JOHNS HOPKINS
UNIVERSITY

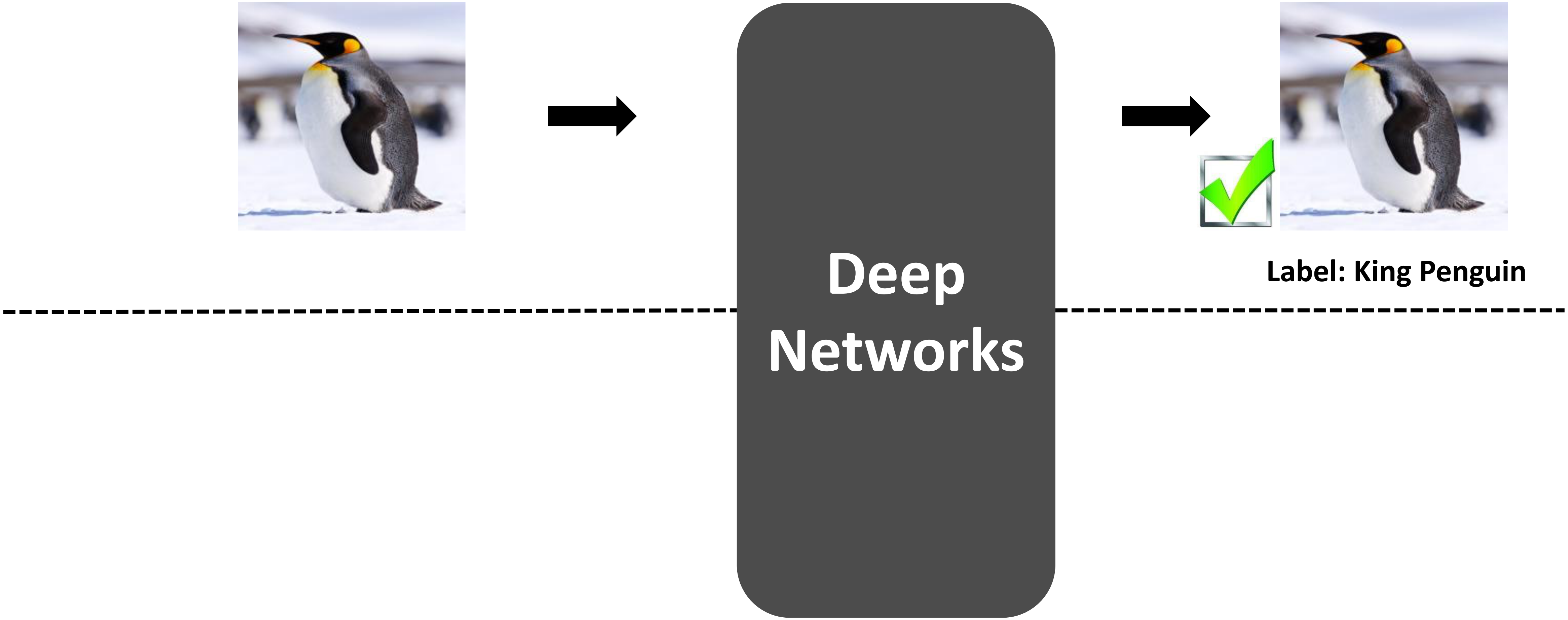



Feature Denoising for Improving Adversarial Robustness

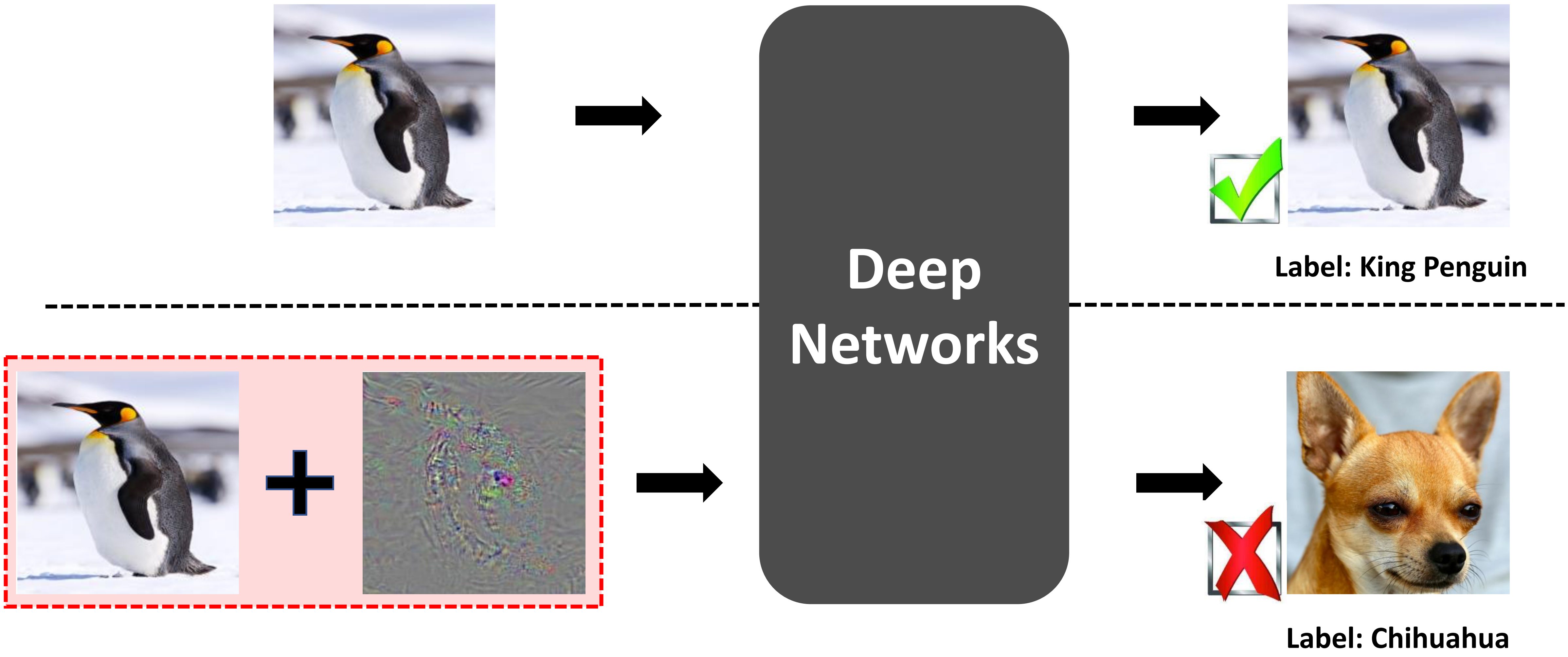
Cihang Xie
Johns Hopkins University

- **Background**
- Towards Robust Adversarial Defense

Deep networks are Good



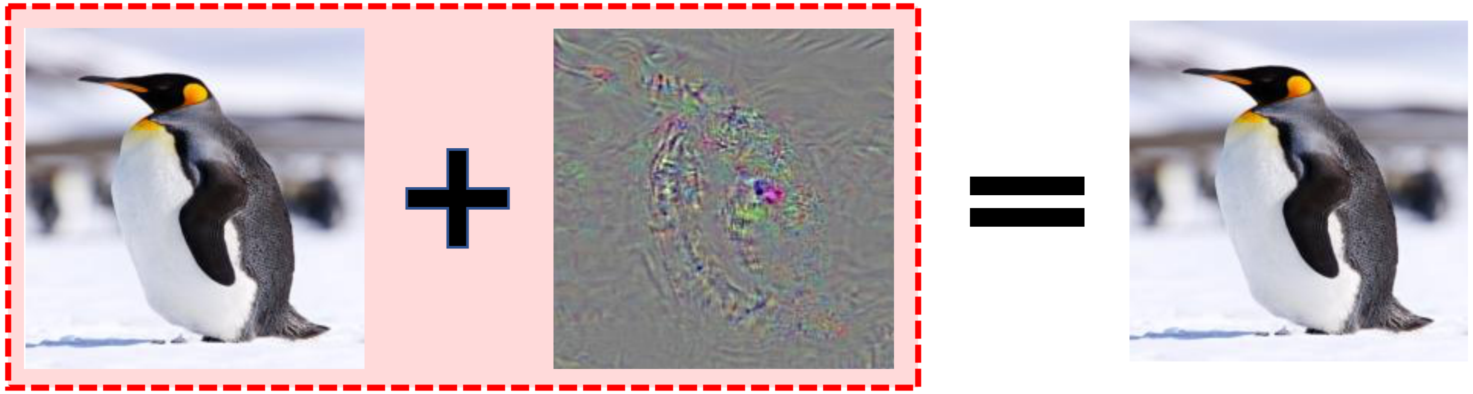
Deep networks are FRAGILE to small & carefully crafted perturbations



Deep networks are FRAGILE to small & carefully crafted perturbations

We call such images as

Adversarial Examples



Adversarial Examples can exist on Different Tasks

semantic segmentation

pose estimation

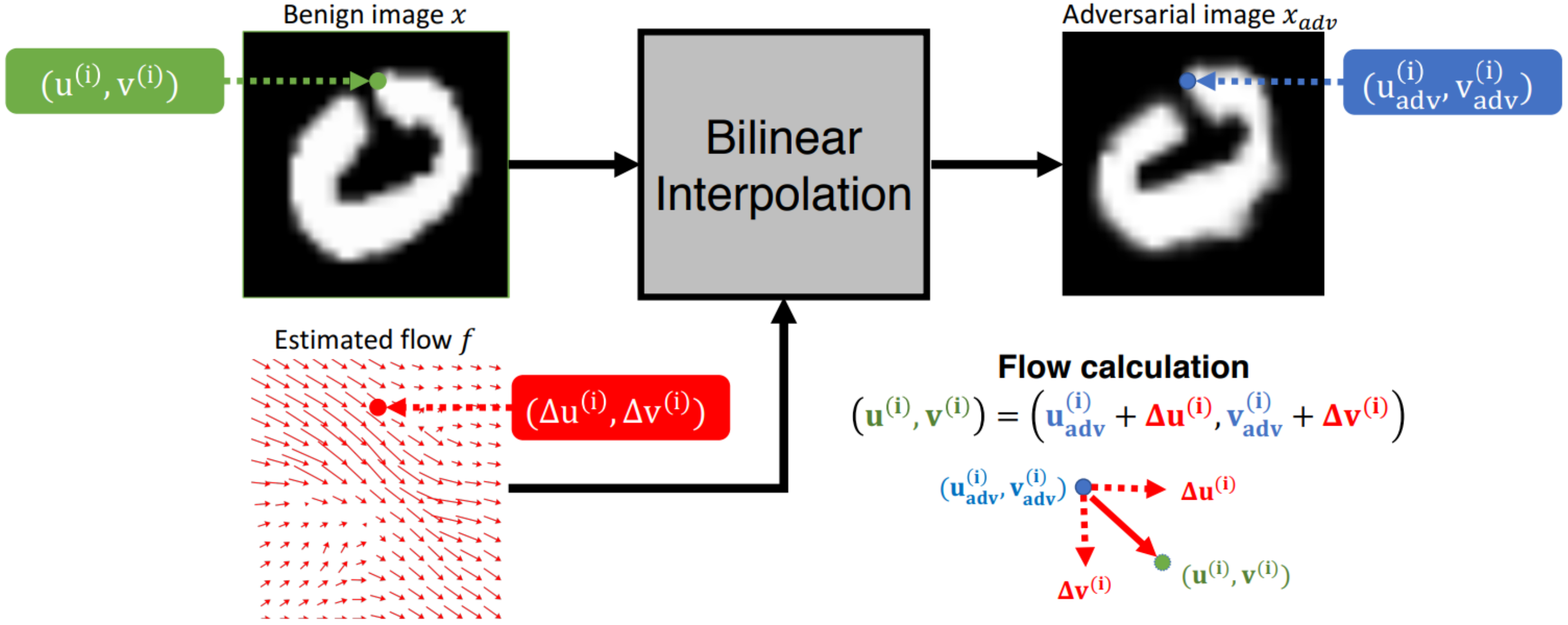
South Africa's historic Soweto township marks its 100th birthday on Tuesday in a mood of optimism.
57% **World**

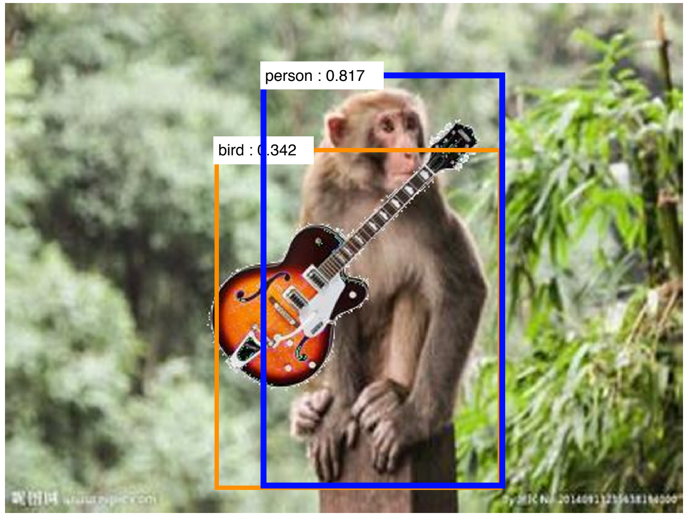
South Africa's historic Soweto township marks its 100th birthday on Tuesday in a moo**P** of optimism.
95% **Sci/Tech**

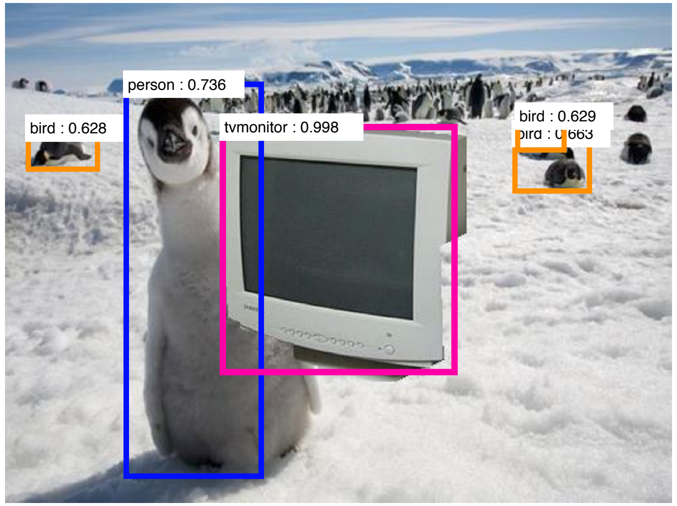
text classification

[1] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. "Adversarial examples for semantic segmentation and object detection." In *ICCV*. 2017.
[2] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. "Houdini: Fooling deep structured prediction models." In *NeurIPS.* 2018.
[3] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. "HotFlip: White-Box Adversarial Examples for Text Classification." In *ACL*. 2018.

Adversarial Examples can be created other than Adding Perturbation







[4] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. "Spatially transformed adversarial examples." In *ICLR.* 2018.

[5] Jianyu Wang, Zhishuai Zhang, Cihang Xie, et al. "Visual concepts and compositional voting." In *Annals of Mathematical Sciences and Applications.* 2018 .

Adversarial Examples can exist on The Physical World

[6] Lifeng Huang, Chengying Gao, Yuyin Zhou, Changqing Zou, Cihang Xie, Alan Yuille, Ning Liu. "UPC: Learning Universal Physical Camouflage Attacks on Object Detectors," Arxiv, 2019

Generating Adversarial Example is SIMPLE:

$$\textbf{maximize}\ \mathrm{loss}(f(x+\mathbf{r}), y^{true}; \theta)$$

$\uparrow$

Maximize the loss function w.r.t. **Adversarial Perturbation r**

Generating Adversarial Example is SIMPLE:

$$\textbf{maximize}\ \mathrm{loss}(f(x+\mathbf{r}), y^{true}; \theta)$$

↑

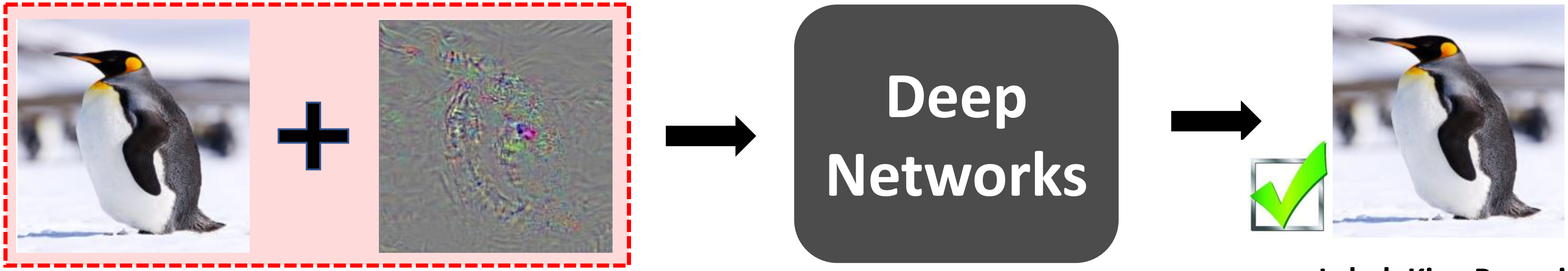
Maximize the loss function w.r.t. **Adversarial Perturbation r**

$$\textbf{minimize}\ \mathrm{loss}(f(x), y^{true}; \boldsymbol{\theta});$$

↑

Minimize the loss function w.r.t. **Network Parameters θ**

- Background
- **Towards Robust Adversarial Defense**

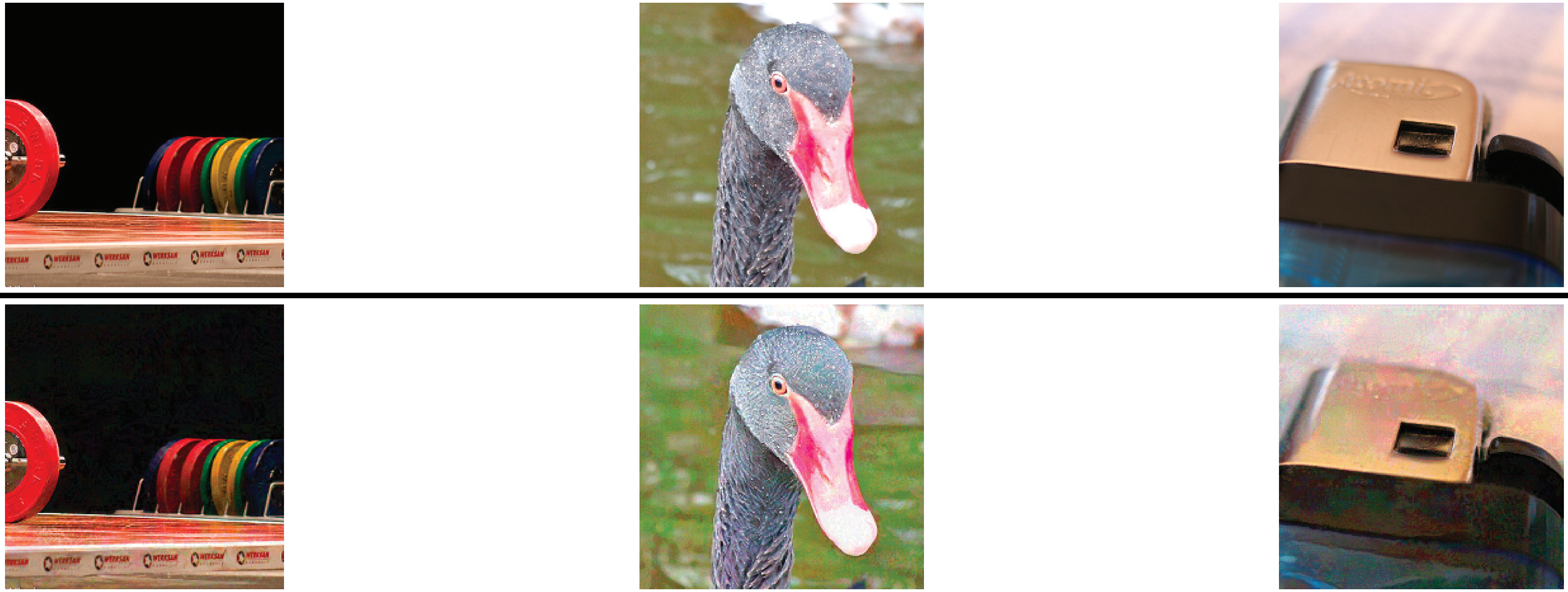


Label: King Penguin

Observation: Adversarial perturbations are SMALL on the pixel space

Clean

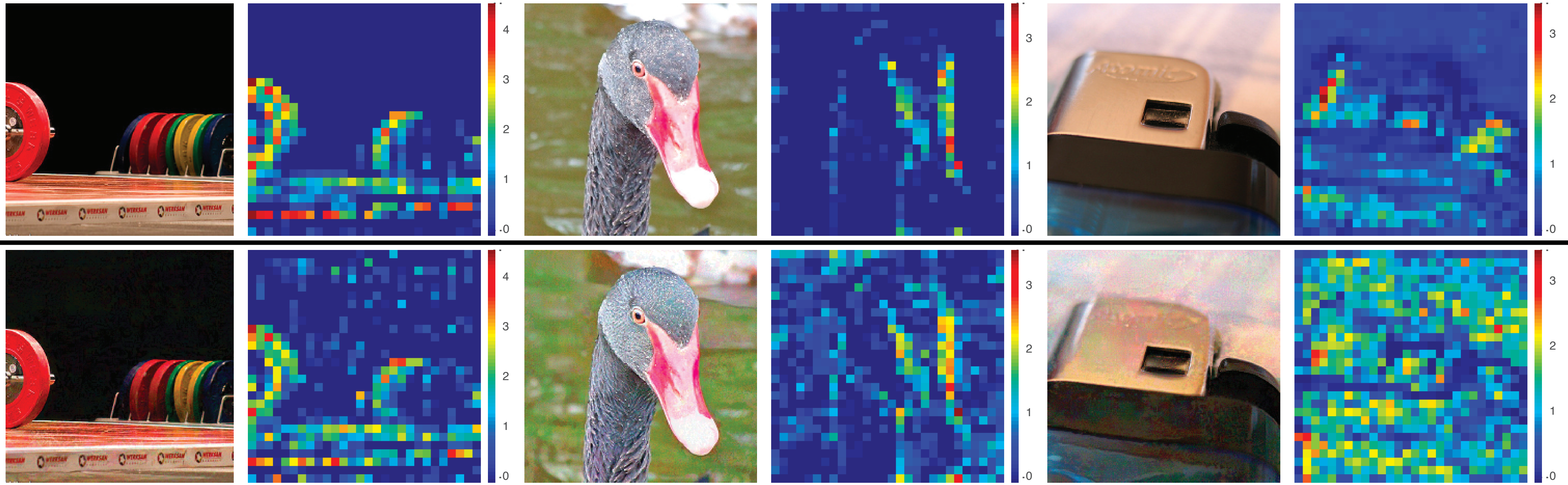
Adversarial



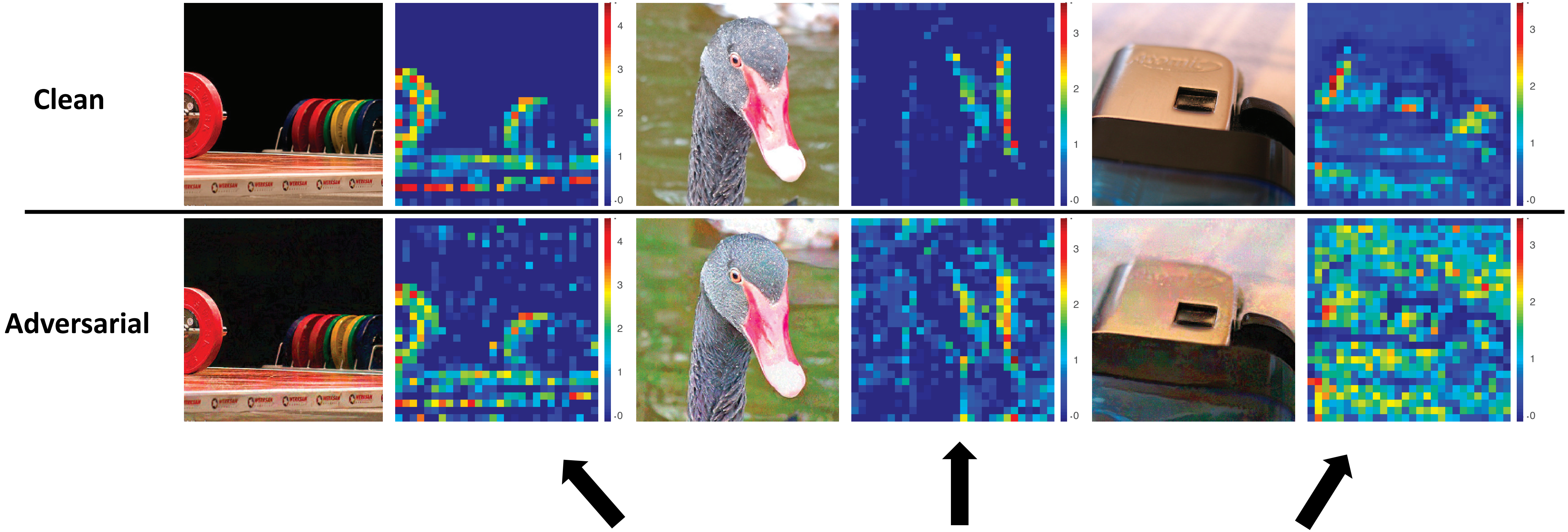
Observation: Adversarial perturbations are BIG on the feature space

Clean

Adversarial



Observation: Adversarial perturbations are BIG on the feature space



We should **DENOISE** these feature maps

Our Solution: Denoising at feature level

Traditional Image Denoising Operations:

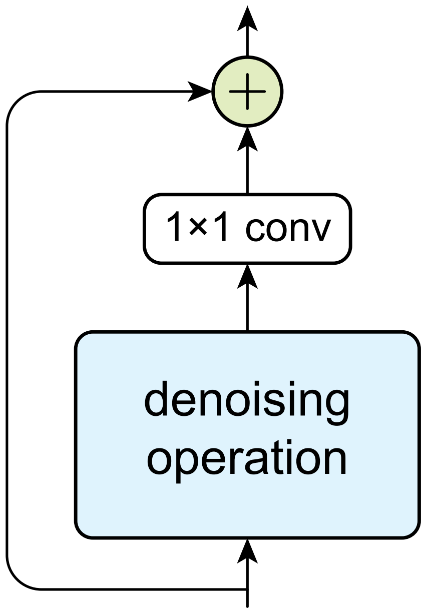
Local filters (predefine a local region $\Omega(i)$ for each pixel i):

- Bilateral filter $y_i = \frac{1}{C(x_i)} \sum_{\forall j \in \Omega(i)} f(x_i, x_j) x_j$
- Median filter $y_i = median\{\forall j \in \Omega(i): x_j\}$
- Mean filter $y_i = \frac{1}{C(x_i)} \sum_{\forall j \in \Omega(i)} x_j$

Non-local filters (the local region $\Omega(i)$ is the whole image I):

- Non-local means $y_i = \frac{1}{C(x_i)} \sum_{\forall j \in I} f(x_i, x_j) x_j$

Denoising Block Design



Denoising operations may **lose information**

- we add a **residual connection** to balance the tradeoff between removing noise and retaining original signal

Training Strategy: Adversarial training

- Core Idea: train with adversarial examples

Training Strategy: Adversarial training

- Core Idea: train with adversarial examples

$$\min_{\theta} \underbrace{\max_{\boldsymbol{r}} \ \text{loss}(f(x+r), \text{ytrue}; \theta)}_{\textbf{max step: generate adversarial perturbation}}$$

Training Strategy: Adversarial training

- Core Idea: train with adversarial examples

$$\min_{\theta} \underbrace{\max_{\boldsymbol{r}} \ \mathrm{loss}(f(x+r), \mathrm{ytrue}; \boldsymbol{\theta})}_{\text{max step: generate adversarial perturbation}}$$

max step: generate adversarial perturbation

min step: optimize network parameters

Two Ways for Evaluating Robustness

Defending Against White-box Attacks

- Attackers know everything about models
- Directly maximize loss(f(x+**r**), y^{true}; θ)

Two Ways for Evaluating Robustness

Defending Against White-box Attacks

- Attackers know everything about models
- Directly maximize loss(f(x+**r**), y^{true}; θ)

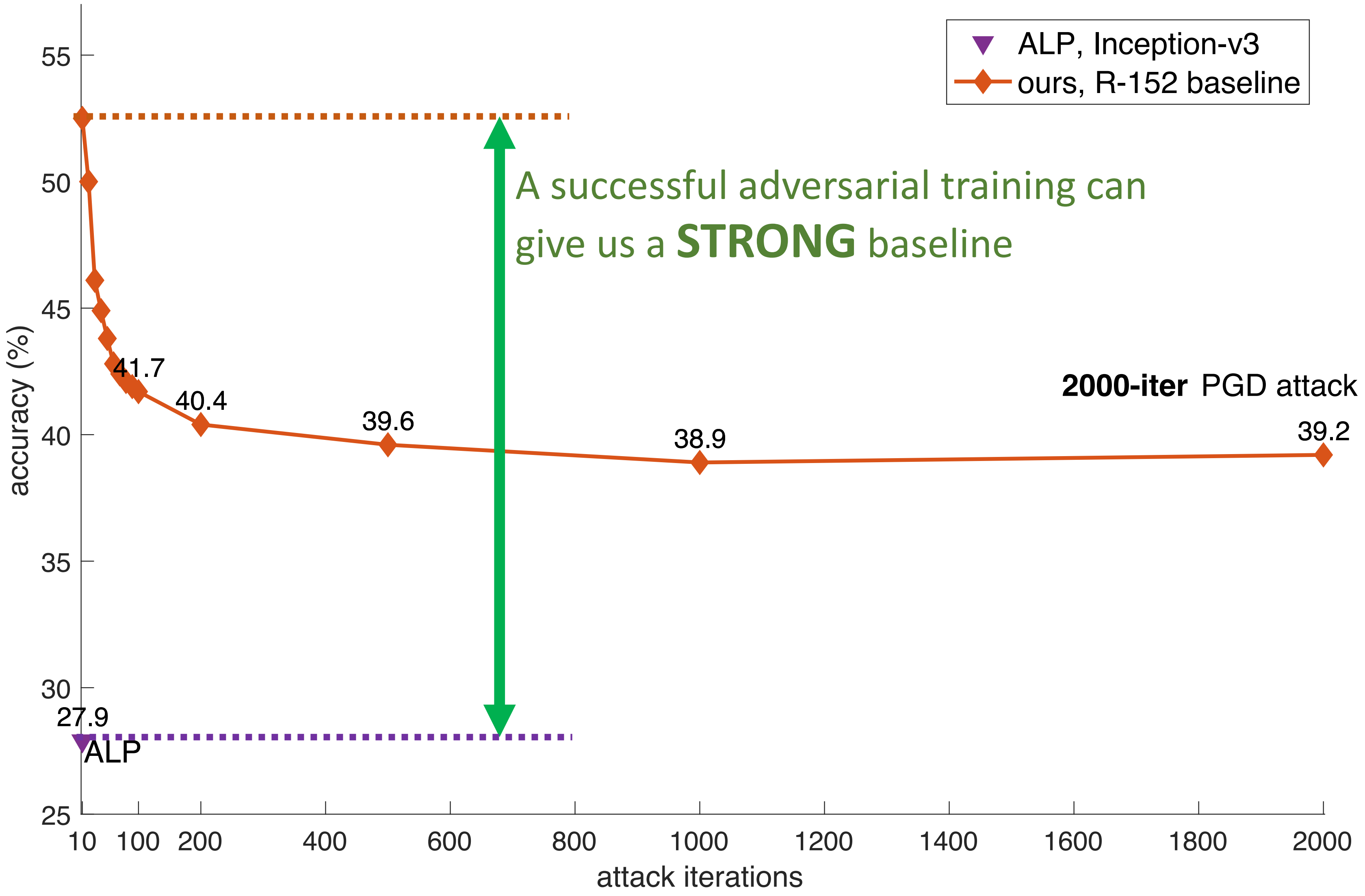
Defending Against Blind Attacks

- Attackers know nothing about models
- Attackers generate adversarial examples using substitute networks **(rely on transferability)**

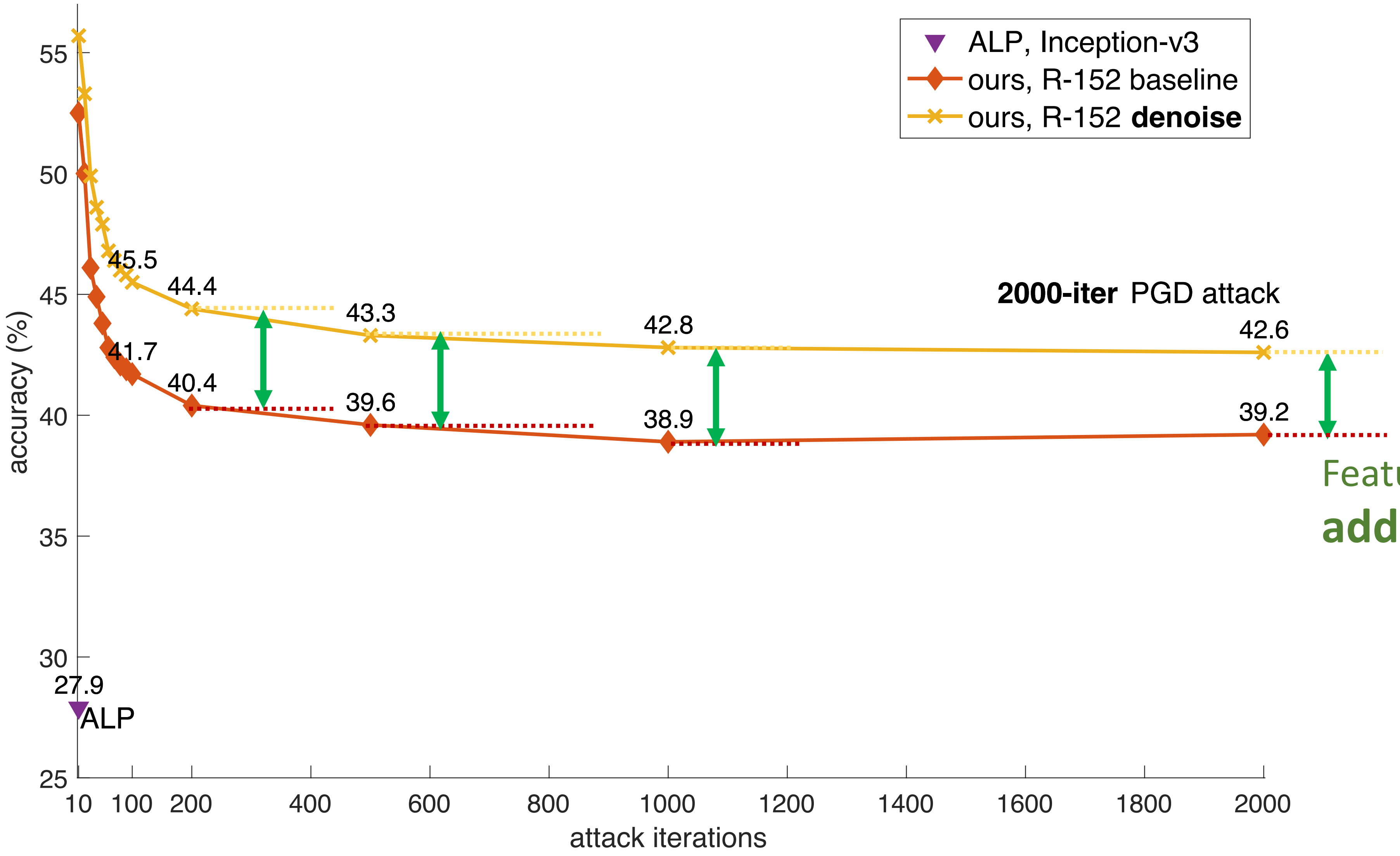
Defending Against White-box Attacks

- Evaluating against adversarial attackers with attack iteration up to 2000
  (more attack iterations indicate stronger attacks)

Defending Against White-box Attacks – Part I

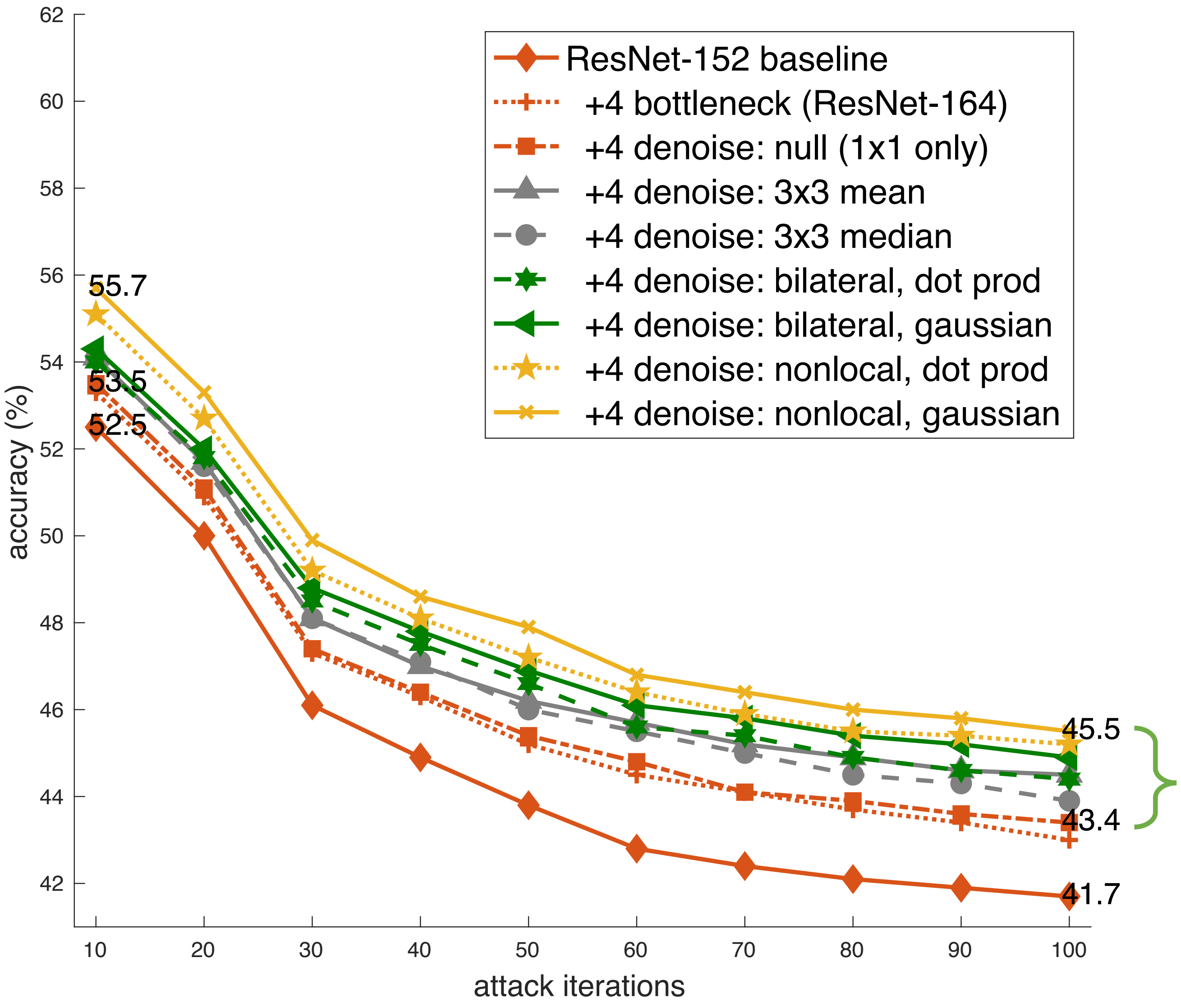


Defending Against White-box Attacks – Part I



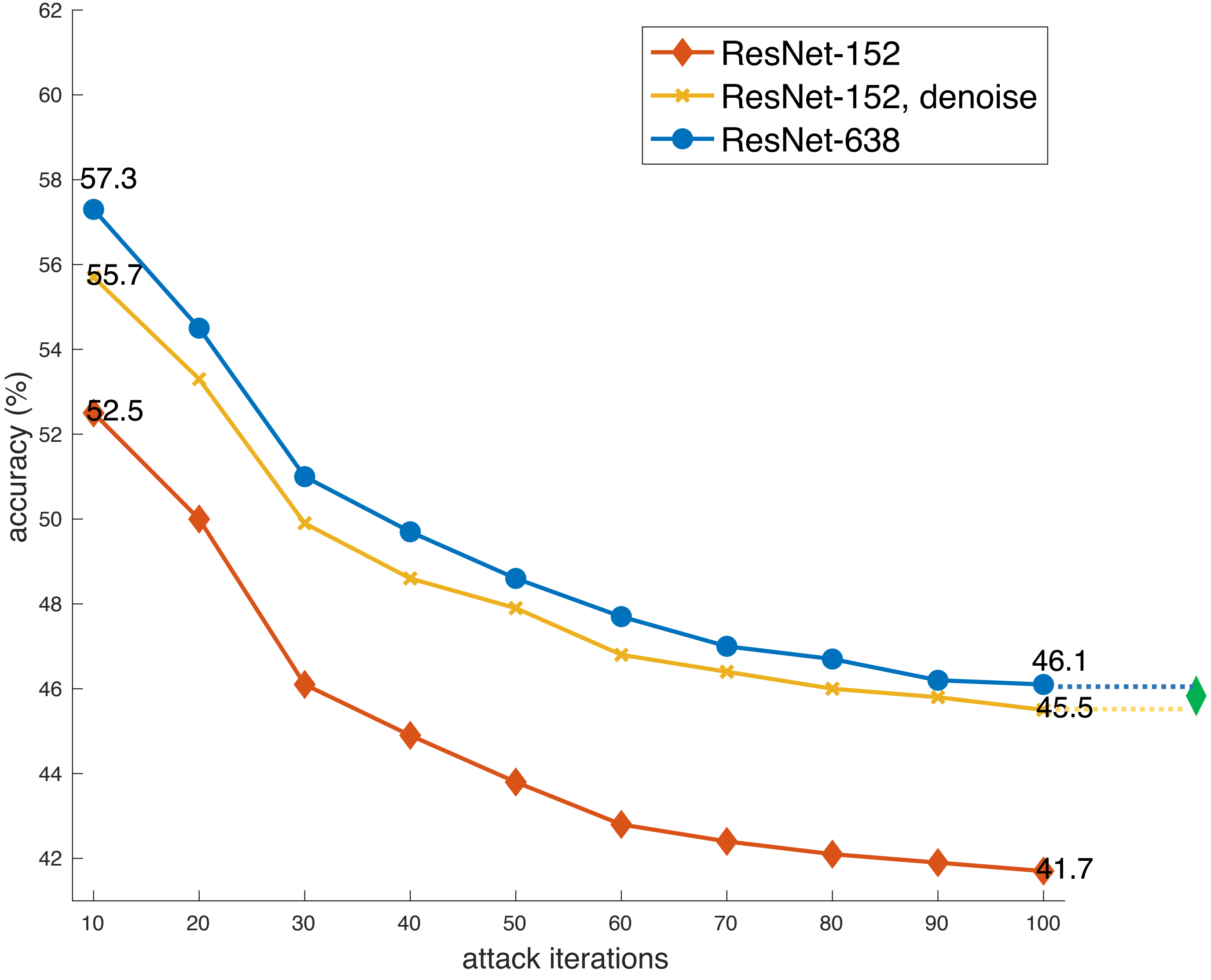
Feature Denoising can give us **additional** benefits

Defending Against White-box Attacks – Part II



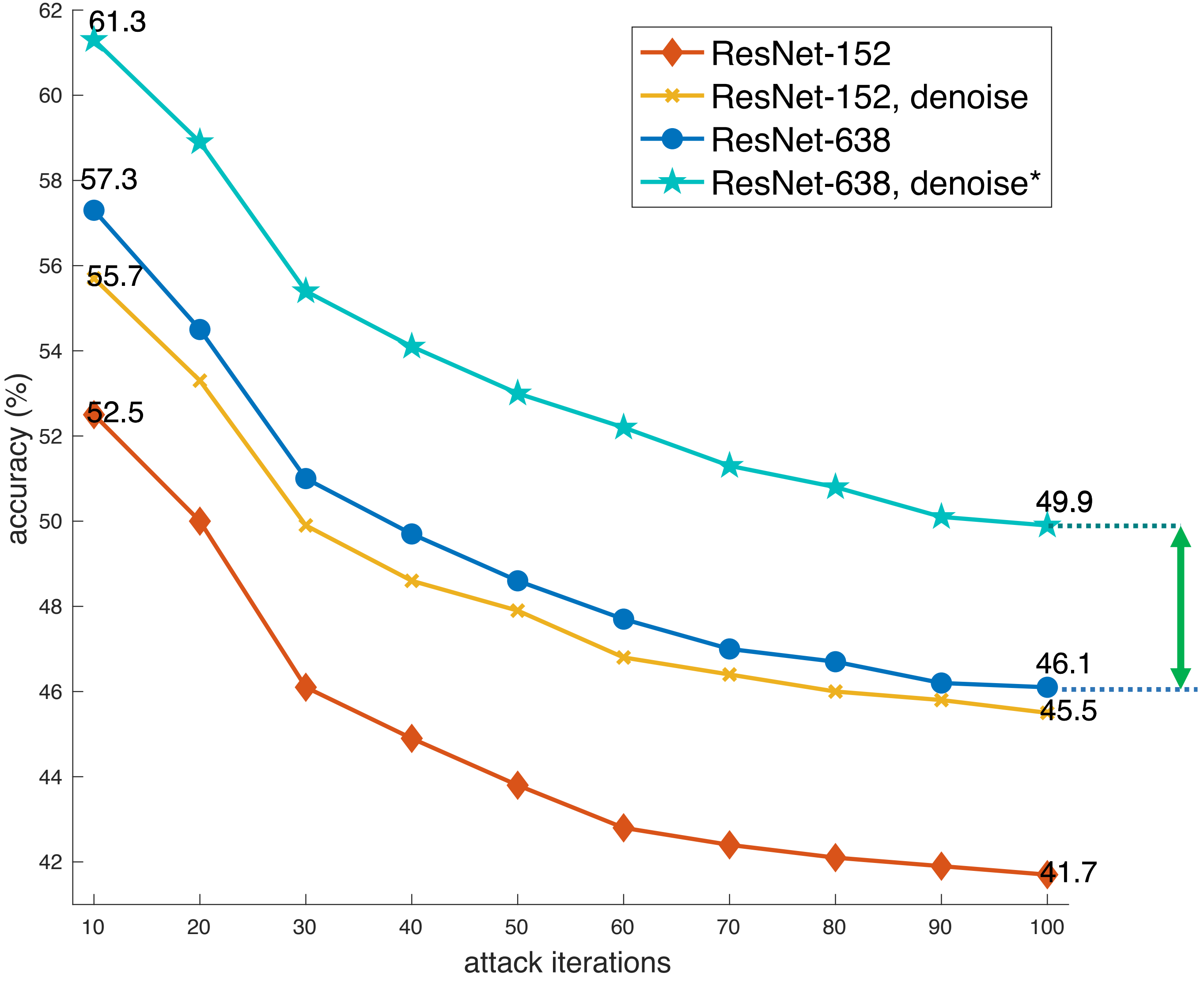
All denoising operations can help

Defending Against White-box Attacks – Part III



Feature Denoising is nearly as powerful as adding ~500 additional layers

Defending Against White-box Attacks – Part III



Feature Denoising can still provide benefits for the VERY deep ResNet-638

Defending Against Blind Attacks

- Offline evaluation against 5 BEST attackers from NeurIPS Adversarial Competition 2017
- Online competition against 48 UNKNOWN attackers in CAAD 2018

Defending Against Blind Attacks

- Offline evaluation against 5 BEST attackers from NeurIPS Adversarial Competition 2017
- Online competition against 48 UNKNOWN attackers in CAAD 2018

CAAD 2018 “all or nothing” criterion: an image is considered correctly classified only if the model correctly classifies all adversarial versions of this image created by all attackers

Defending Against Blind Attacks --- CAAD 2017 Offline Evaluation

model	accuracy (%)
CAAD 2017 winner	0.04
CAAD 2017 winner, under 3 attackers	13.4
ours, R-152 baseline	43.1
+4 denoise: null (1×1 only)	44.1
+4 denoise: non-local, dot product	46.2
+4 denoise: non-local, Gaussian	**46.4**
+all denoise: non-local, Gaussian	**49.5**

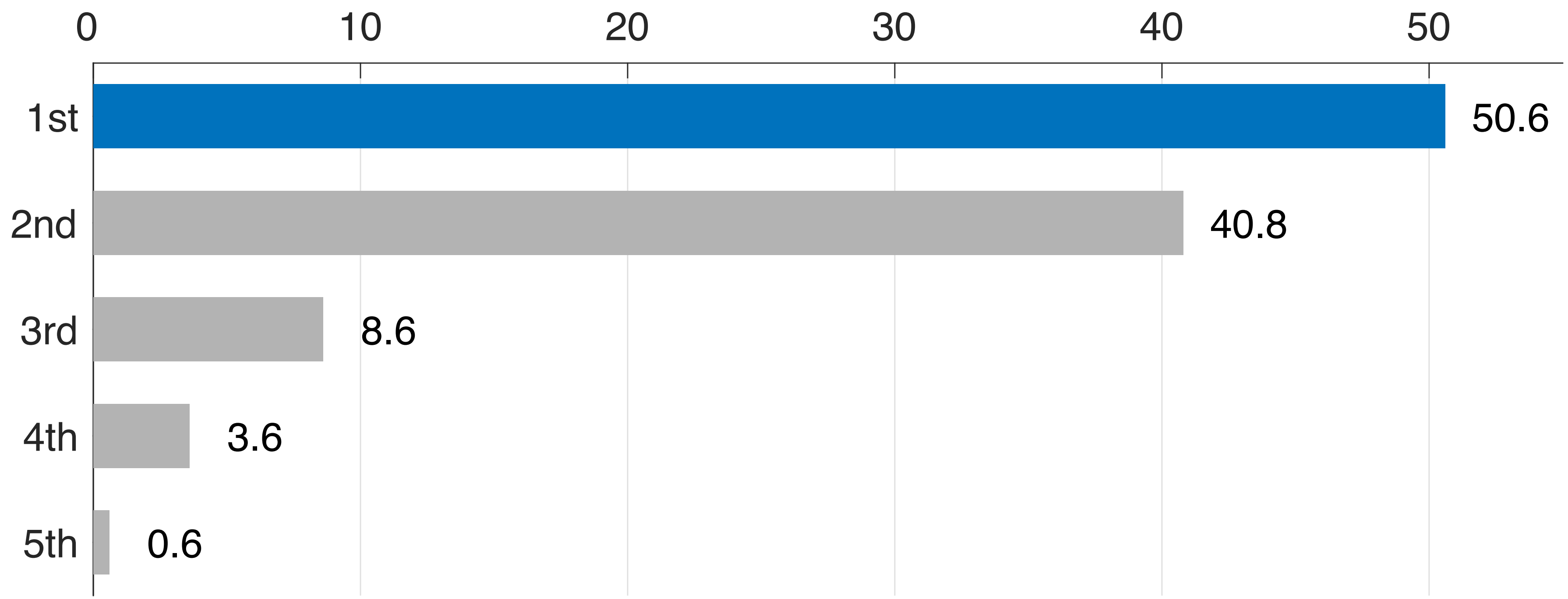
Defending Against Blind Attacks --- CAAD 2017 Offline Evaluation

model	accuracy (%)
CAAD 2017 winner	0.04
CAAD 2017 winner, under 3 attackers	13.4
ours, R-152 baseline	43.1
+4 denoise: null (1×1 only)	44.1
+4 denoise: non-local, dot product	46.2
+4 denoise: non-local, Gaussian	**46.4**
+all denoise: non-local, Gaussian	**49.5**

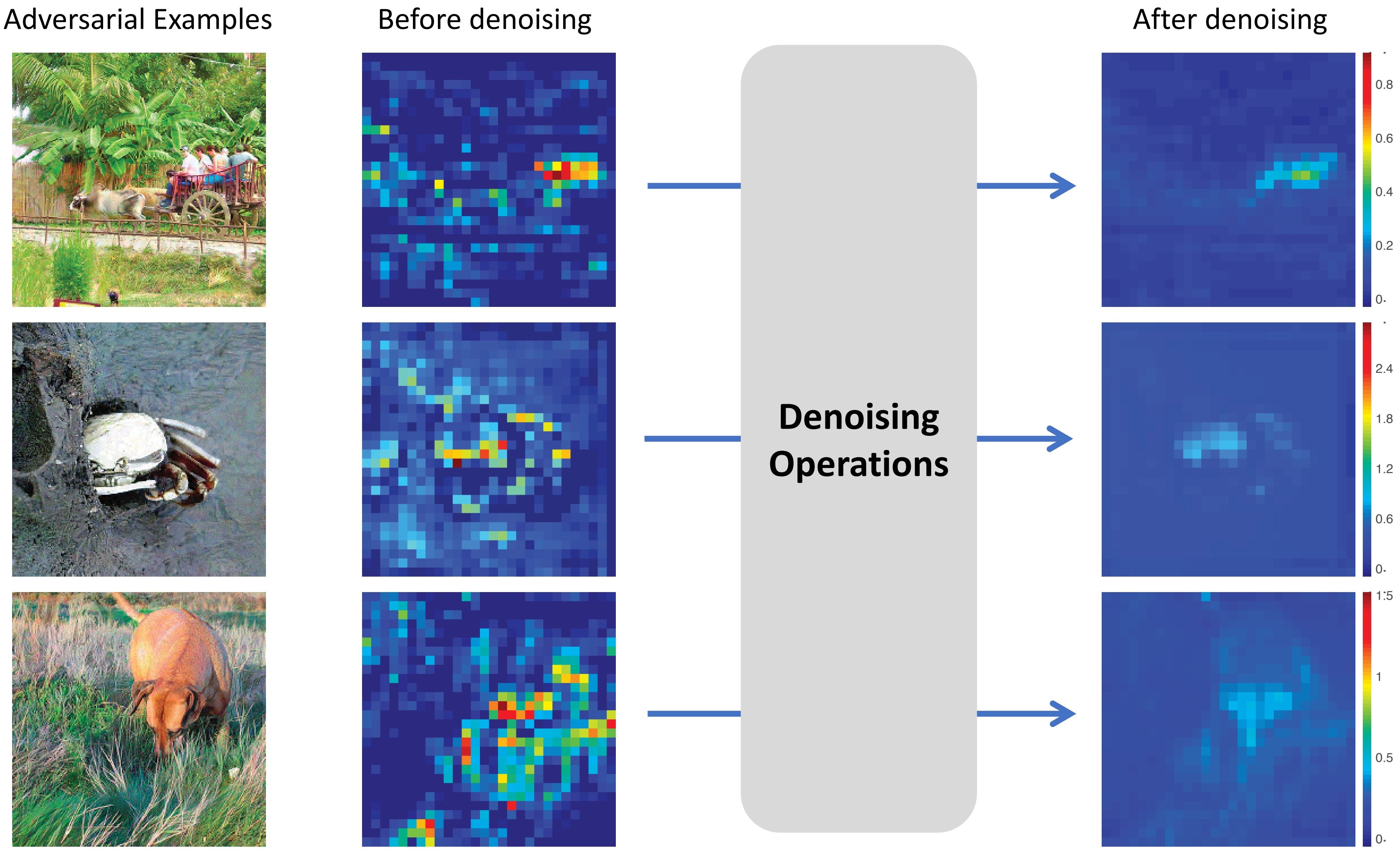
Defending Against Blind Attacks --- CAAD 2017 Offline Evaluation

model	accuracy (%)
CAAD 2017 winner	0.04
CAAD 2017 winner, under 3 attackers	13.4
ours, R-152 baseline	43.1
+4 denoise: null (1×1 only)	44.1
+4 denoise: non-local, dot product	46.2
+4 denoise: non-local, Gaussian	**46.4**
+all denoise: non-local, Gaussian	**49.5**

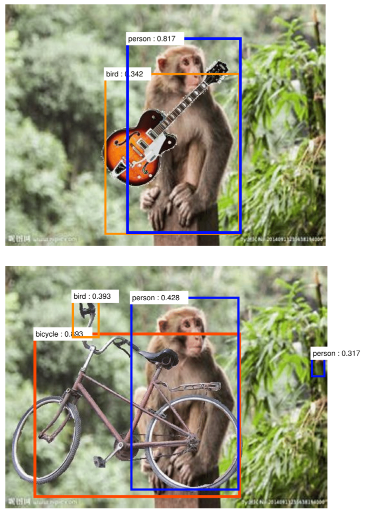
Defending Against Blind Attacks --- CAAD 2018 Online Competition



Visualization



Defending against adversarial attacks is still a long way to go…



Questions?