A New Defense Against Adversarial Images: Turning a Weakness Into a Strength
Tao Yu*, Shengyuan Hu*, Chuan Guo1, Wei-Lun Chao2, Killian Q. Weinberger1

1 Department of Computer Science, Cornell University.
2 Department of Computer Science and Engineering, The Ohio State University

---

**Background**

Neural Networks are prone to imperceptible changes in the input -- adversarial perturbations -- that alter the model’s decision entirely.

- **Common and inevitable:**
  - Gradient-guided search can perturb real images to any other class.
  - Any classifier has a fundamental limit on the robustness that it can achieve for adversarial examples.
- **Hard to defend against:**
  - Defenses such as adversarial training are very slow and not suitable for large datasets like ImageNet.
  - State-of-the-art defense (TRADES) achieves only 56% accuracy against strongest attack with a drop of 11% clean accuracy on CIFAR10.

---

**Vulnerability of Existing Defenses**

Attackers can easily bypass defenses by optimizing them in black-box and white-box settings.

- **Black-box:** Attackers can access model and defense structure, but not any parameter, one can use decision-based methods or natural gradients to optimize the defense criterion.
- **White-box:** Attackers have full knowledge of model and defense, one can use gradient-guided methods for differentiable functions or approximate non-differentiable functions with identity (BPDA) to optimize defense criterion.

---

**Methodology**

We propose a set of seemingly contradictory criteria to detect adversarial examples.

- **Robustness to random noise (C1):**
  - Prediction of a real image $x_0$ is robust to random noise, i.e., low density of adversarial perturbations
  - Input $x'$ can be detected effectively.
- **Existence of nearby adversarial examples (C2):**
  - Gradient-guided attacks can easily find an adversarial example of $x_0$. However, the optimization against C1 makes it hard to find an “adversarial example” of the adversarial example $x''$.
  - Adversarial example $x''$ can be detected effectively.
- **Contradictory optimization for attackers:**
  - Optimizing C1 pulls the adversarial example away from the boundary (towards $x''$).
  - Optimizing C2 pulls the adversarial example close to the boundary (towards $x'$).
  - Optimizing C1+C2 during attack leads to competing objectives!