ_2 distortion of 2.32 introduced by these transformations. We also evaluated against attacks that use transformations that introduce a greater distortion. Specifically, we increase σ to $\sigma = 0.01$ (the highest value

What makes a *proof* convincing?



Ten Signs a Claimed Mathematical Breakthrough is Wrong

1. The authors don’t use TeX.
2. The authors don’t understand the question.
3. The approach seems to yield something much stronger and maybe even false.
4. The approach conflicts with a known impossibility result.
5. The authors themselves switch to weasel words by the end.
6. The paper jumps into technicalities without presenting a new idea.
7. The paper doesn’t build on (or in some cases even refer to) any previous work.
8. The paper wastes lots of space on standard material.
9. The paper waxes poetic about “practical consequences”.
10. The techniques just seem too wimpy for the problem at hand.

Ten Signs a Claimed Mathematical Breakthrough is Wrong

1. The authors don't use TeX.

Some of these don't really apply to ML...

Theorem 1.1.1.
For every algorithm ω, which solves the OWMF(F) problem, the computational complexity of ω is at least

$$floor\ (floor(2^{(q-1)/3} / (2^{2/3} + 2^{1/3} + 1)) / 2).$$

Proof:
Ad absurdum, suppose there is an algorithm ω which solves the OWMF(F) problem, such that, the computational complexity of ω is less than

$$\text{floor (floor}(2^{(q-1)/3} / (2^{2/3} + 2^{1/3} + 1)) / 2).$$

By Lemma 1.1.6.

$$|\,\text{PossibleOnOne}(\alpha_0)\,| \geq \text{floor}(2^{(q-1)/3} / (2^{2/3} + 2^{1/3} + 1)),$$

and, by definition

$$|\,\text{AllPossible}\,| = |\,\text{PossibleOnOne}(\alpha_0)\,|$$

or

$$|\,\text{AllPossible}\,| = \text{floor}(\,|\,\text{PossibleOnOne}(\alpha_0)\,|\ / 2).$$

Therefore, a priori, there is at least one element n in AllPossible such that, n is not checked by the algorithm ω and no operation is performed instead.
The algorithm ω solves the OWMF(F) problem and as a result, for the unchecked n, it has been decided if n is a solution to the problem or not.
Thus, by Lemma 1.1.1. (if n is a solution) or by Lemma 1.1.7. (if n is not a solution),

$$(n^3, I_n - b_n) \neq 1,$$

and/or the divisibility of

$$((I_n - b_n)\, n^{\#} - 1) \text{ by } n,$$

have been decided (by the algorithm ω) without performing any operation.
By Lemma 1.1.9. and/or by Lemma 1.1.10. such a decision is impossible without performing at least one operation.
As a result, the supposition that there is an algorithm ω which solves the OWMF(F) problem, such that, the computational complexity of ω is less than

$$\text{floor (floor}(2^{(q-1)/3} / (2^{2/3} + 2^{1/3} + 1)) / 2),$$

is false.

~~Ten~~ Four Signs a Claimed ~~Mathematical~~ Robustness Breakthrough is Wrong

1. There is no proof.

CTRL + F *adaptive*

4 NON-ADAPTIVE EVALUATION

Having described our defense proposal, we begin by demonstrating that it has at least some potential utility: it effectively detects existing (unmodified) black-box query attacks. While there are many black-box (hard-label) attacks, they fall roughly into two categories:

- **Gradient estimation** attacks at their core operate like standard white-box gradient-based attacks (as described in Section 2.1). However, because they do not have access to the gradient, these types of attacks instead estimate the gradient by repeatedly querying the model.
- **Boundary following** attacks, in contrast, first identify the decision boundary of the neural network, at a potentially far-away point, and then take steps following the boundary to locate the nearest point on the boundary to the target image.

We evaluate against one representative attack from each category.

4.1 Attack Setup

For each attack studied, we use the *targeted* variant, where the adversary generates an adversarial example chosen so that the resulting adversarial example x is classified as a target class t and is within a distance ϵ of an original image x. The original image and target class are chosen randomly. We call an attack successful if the ℓ_∞ distortion is below $\varepsilon = 0.05$. While most white-box work on CIFAR-10 considers the smaller distortion bound of $\varepsilon = 0.031 \approx 8/255$, we choose this slightly larger distortion because black-box attacks are known to be more difficult to generate and so we give the adversary slightly more power to compensate.

NES [22] is one of the two most prominent gradient-estimation attacks (along with SPSA [39]). It estimates the gradient at a point by averaging the confidence scores of randomly sampled nearby points, and then uses projected gradient descent [30] to perturb an image of the target class until it is sufficiently close to the original image. In the hard label case, the confidence score for a point is approximated by taking a Monte Carlo sample of nearby points, and then computing the score for a class as the fraction of nearby points with that class.

"sanity checks that the theorem isn't completely wrong…"

6 ADAPTIVE ATTACK EVALUATION

Given that our proposed defense effectively prevents existing black-box attacks, we now study whether or not it can prevent more sophisticated attacks. We find that while it is possible to degrade the effectiveness of the defense, we can not defeat it completely. We study three categories of attacks: gradient attacks (specifically, variants of NES with various kinds of query blinding), boundary-following attacks (specifically, variants of the boundary attack with query blinding), and hybrid attacks (which combine gradient attacks with a surrogate model).

"actual proof!"

this paper's adaptive evaluation is actually **6x longer** than its non-adaptive evaluation ☺

Four Robustness

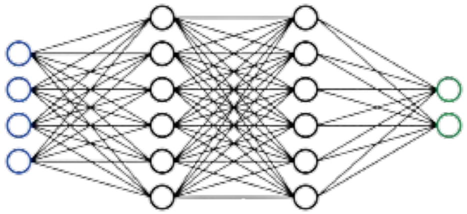
~~Ten~~ Signs a Claimed ~~Mathematical~~ Breakthrough is Wrong

1. There is no proof.

Proof. *The proof will be released upon paper acceptance.* ∎

CTRL + F *code*

reproducible evaluation



pretrained models

Four Robustness

~~Ten~~ Signs a Claimed ~~Mathematical~~ Breakthrough is Wrong

2. There are many proofs.

We first present multiple adaptive attacks

various strong adaptive attacks

evaluate 7 potential adaptive attacks.

a variety of strong adaptive attacks,

A strong evaluation should be about *quality*, not *quantity*

Ten Signs a Claimed Mathematical Breakthrough is Wrong

3. The approach seems to yield something much stronger and maybe even false.

$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_3 \vee \neg x_4 \vee x_7)$

Theorem: 3-SAT $\notin$ P

$(x_1 \vee x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_3 \vee x_5)$

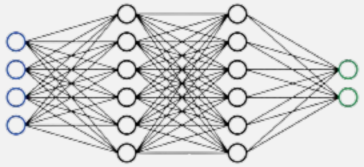
Theorem: 2-SAT $\notin$ P

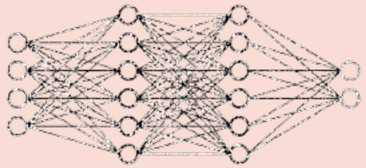
Proof

If the *proof still works*
for a *theorem that is false*,
there is clearly something wrong!

Four ~~Ten~~ Signs a Claimed Robustness ~~Mathematical~~ Breakthrough is Wrong

3. The approach seems to yield something much stronger and maybe even false.

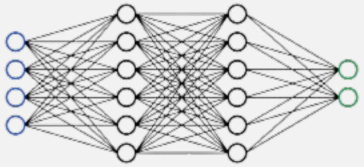
 is robust

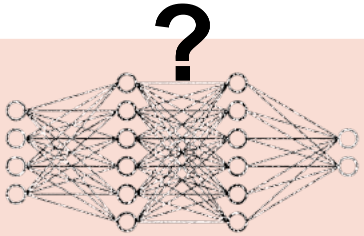
 is robust

Evaluation

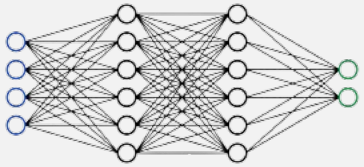
If the *evaluation still passes* *(all attacks fail)*
for a *defense that is broken*,
there is clearly something wrong!

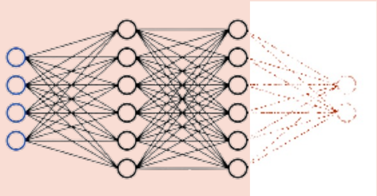
Building a minimally-altered, broken defense: the binarization test.

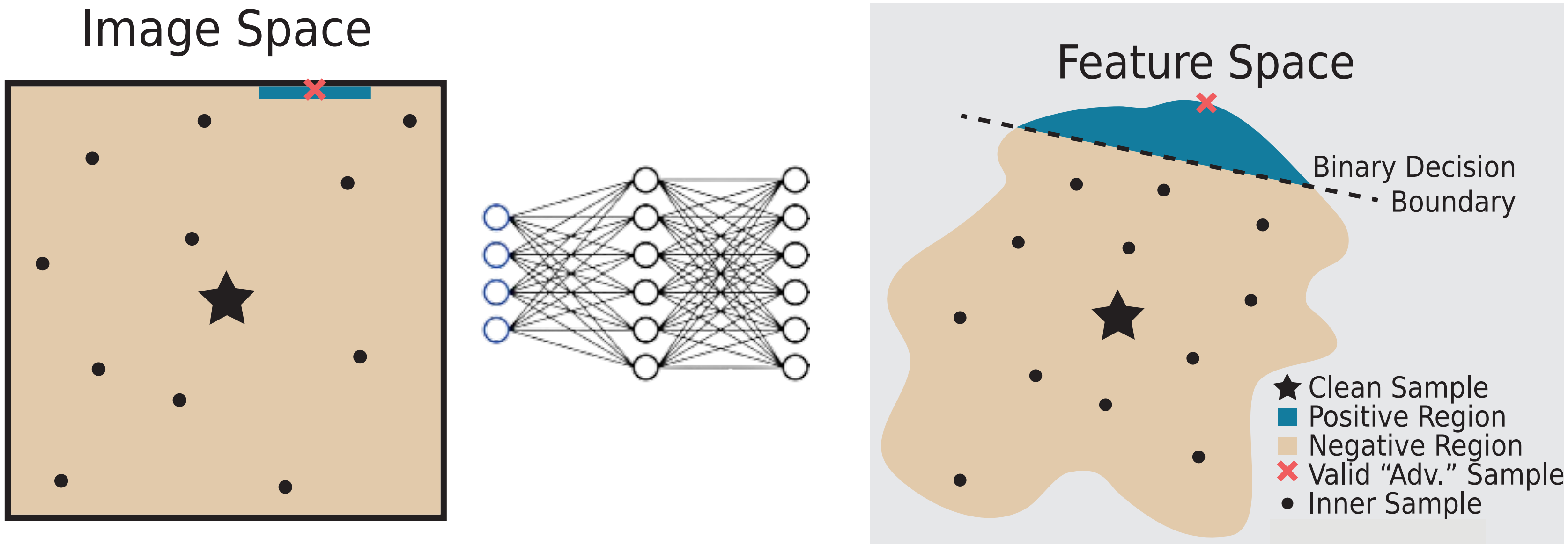
 is robust

 is robust

Building a minimally-altered, broken defense: the binarization test.

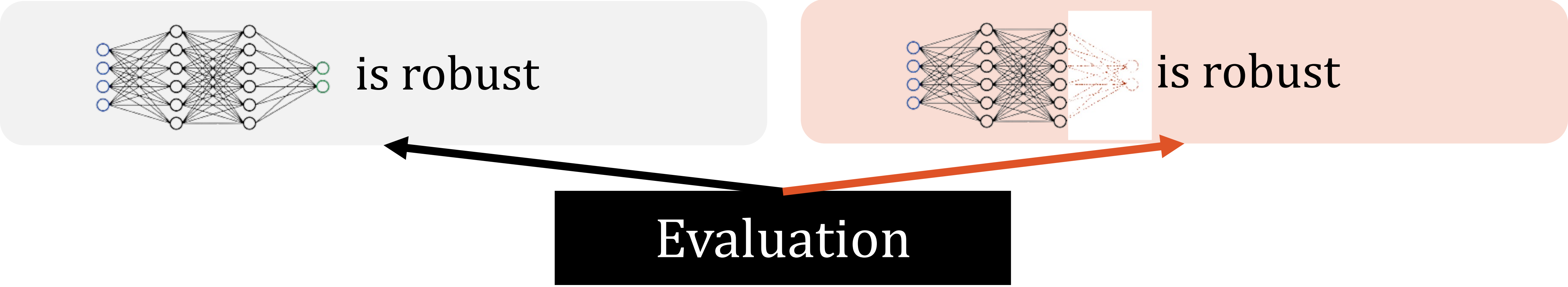

is robust


is robust



"Increasing Confidence in Adversarial Robustness Evaluations", Zimmermann et al., https://arxiv.org/abs/2206.13991

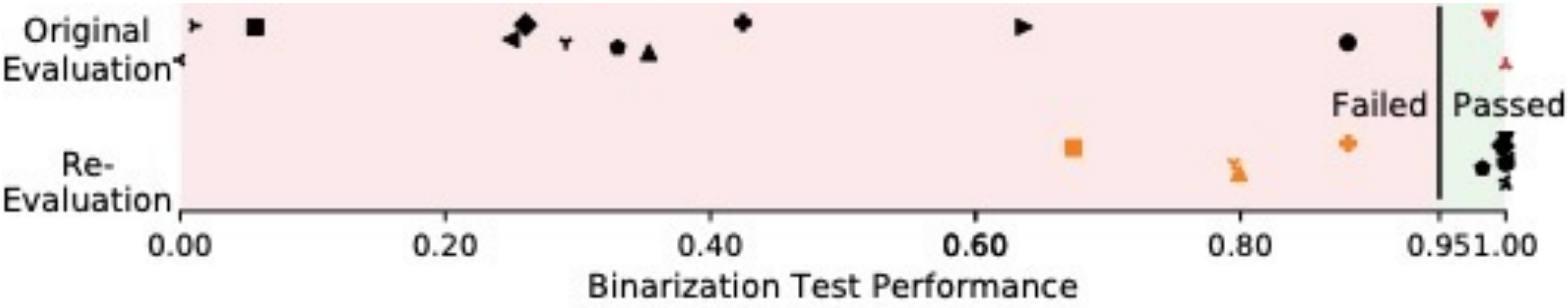
Building a minimally-altered, broken defense: the binarization test.



If the *evaluation were strong*
it would break the non-robust defense

"Increasing Confidence in Adversarial Robustness Evaluations", Zimmermann et al., https://arxiv.org/abs/2206.13991

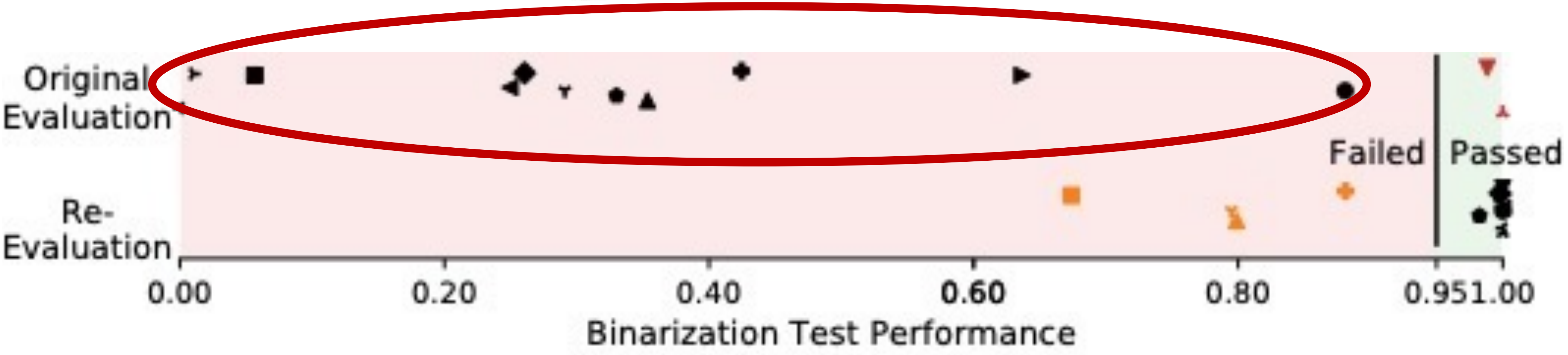
The binarization test identifies flawed evaluations.



13 defenses where the original evaluation overestimated robustness, compared to a future re-evaluation

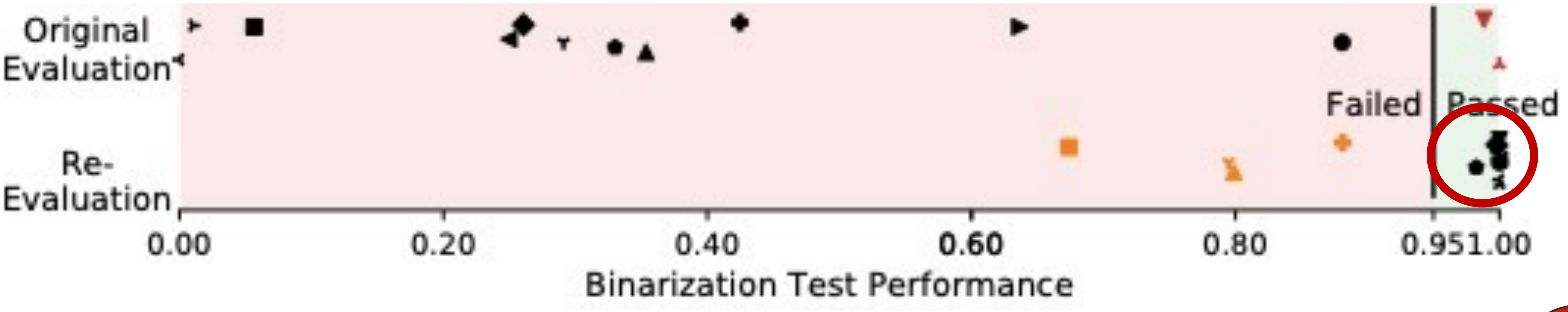
"Increasing Confidence in Adversarial Robustness Evaluations", Zimmermann et al., https://arxiv.org/abs/2206.13991

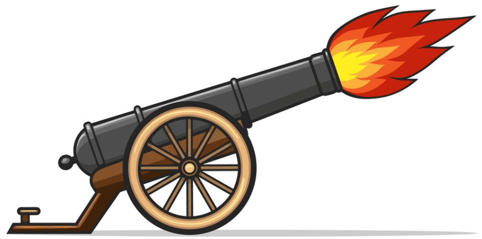
Weak evaluations fail the test.



"Increasing Confidence in Adversarial Robustness Evaluations", Zimmermann et al., https://arxiv.org/abs/2206.13991

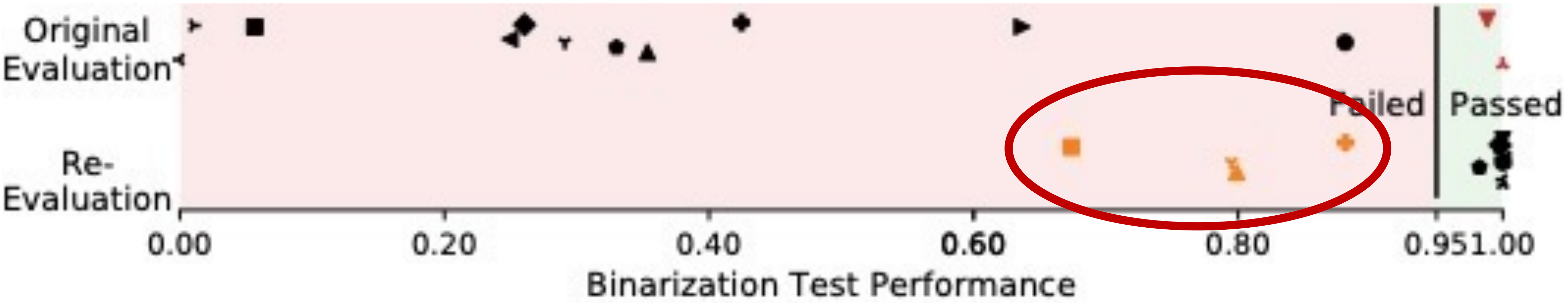
Strong adaptive evaluations (which broke the defenses) pass the test.





"Increasing Confidence in Adversarial Robustness Evaluations", Zimmermann et al., https://arxiv.org/abs/2206.13991

Some adaptive attacks break defenses but remain quite weak.



"Increasing Confidence in Adversarial Robustness Evaluations", Zimmermann et al., https://arxiv.org/abs/2206.13991

Our test can have false positives.

original evaluations are not “completely wrong”

“Increasing Confidence in Adversarial Robustness Evaluations”, Zimmermann et al., https://arxiv.org/abs/2206.13991

Four Robustness

~~Ten~~ Signs a Claimed ~~Mathematical~~ Breakthrough is Wrong

3. The evaluation fails to break a non-robust defense.

A convincing evaluation should *distinguish* robust defenses from broken ones!

Ten Signs a Claimed Mathematical Breakthrough is Wrong

4. The approach conflicts with a known impossibility result.

Theorem:
Technique X
won't help you solve
P vs NP

RELATIVIZATIONS OF THE $\mathscr{P} =? \mathscr{NP}$ QUESTION*

THEODORE BAKER†, JOHN GILL‡ AND ROBERT SOLOVAY¶

Natural Proofs

Alexander A. Razborov*

School of Mathematics, Institute for Advanced Study, Princeton, New Jersey 08540; and Steklov Mathematical Institute, Vavilova 42, 117966, GSP-1, Moscow, Russia

and

Steven Rudich†

Algebrization: A New Barrier in Complexity Theory

Scott Aaronson*
MIT

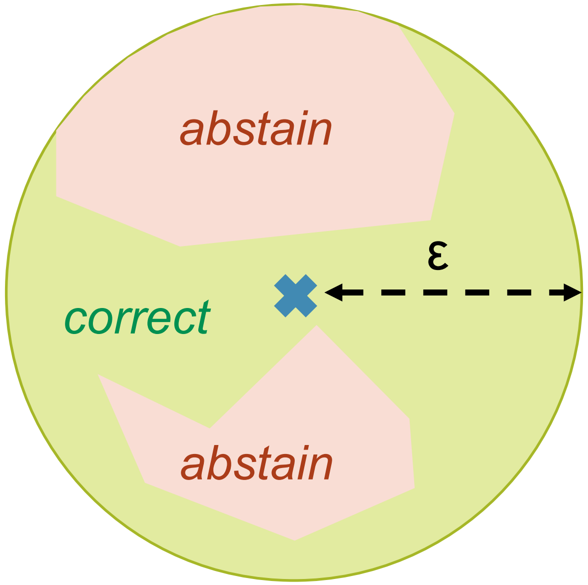
Avi Wigderson†
Institute for Advanced Study

Can we show such an *impossibility result* for adversarial ML?

Theorem:
Technique X
won't help you build
a robust model

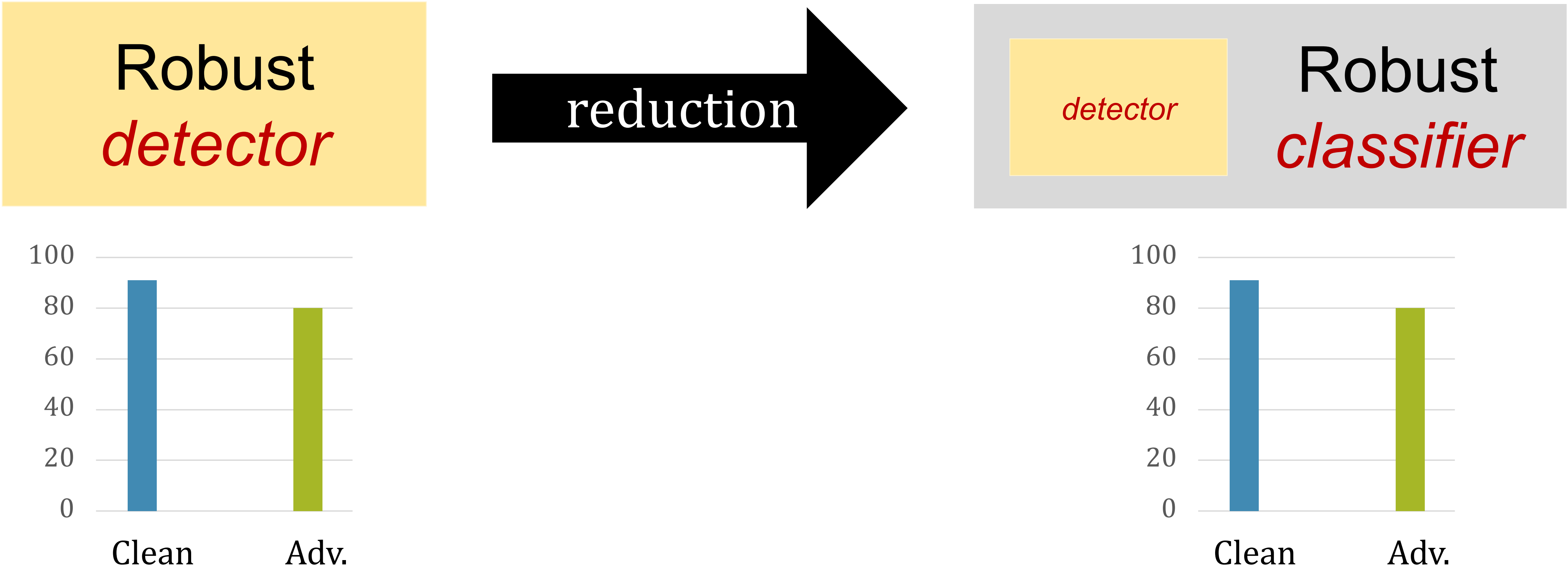
One attempt: a barrier for *detecting* adversarial examples.

Theorem:
Detecting attacks won't help you build a robust model



"Detecting Adversarial Examples Is (Nearly) As Hard As Classifying Them", ICML 2022, https://arxiv.org/abs/2107.11630

We show a reduction from robust detection to classification.



We show a partial reduction from robust detection to classification.

Robust *detector*

reduction

detector

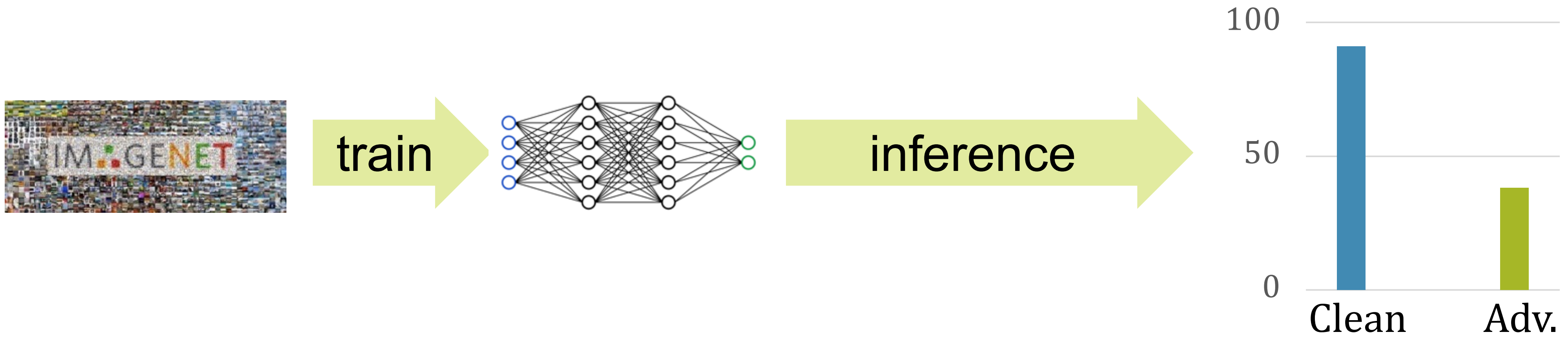
Robust *classifier*

- efficient
- robust at distance ε

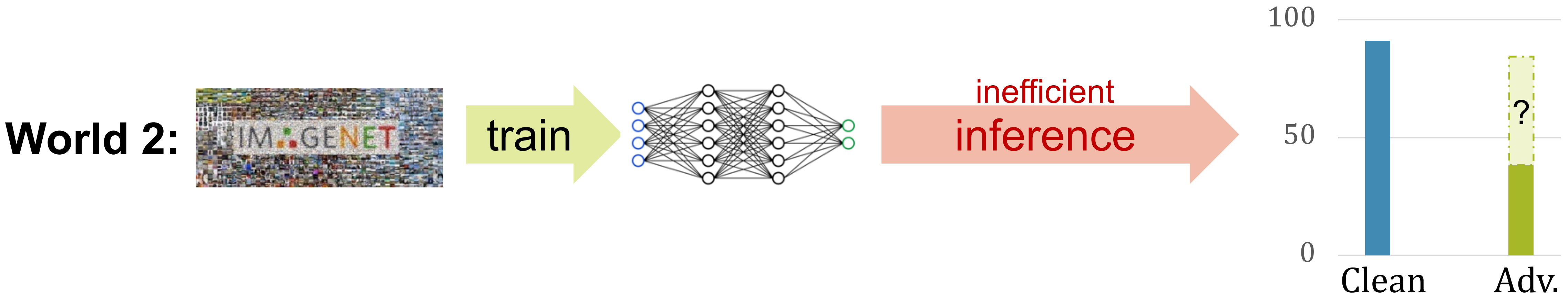
- *inefficient* *(at inference)*
- robust at distance $\varepsilon/2$

Strongly robust detectors imply a *breakthrough in robust classification.*

World 1:



Strongly robust detectors imply a *breakthrough in robust classification.*

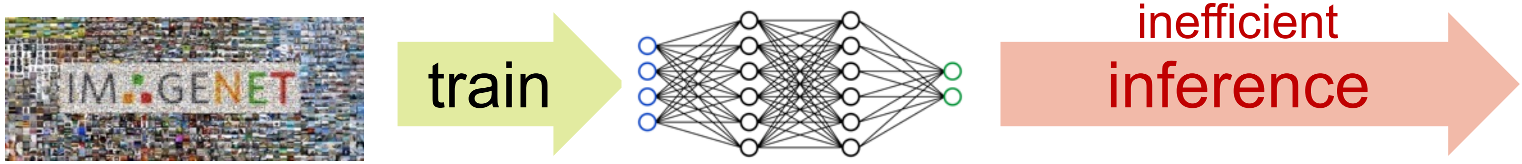


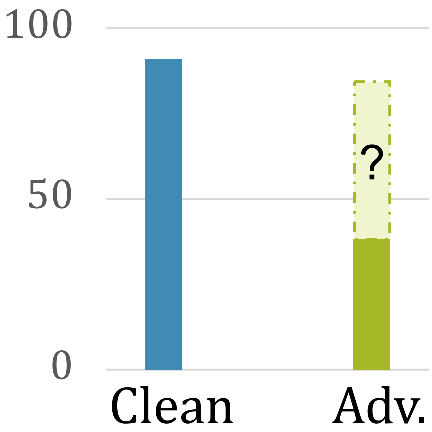
Can we build much more robust classifiers in **World 2**?

(we don't know...)

Strongly robust detectors imply a *breakthrough in robust classification.*

World 2:



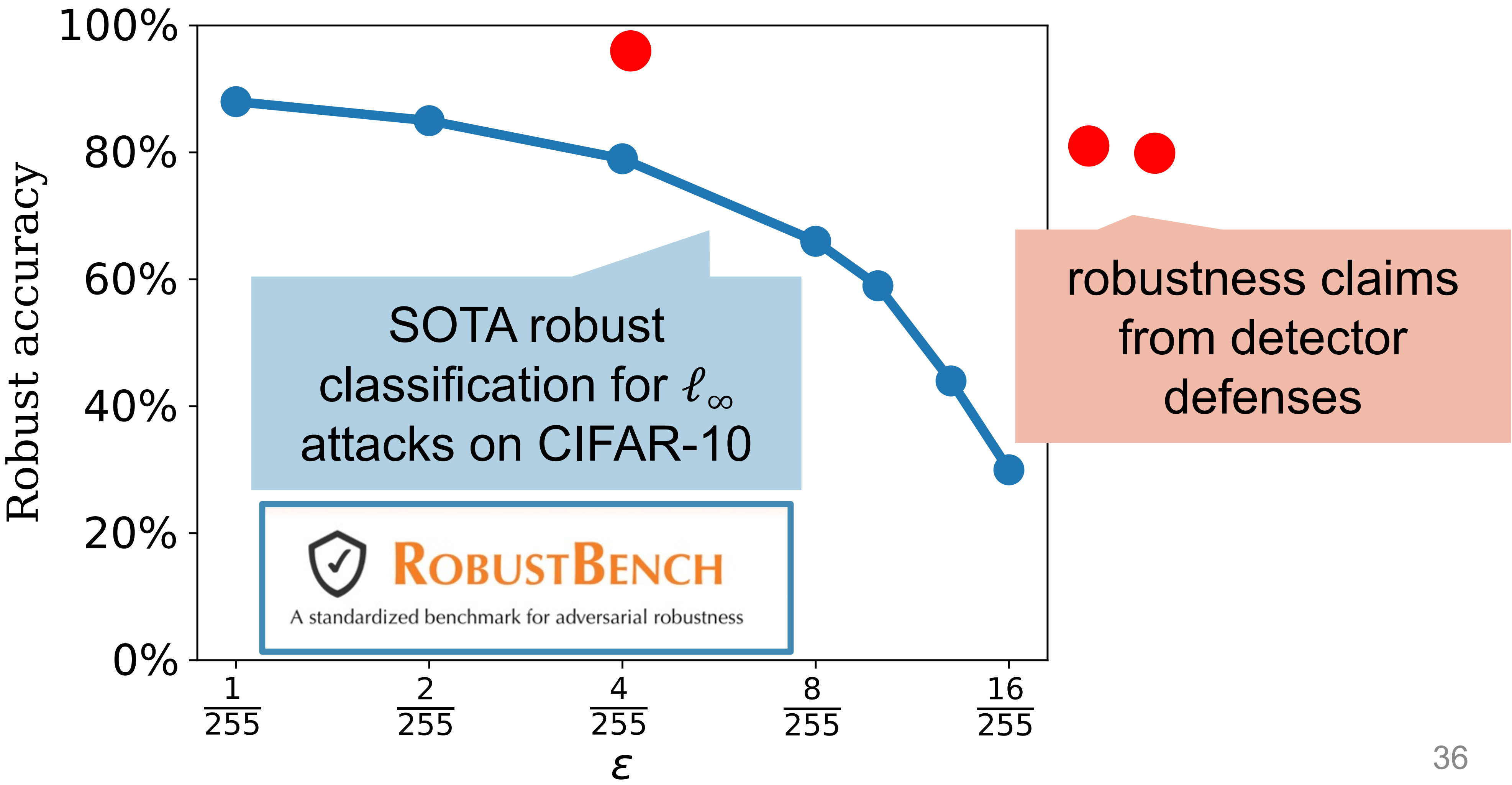


Can we build much more robust classifiers in **World 2**?

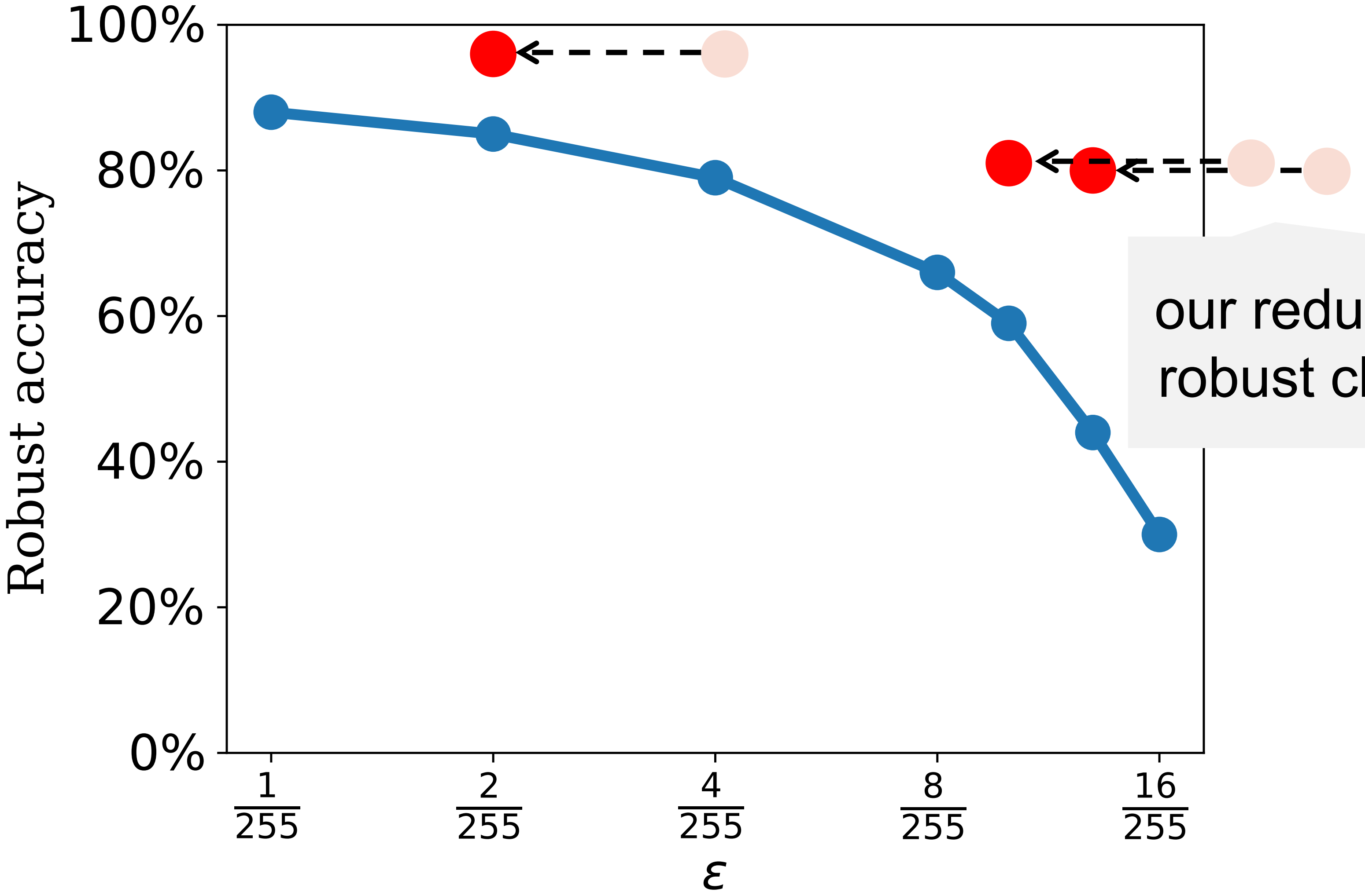
(we don't know...)

But any sufficiently robust detector implies a positive answer!

Many detectors *implicitly* claim such a breakthrough!

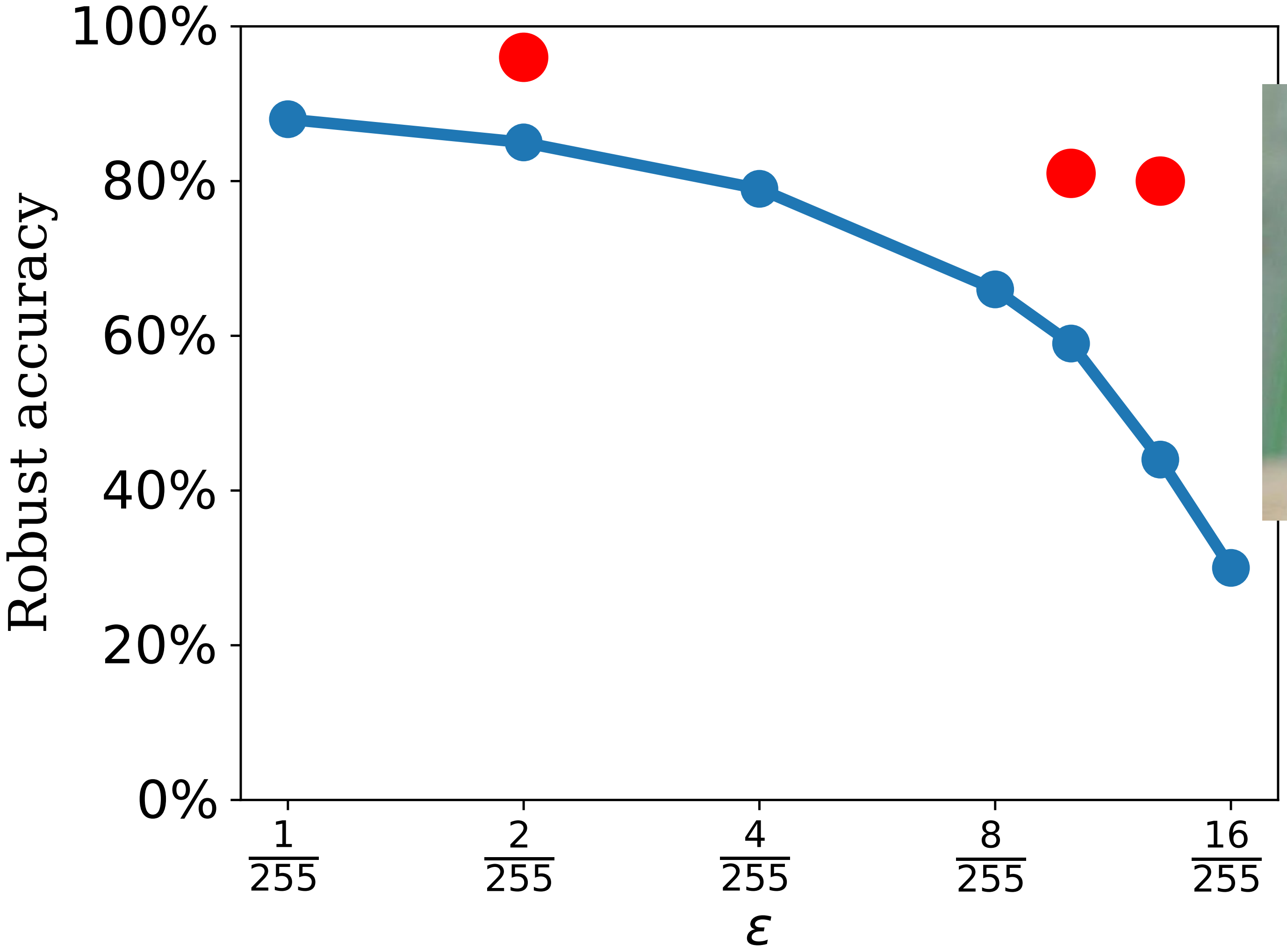


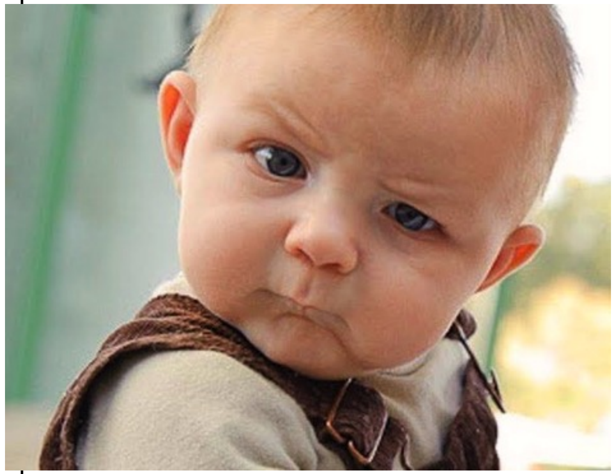
Many detectors *implicitly* claim such a breakthrough!



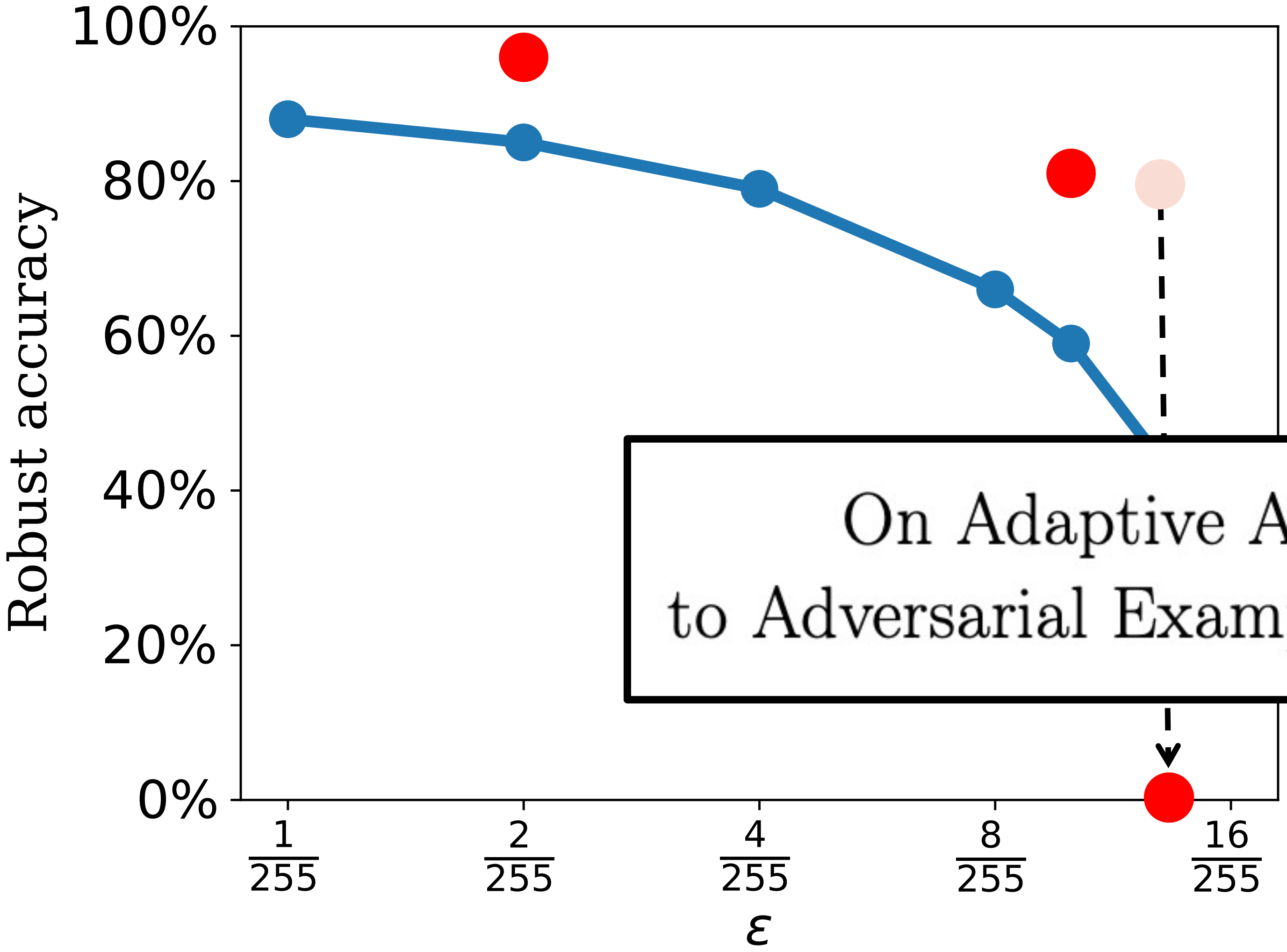
our reduction implies a robust classifier for ε/2

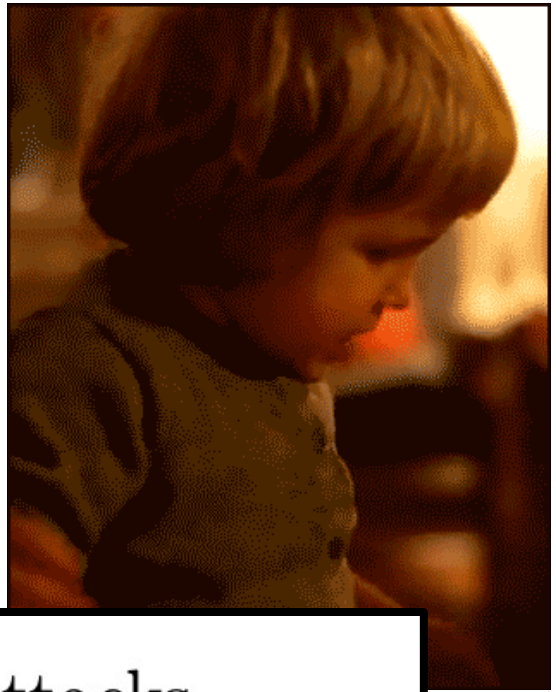
Optimistic view: this is a *breakthrough* in (inefficient) robust classification!





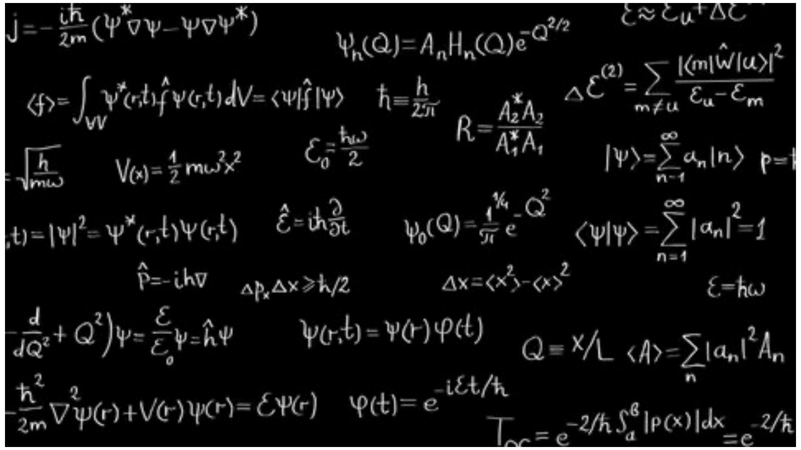
Pessimistic (*realistic?*) view:
These detectors are *not robust!*



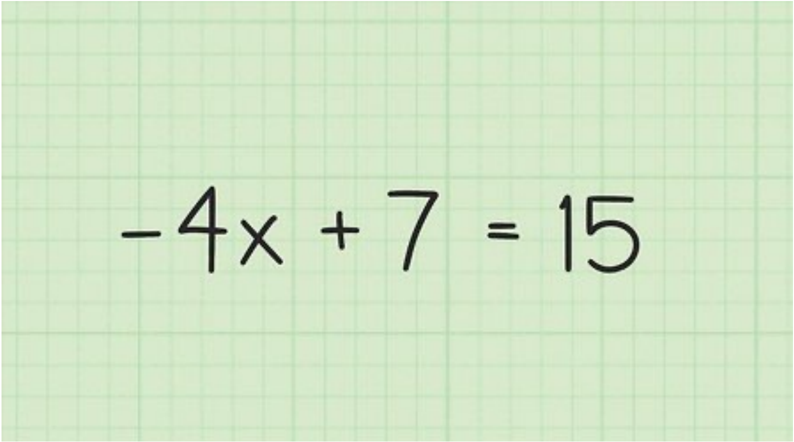


On Adaptive Attacks
to Adversarial Example Defenses

Robust detection is as hard as classification.

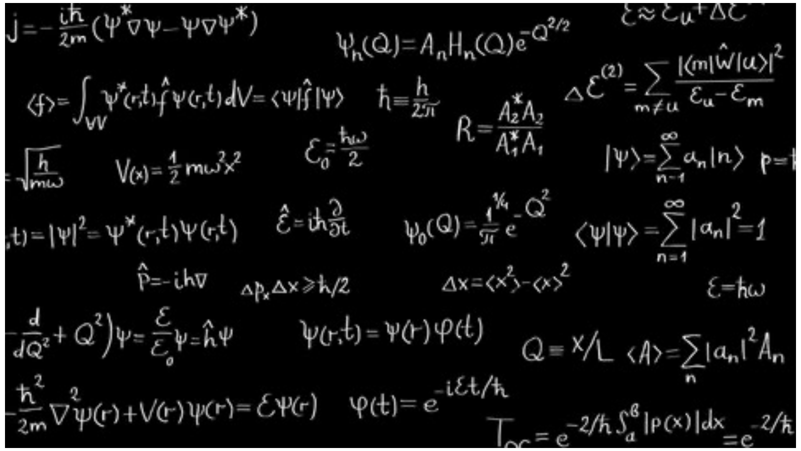


Robust classification

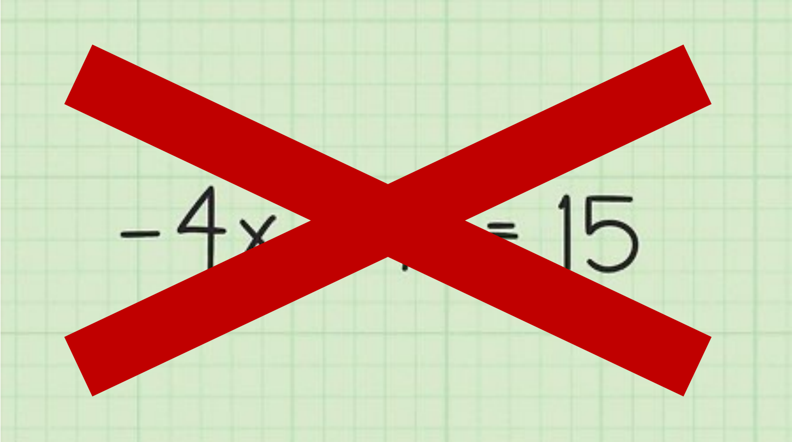


Robust detection

Robust detection is as hard as classification.



Robust classification



Robust detection

~~Ten~~ Four Signs a Claimed ~~Mathematical~~ Robustness Breakthrough is Wrong

4. Breakthrough results using only "weak" techniques.

detectors

denoisers, preprocessors

randomness

provably weak!

empirically weak

Treat your ML defense like a **theorem!**

Defense evaluations that aren't convincing are like theorems without proofs...

Four Robustness

~~Ten~~ Signs a Claimed ~~Mathematical~~ Breakthrough is Wrong

1. **There is no adaptive attack (or no code).** (no proof)
2. **There are many partial adaptive attacks.** (many proofs)
3. **The evaluation fails to break a non-robust defense.** (proof idea still holds for false theorems)
4. **Breakthrough results using only "weak" techniques.** (proof idea is believed/known to fail)