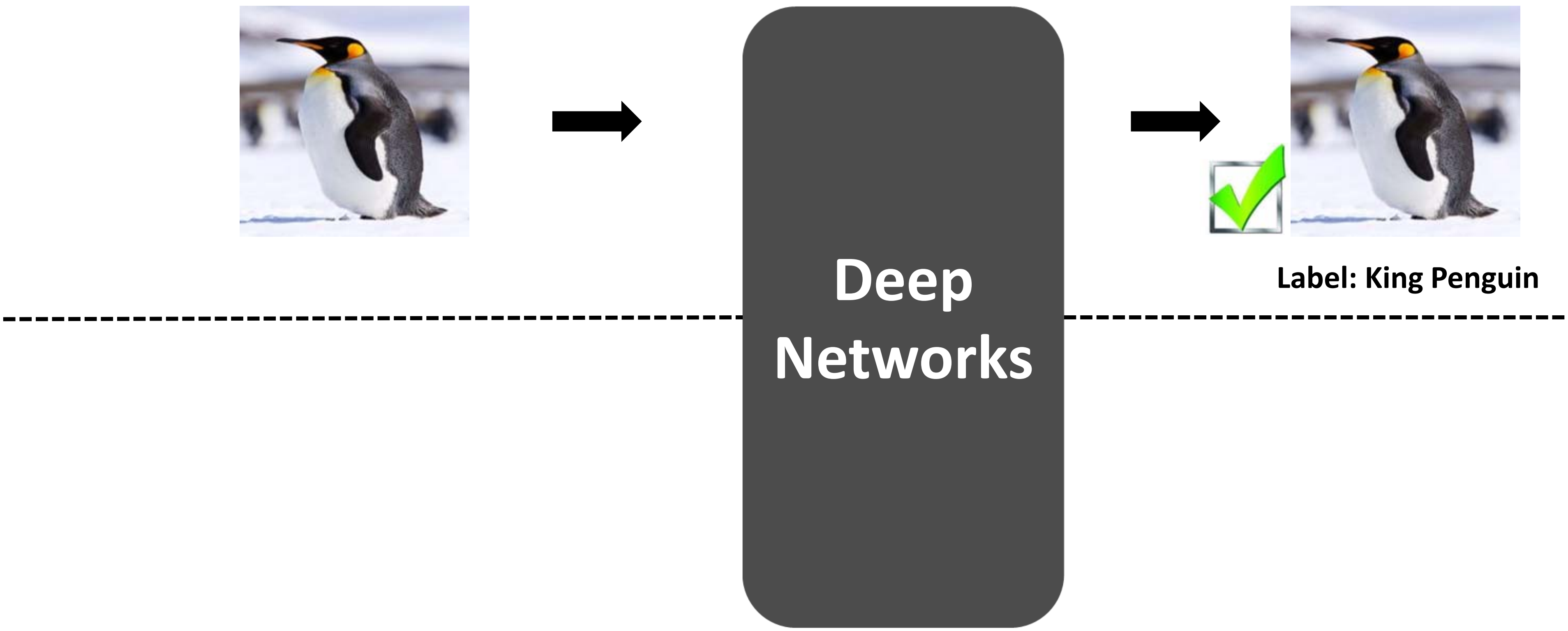
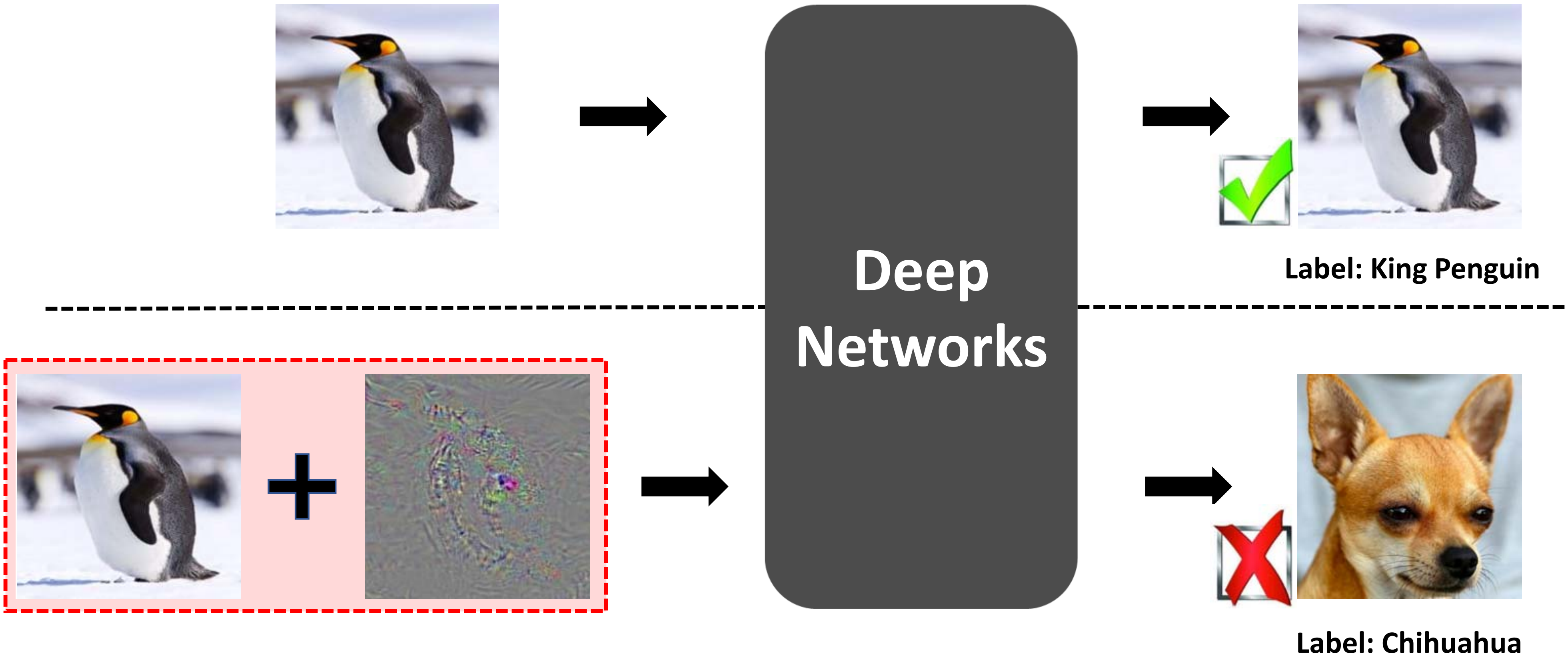
Intriguing Adversarial Examples
&
How To Defend Against Them

Cihang Xie
Johns Hopkins University

Deep networks are **Good**



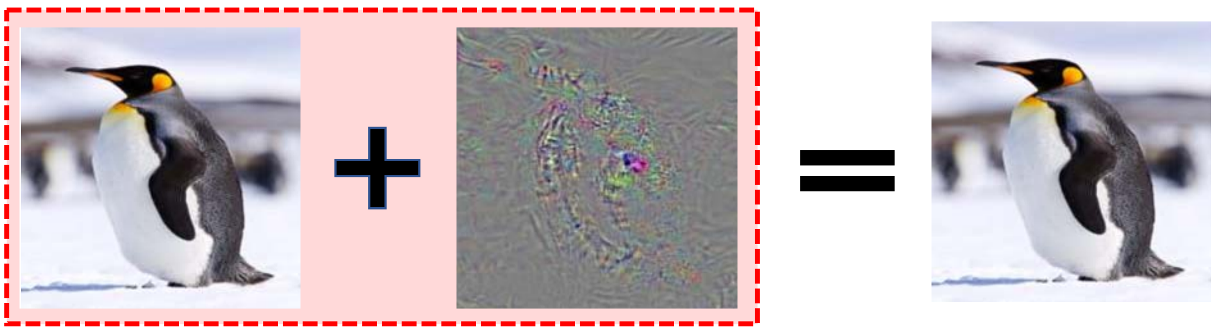
Deep networks are FRAGILE to small & carefully crafted perturbations



Deep networks are FRAGILE to small & carefully crafted perturbations

We call such images as

Adversarial Examples



Generating Adversarial Example is SIMPLE:

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

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

Generating Adversarial Example is SIMPLE:

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

$\Uparrow$

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

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

$\Uparrow$

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

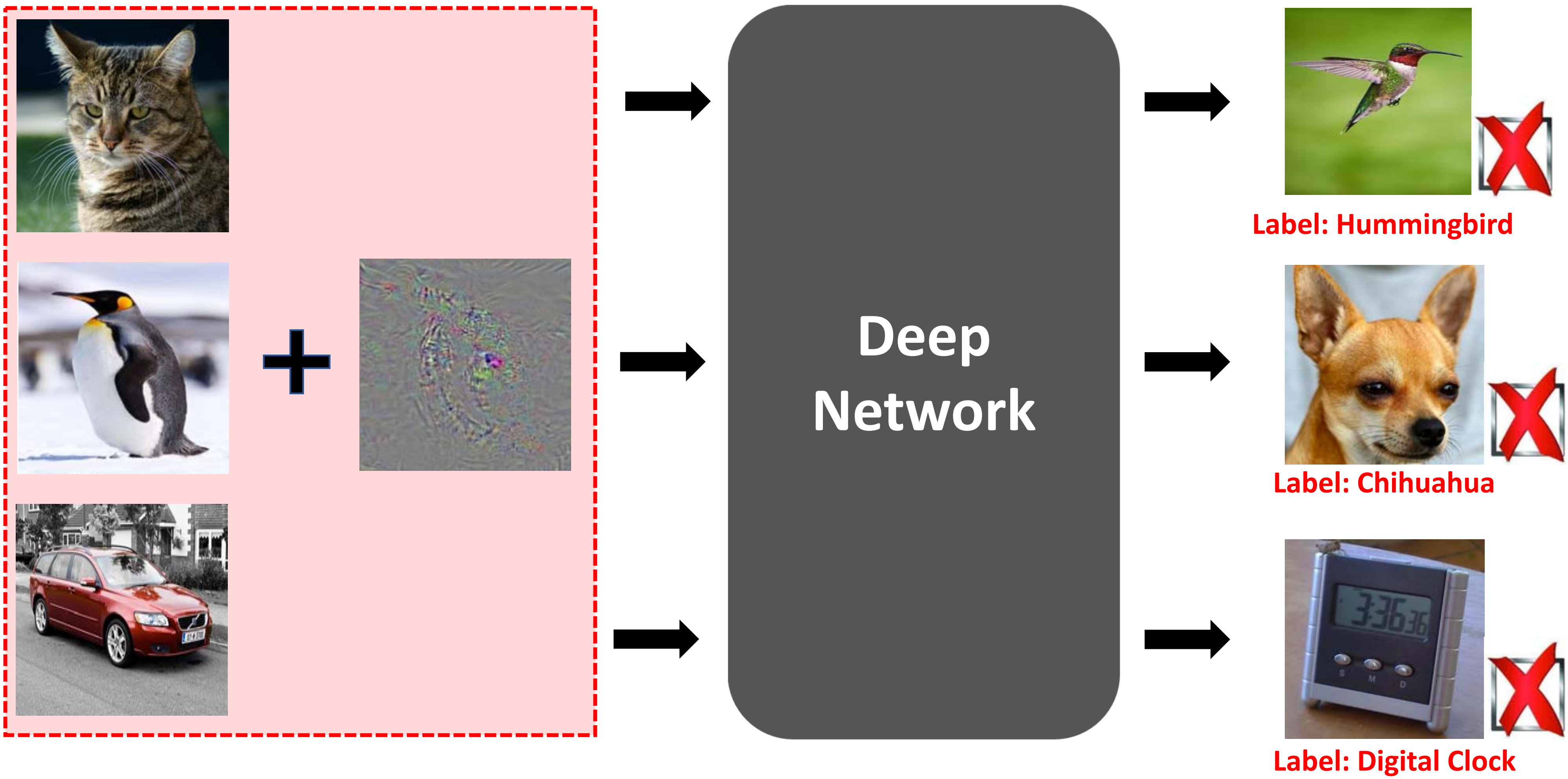
Part I: Intriguing Properties of Adversarial Examples

- {Image, Model, Task}-Agnostic
- Beyond Pixel Perturbation
- Existence in Physical World

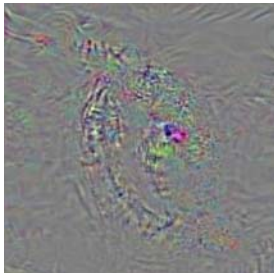
Part I: Intriguing Properties of Adversarial Examples

- **{Image, Model, Task}-Agnostic**
- Beyond Pixel Perturbation
- Existence in Physical World

Adversarial Perturbations can be Image Agnostic



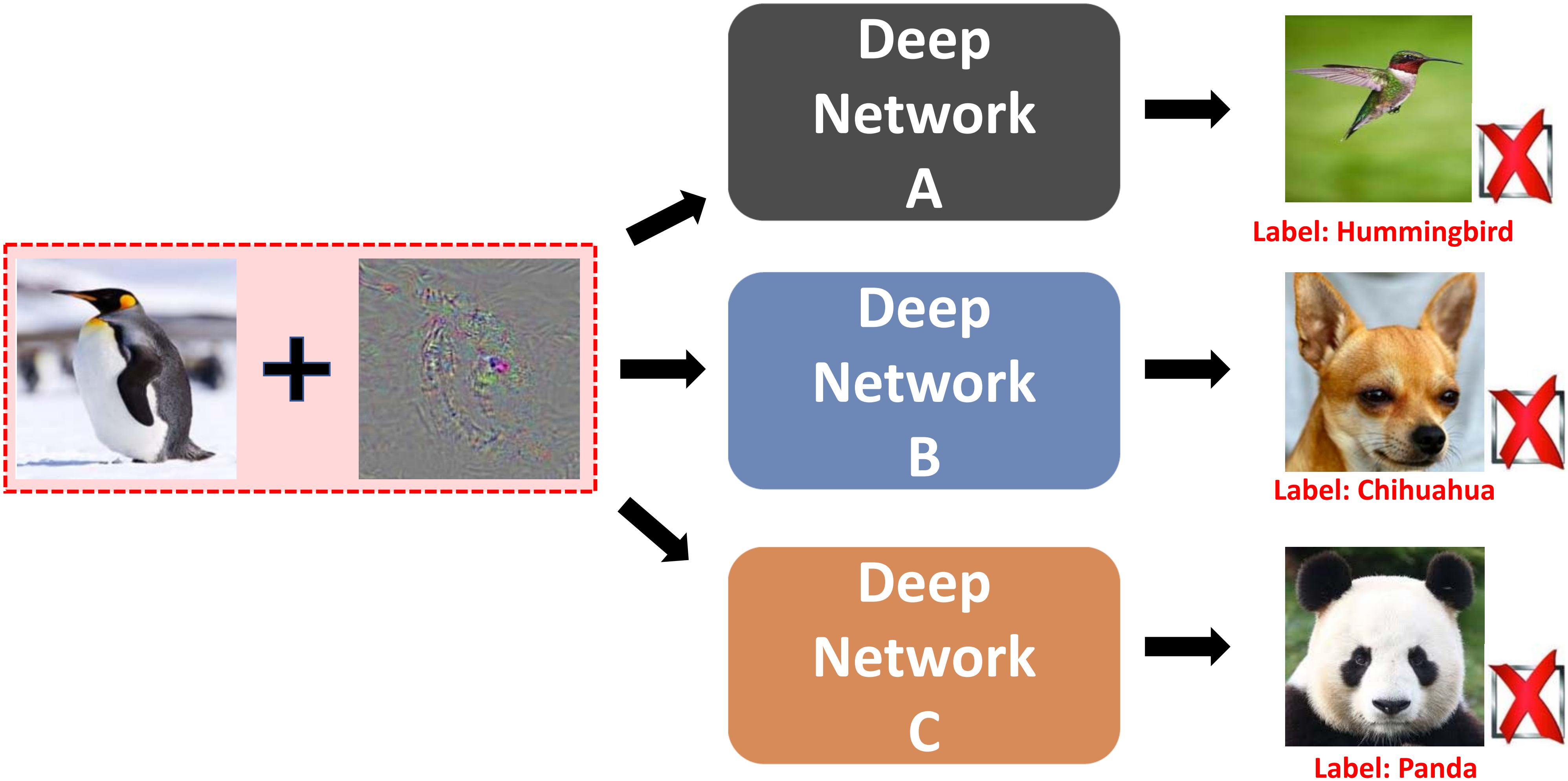
Adversarial Perturbations can be Image Agnostic



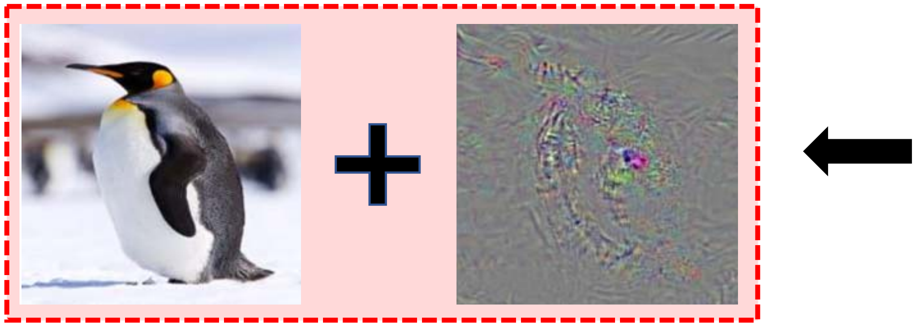
We call such perturbations as

Universal Adversarial Perturbations

Adversarial Examples can be Model Agnostic



Adversarial Examples can be Model Agnostic



We call such images as

Transferable Adversarial Examples

Adversarial Examples can be Task Agnostic

Adversarial examples **EXIST** on different tasks

Adversarial Examples can be Task Agnostic

Adversarial examples **EXIST** on different tasks

semantic segmentation

Adversarial Examples can be Task Agnostic

Adversarial examples **EXIST** on different tasks

semantic segmentation

pose estimation

Adversarial Examples can be Task Agnostic

Adversarial examples **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**

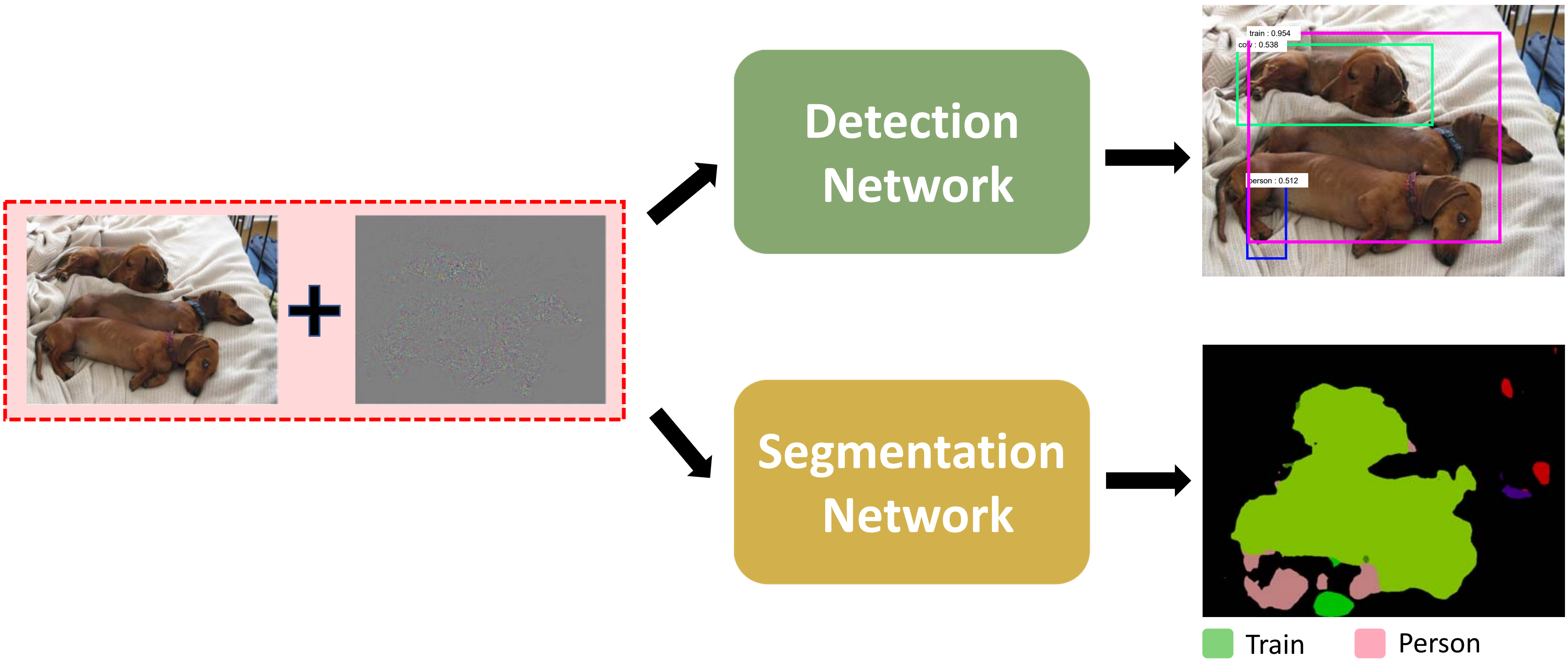
text classification

Adversarial Examples can be Task Agnostic

Adversarial examples **TRANSFER** between different tasks

Adversarial Examples can be Task Agnostic

Adversarial examples **TRANSFER** between different tasks



Quantitative Result of Transferability between Different Models [1]

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152
Inc-v3	FGSM	64.6%	23.5%	21.7%	21.7%
	I-FGSM	**99.9%**	14.8%	11.6%	8.9%
	DI^2-FGSM (**Ours**)	**99.9%**	35.5%	27.8%	21.4%
	MI-FGSM	**99.9%**	36.6%	34.5%	27.5%
	M-DI^2-FGSM (**Ours**)	**99.9%**	**63.9%**	**59.4%**	**47.9%**

Adversarial examples generated on Inc-v3 can attack Inc-v4, IncRes-v2 and Res-152 with high success rate.

[1] Xie, Cihang, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. "Improving transferability of adversarial examples with input diversity." In *CVPR*, 2019

Quantitative Result of Transferability between Different Models [1]

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152
Inc-v3	FGSM	64.6%	23.5%	21.7%	21.7%
	I-FGSM	**99.9%**	14.8%	11.6%	8.9%
	DI^2-FGSM (**Ours**)	**99.9%**	35.5%	27.8%	21.4%
	MI-FGSM	**99.9%**	36.6%	34.5%	27.5%
	M-DI^2-FGSM (**Ours**)	**99.9%**	**63.9%**	**59.4%**	**47.9%**

Adversarial examples generated on Inc-v3 can attack Inc-v4, IncRes-v2 and Res-152 with high success rate.

This transfer phenomenon may indicates

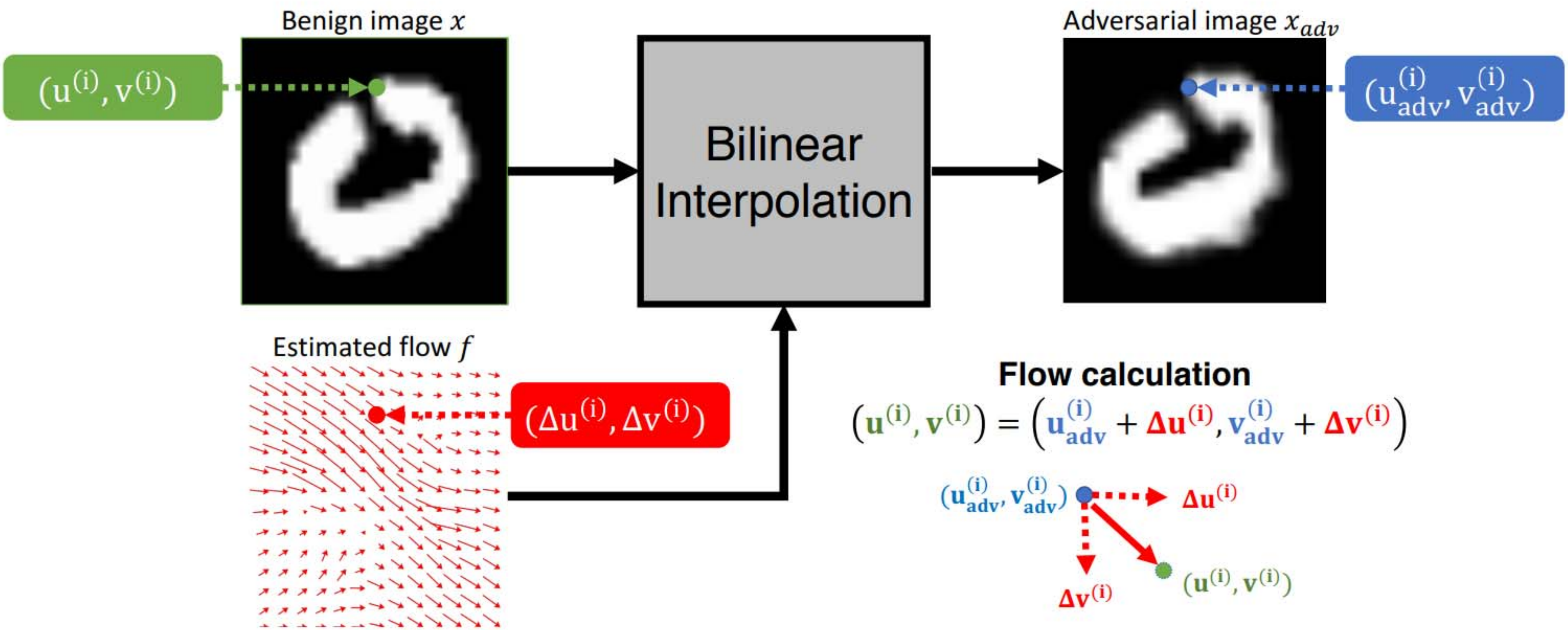
Different Networks Learn Similar Representations

[1] Xie, Cihang, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. "Improving transferability of adversarial examples with input diversity." In *CVPR*, 2019

Part I: Intriguing Properties of Adversarial Examples

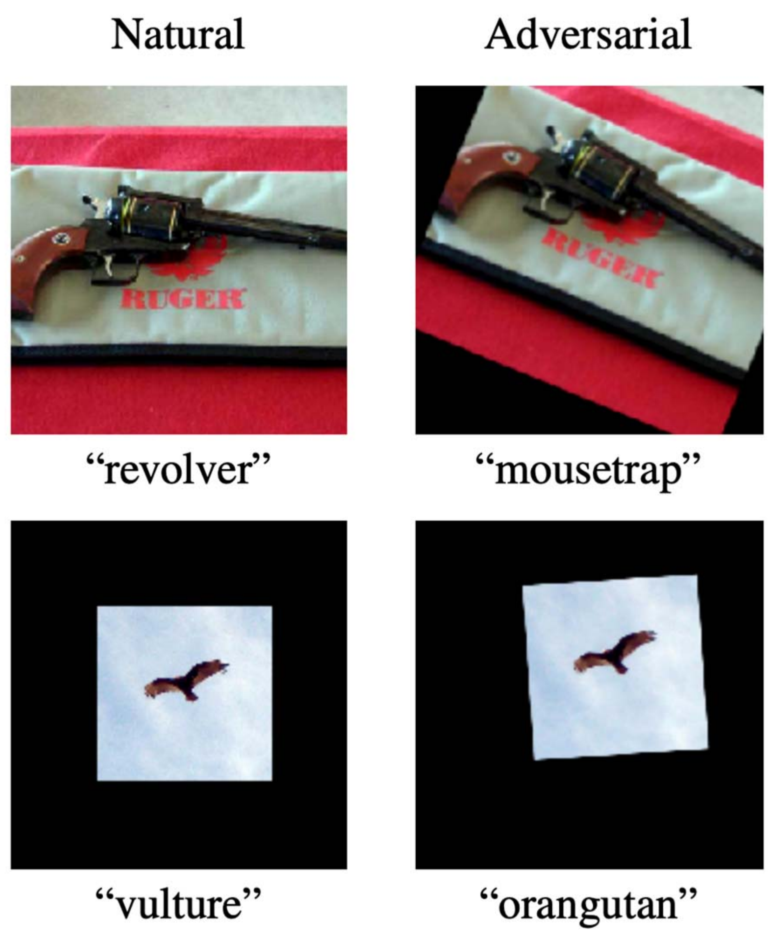
- {Image, Model, Task}-Agnostic
- **Beyond Pixel Perturbation**
- Existence in the Physical World

Beyond Pixel Perturbations --- Spatially Transformed Adversary [2]



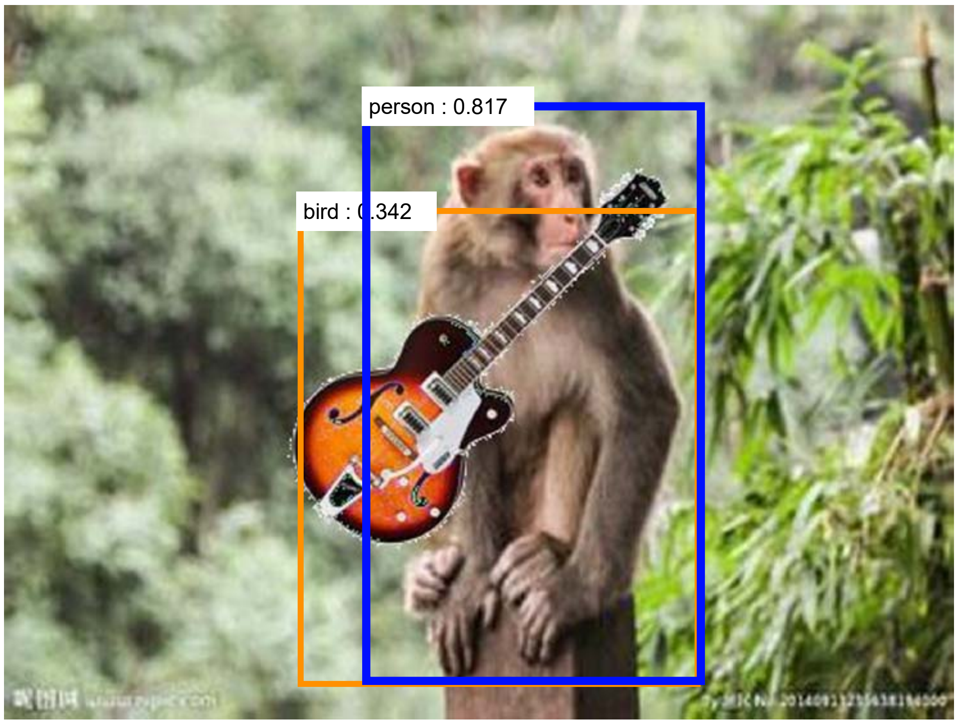
[2] Xiao, Chaowei, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. "Spatially transformed adversarial examples." In *ICLR.* 2018.

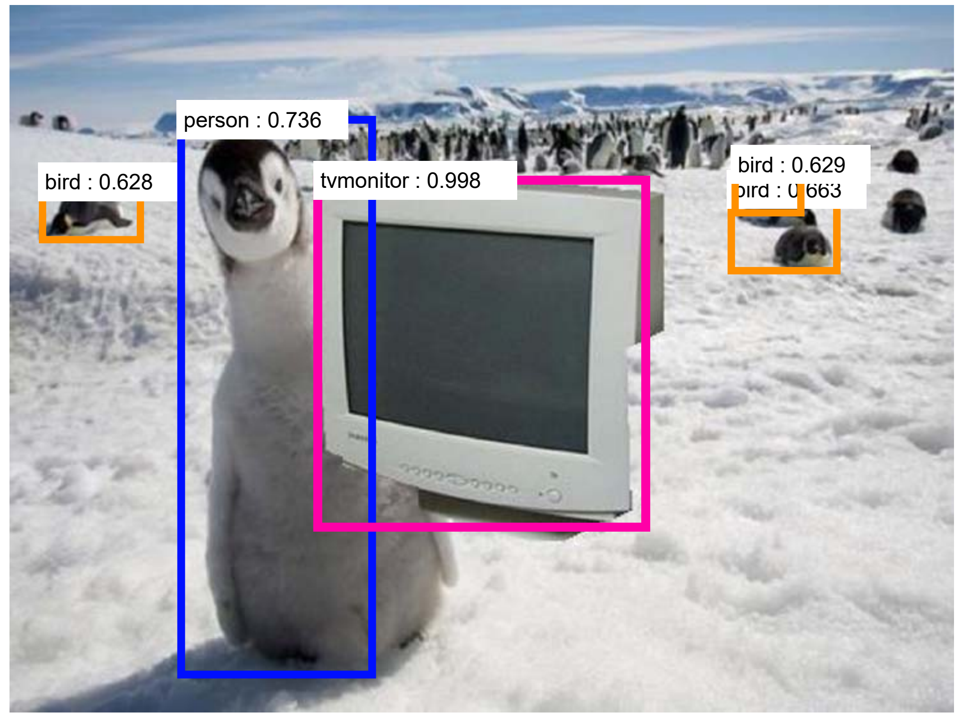
Only Rotation & Translation Are Enough! [3]



[2] Engstrom, Logan, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. "A rotation and a translation suffice: Fooling cnns with simple transformations." In *ICML.* 2019

Beyond Pixel Perturbations --- Adversarial Context Examples [4]



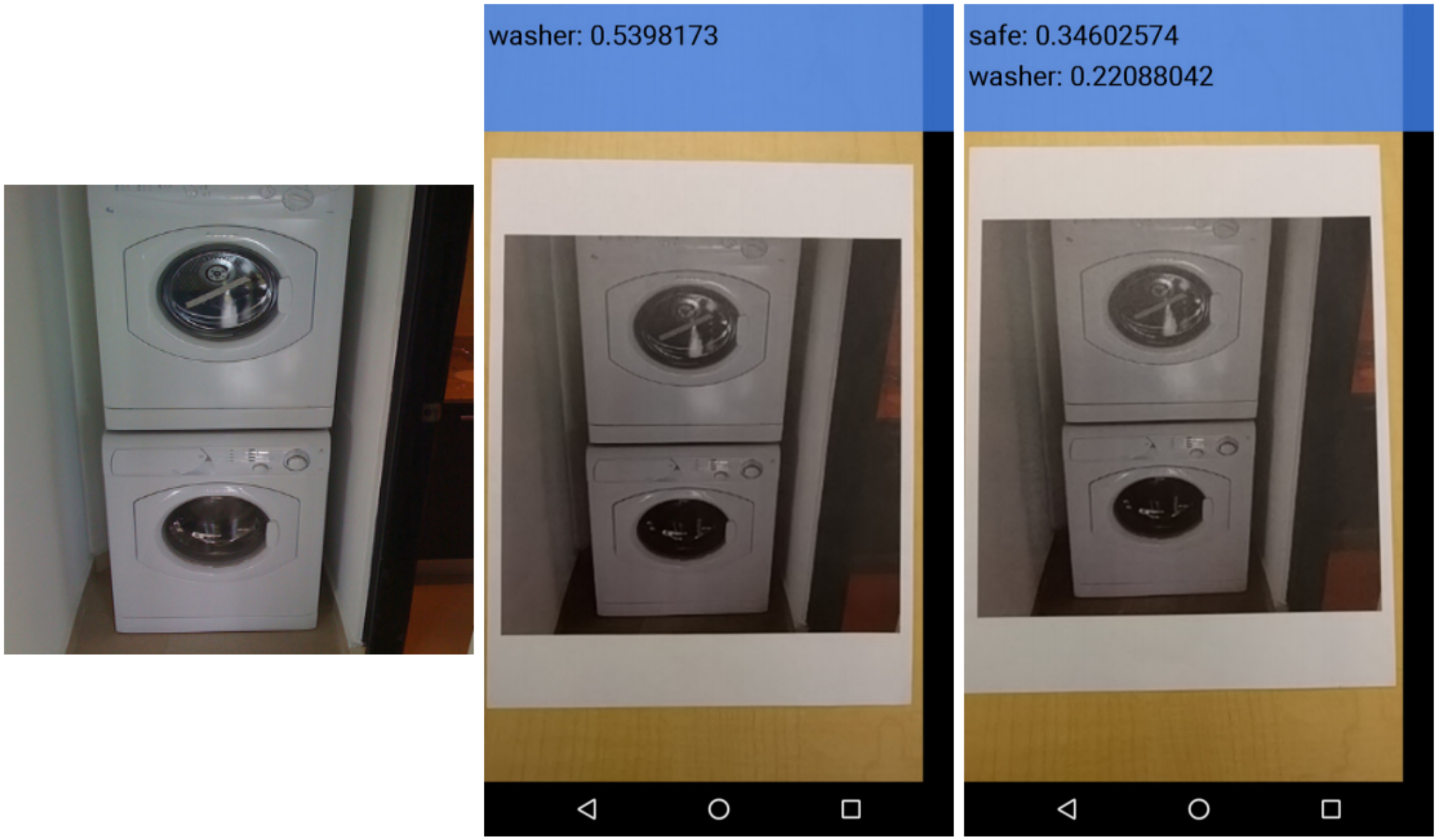


[4] Wang, Jianyu, Zhishuai Zhang, Cihang Xie, et al. "Visual concepts and compositional voting." In *Annals of Mathematical Sciences and Applications.* 2018 .

Part I: Intriguing Properties of Adversarial Examples

- {Image, Model, Task}-Agnostic
- Beyond Pixel Perturbation
- **Existence in the Physical World**

Existence in the Physical World --- Imperceptible Perturbations [5]



(a) Image from dataset (b) Clean image (c) Adv. image

[5] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. "Adversarial examples in the physical world." In *ICLR Workshop.* 2017.

Existence in the Physical World --- Perceptible Perturbations [6]

With these adversarial stickers, networks cannot recognize stop signs.

[6] Eykholt, Kevin, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, et al. "Robust physical-world attacks on deep learning models." In *CVPR.* 2018.

Extension --- Attacking Object Detectors in the Physical World [7]

[7] Lifeng Huang, et al. "UPA^2: Learning Universal Physical Adversarial Attack on Object Detectors." In *submission*.

Generating Adversarial Example is SIMPLE:

non-targeted attacks: **maximize** loss(f(x+r), y^{true})

targeted attacks: **minimize** loss(f(x+r), y^{target})

Generating Adversarial Examples is SIMILAR TO NETWORK TRAINING

- Objective functions are SIMILIAR:

For network training, want to **minimize** loss(f(x), y^{true}) ;

For adversarial generation, want to **maximize** loss(f(x+r), y^{true});

Generating Adversarial Examples is similar to Training Neural Networks

- Objective functions are SIMILIAR:

For network training, want to **minimize** loss(f(x), y^{true}; θ);

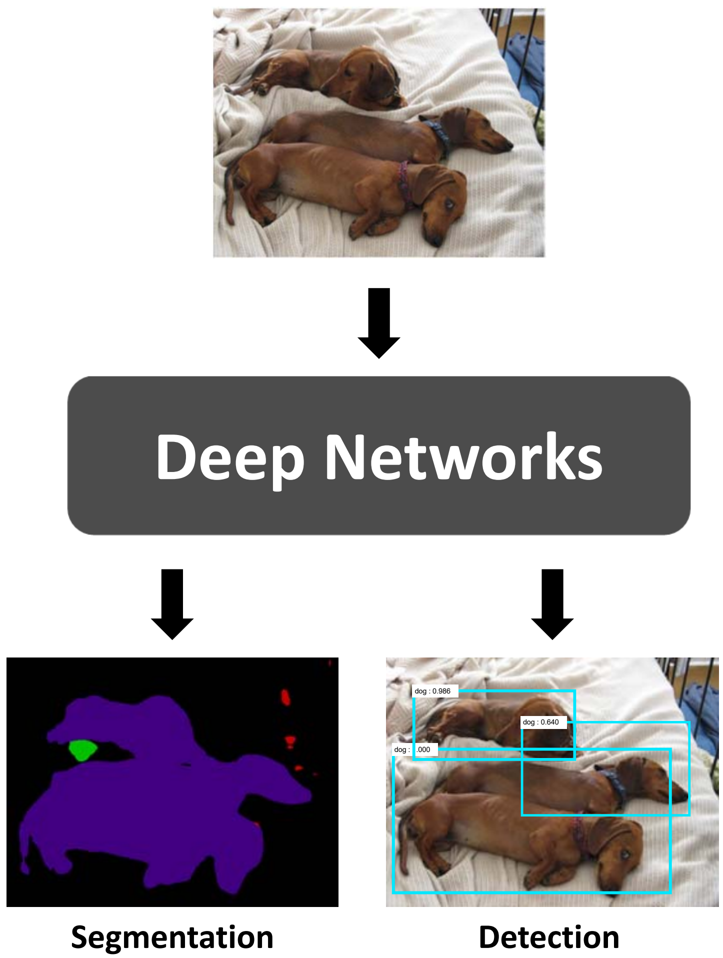
For generating adversary, want to **maximize** loss(f(x+r), y^{true}; θ);

- Optimized variables are DIFFERENT:

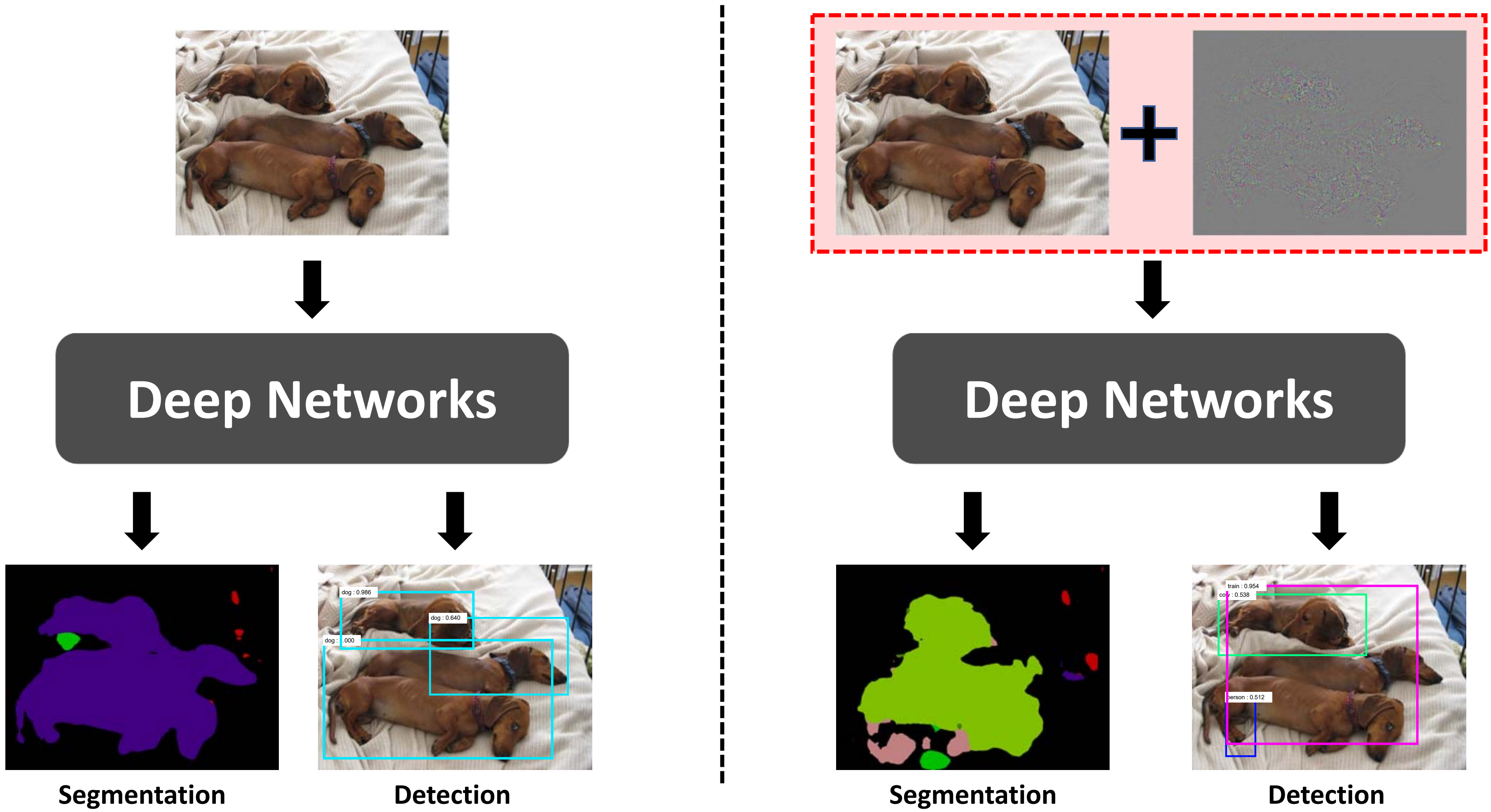
For network training, want to optimize over network parameter θ;

For adversarial generation, want to optimize over perturbation r

Not just for image classification



Not just for image classification, but also for detection and segmentation

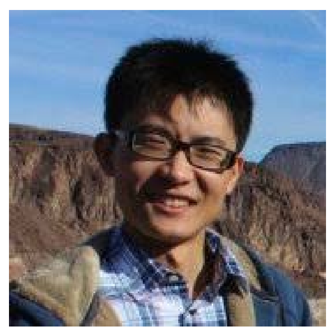
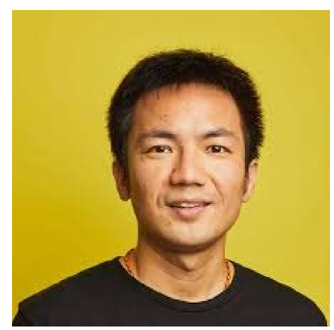


Part I: Towards Transferable Adversarial Attacks

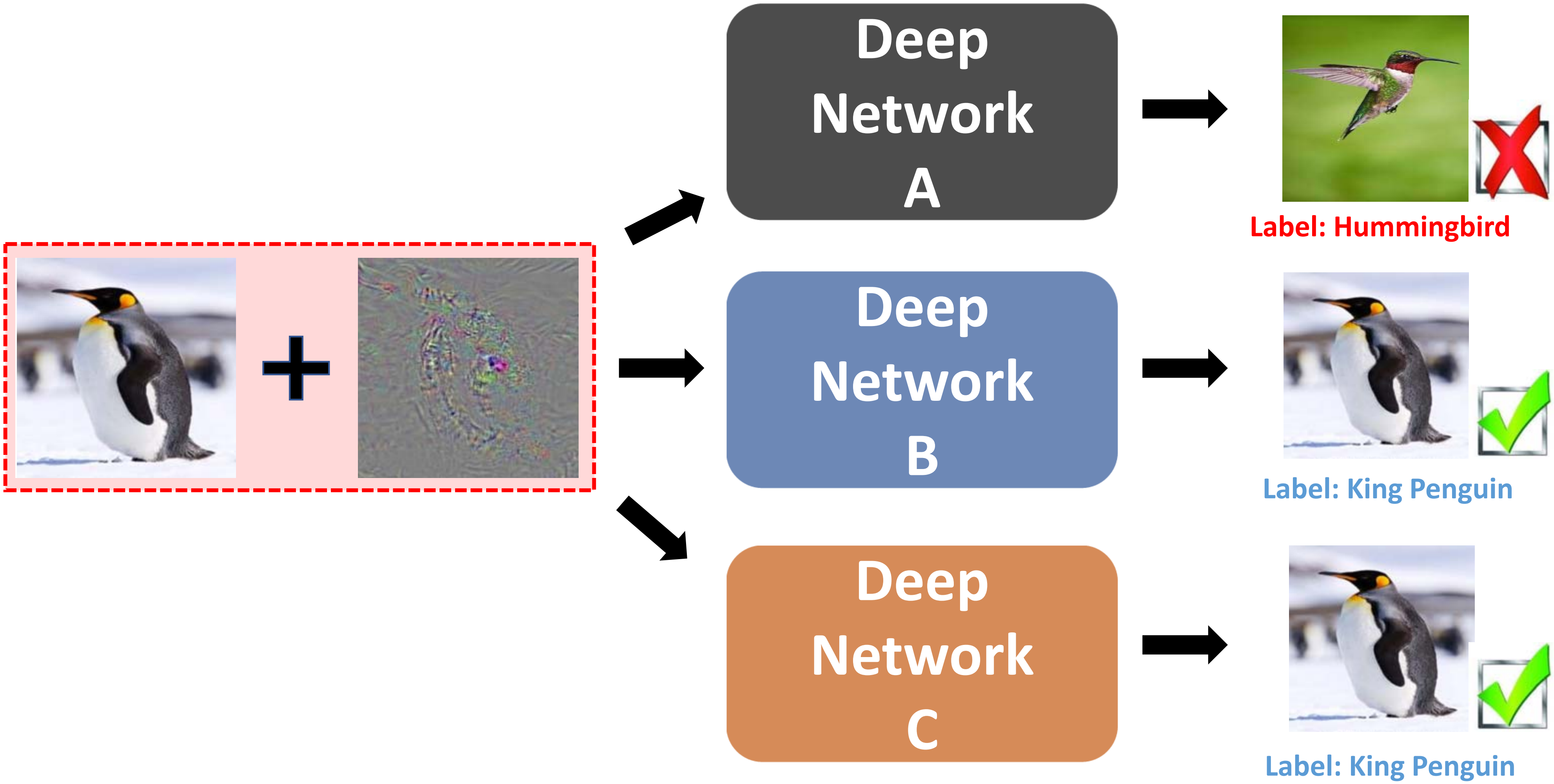
- Diverse Input Patterns

Improving Transferability of Adversarial Examples with Input Diversity (CVPR’19)



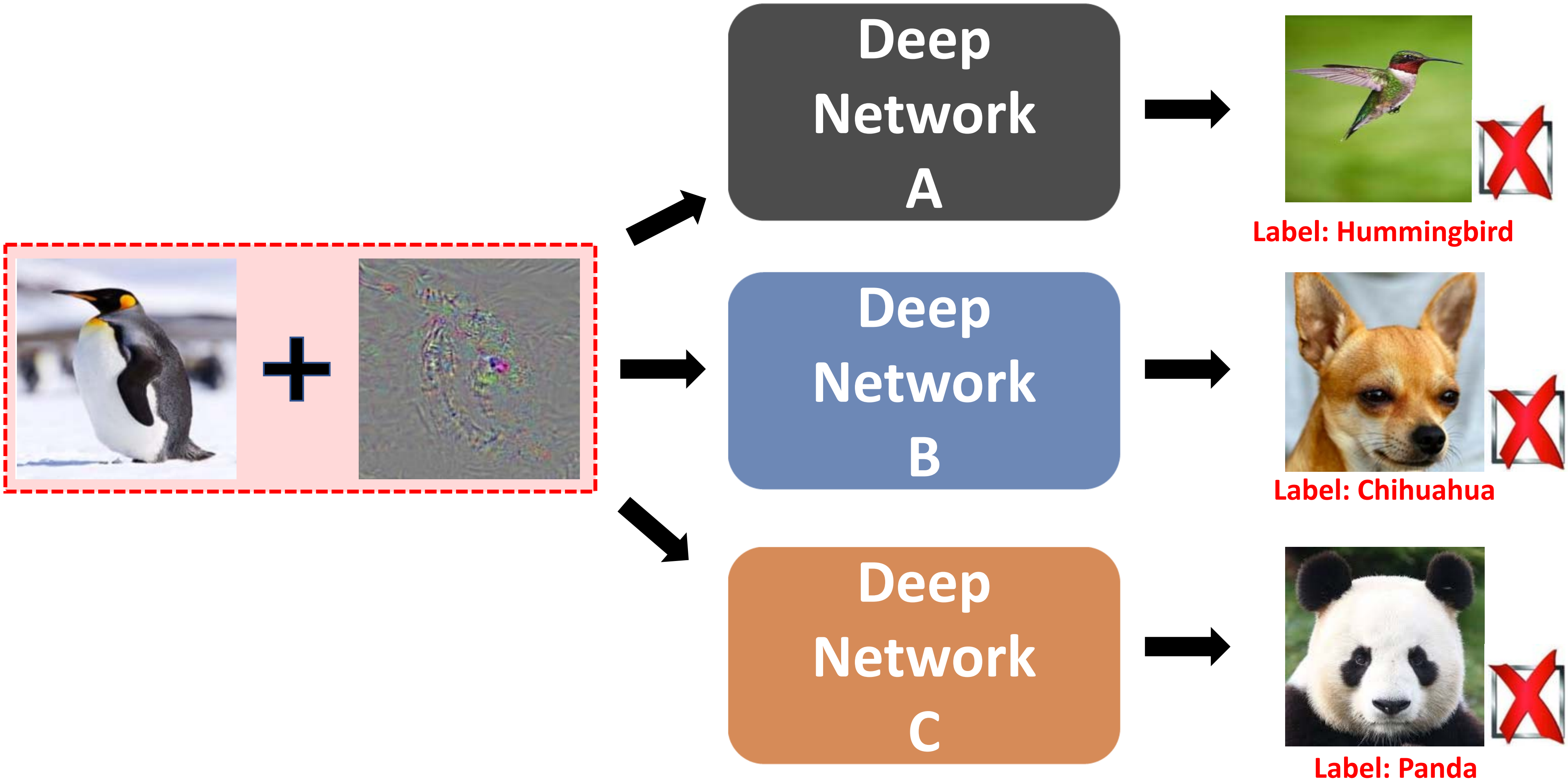

Observation: Traditional Attacks have POOR transferability



Diverse Input Patterns --- observation

Observation: If keep maximizing loss(f(x+r), y^{true}; θ) for multiple steps, the adversarial perturbation r will be overfitted to the network parameter θ --- therefore bad generalization ability

Can we generate STRONGER TRANSFERABLE adversarial examples?



Diverse Input Patterns --- solution

Solution: data augmentation is good at alleviating overfitting

$$\text{maximize loss}(f\ (\underline{\mathbf{T(}}x+r\underline{\mathbf{)}}),\ y^{true};\ \theta)$$

Diverse Input Patterns --- Results

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	FGSM	64.6%	23.5%	21.7%	21.7%	8.0%	7.5%	3.6%
	I-FGSM	**99.9%**	14.8%	11.6%	8.9%	3.3%	2.9%	1.5%
	DI^2-FGSM (**Ours**)	**99.9%**	35.5%	27.8%	21.4%	5.5%	5.2%	2.8%
	MI-FGSM	**99.9%**	36.6%					
	M-DI^2-FGSM (**Ours**)	**99.9%**	**63.9%**					

Our method can generate more transferable adversarial examples on unknown models

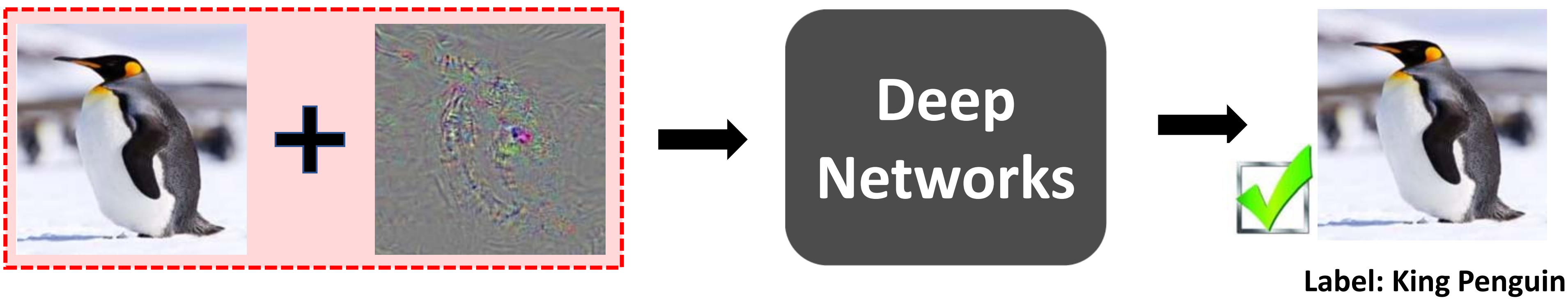
Diverse Input Patterns --- Results

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	FGSM	64.6%	23.5%	21.7%	21.7%	8.0%	7.5%	3.6%
	I-FGSM	**99.9%**	14.8%	11.6%	8.9%	3.3%	2.9%	1.5%
	DI^2-FGSM (**Ours**)	**99.9%**	35.5%	27.8%	21.4%	5.5%	5.2%	2.8%
	MI-FGSM	**99.9%**	36.6%	34.5%	27.5%	8.9%	8.4%	4.7%
	M-DI^2-FGSM (**Ours**)	**99.9%**	**63.9%**	**59.4%**	**47.9%**	**14.3%**	**14.0%**	**7.0%**

Our method can boost the transferability further on recently proposed MI-FGSM

Part II: Towards Robust Adversarial Defense

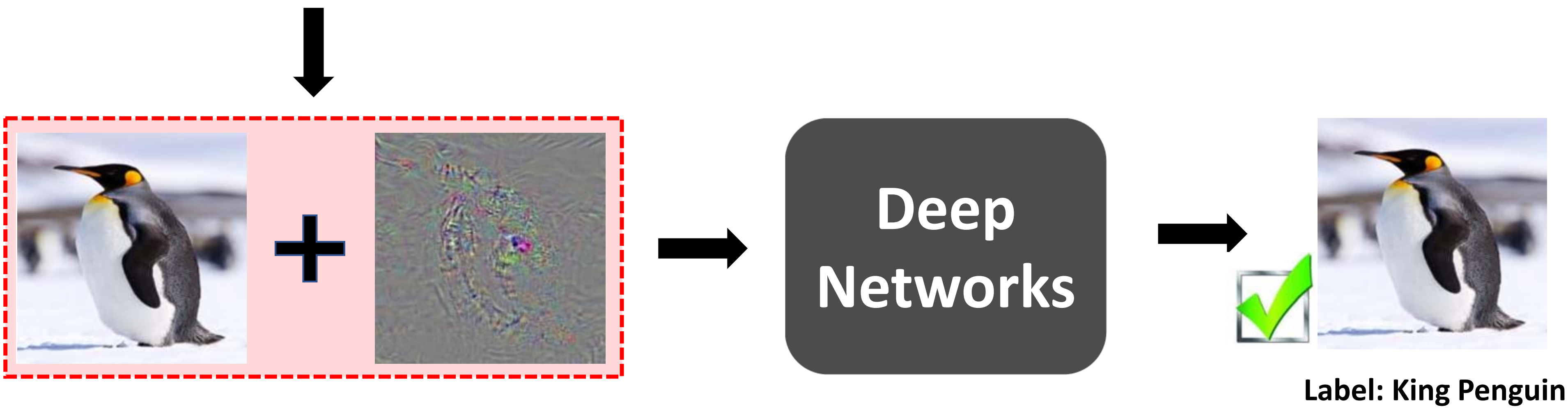
- Robust Input Images
- Robust Network Representations



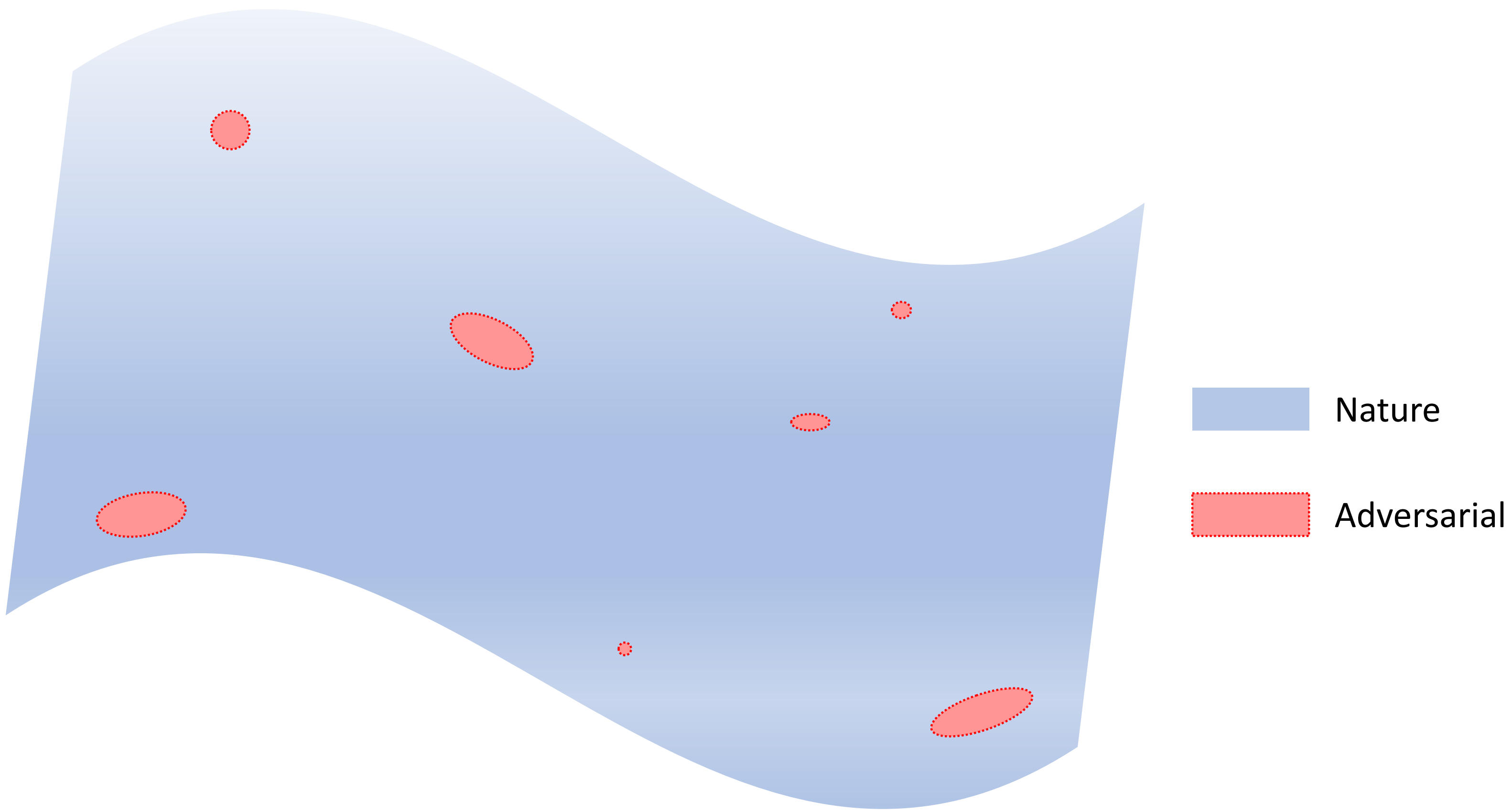
Part II: Towards Robust Adversarial Defense

- **Robust Input Images**
- Robust Network Representations

want to **remove** malicious manipulations from input images



Adversarial examples are SPARSE and ISOLATED on the pixel space

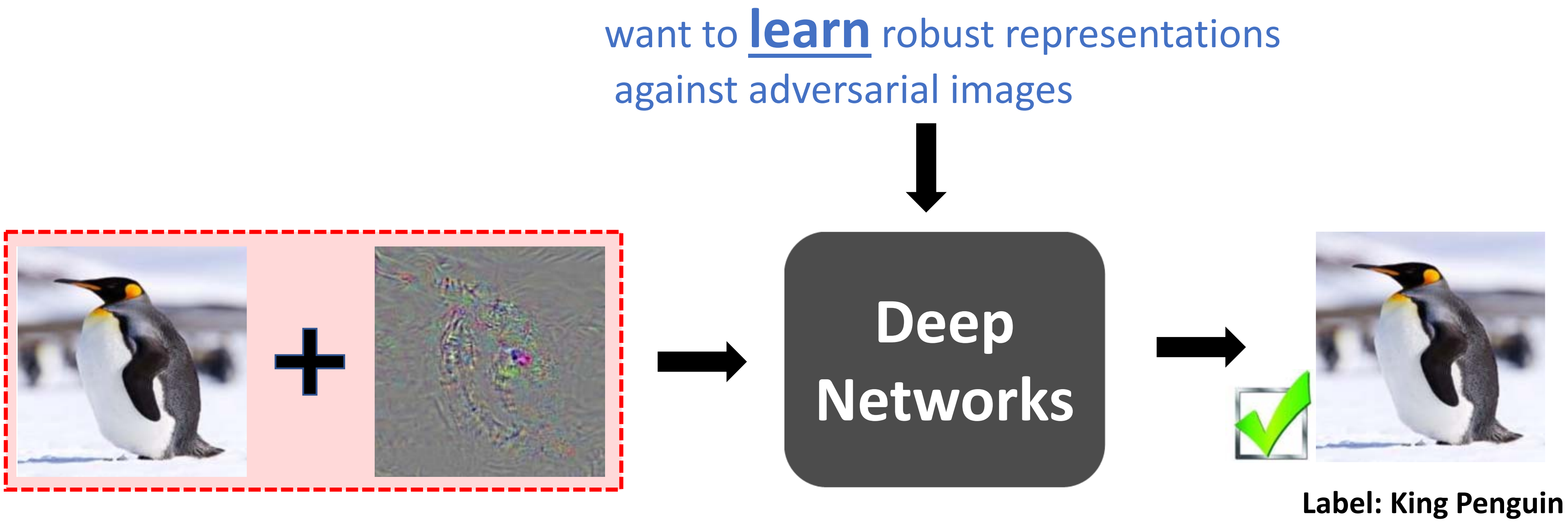


Robust Input Images

- Simple Image Denoiser --- e.g., median filter
- Train a Network for Removing Malicious Perturbations
- Generative Models for Removing Malicious Perturbations

Part II: Towards Robust Adversarial Defense

- Robust Input Images
- **Robust Network Representations**

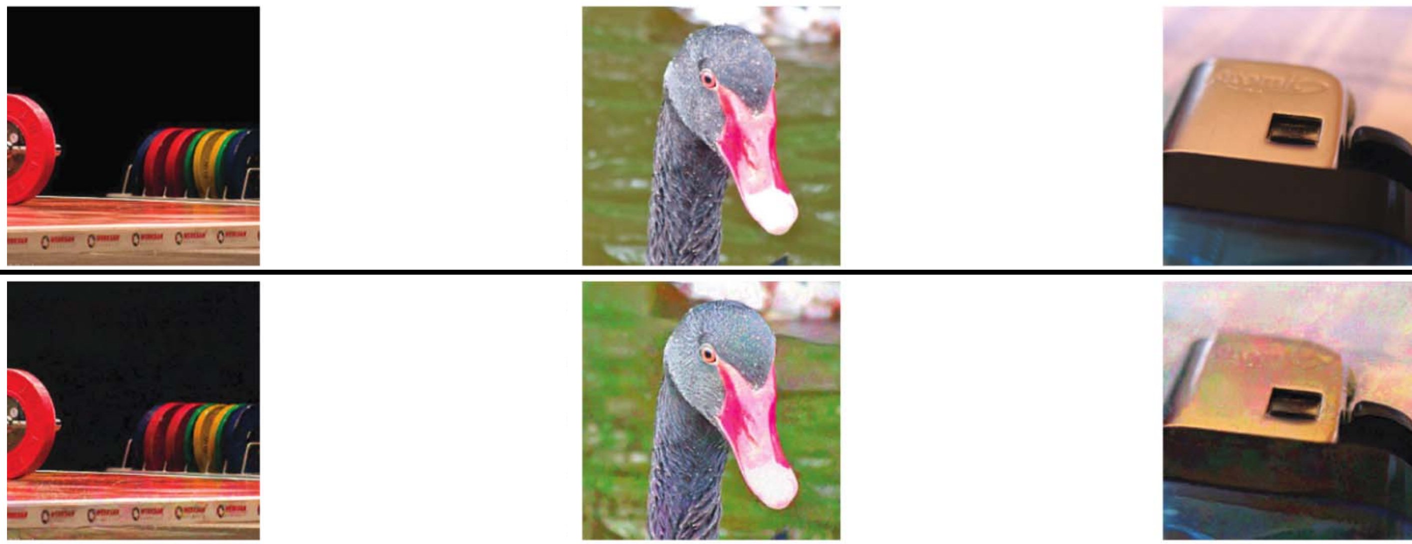


Feature Denoising for Improving Adversarial Robustness (CVPR'19)

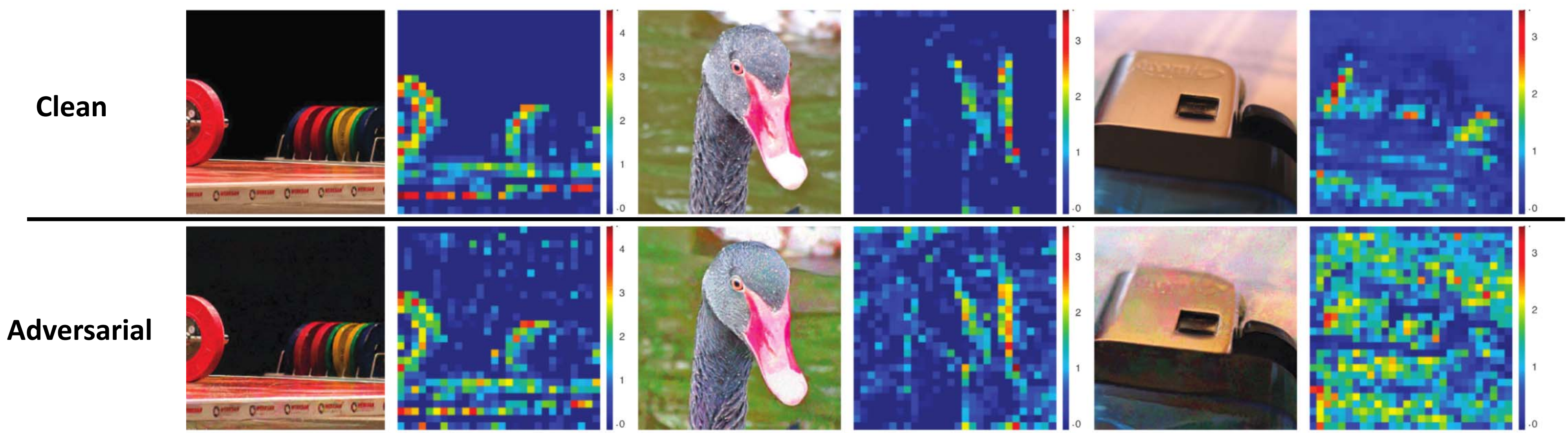
Observation: Adversarial perturbations are SMALL on the pixel space

Clean

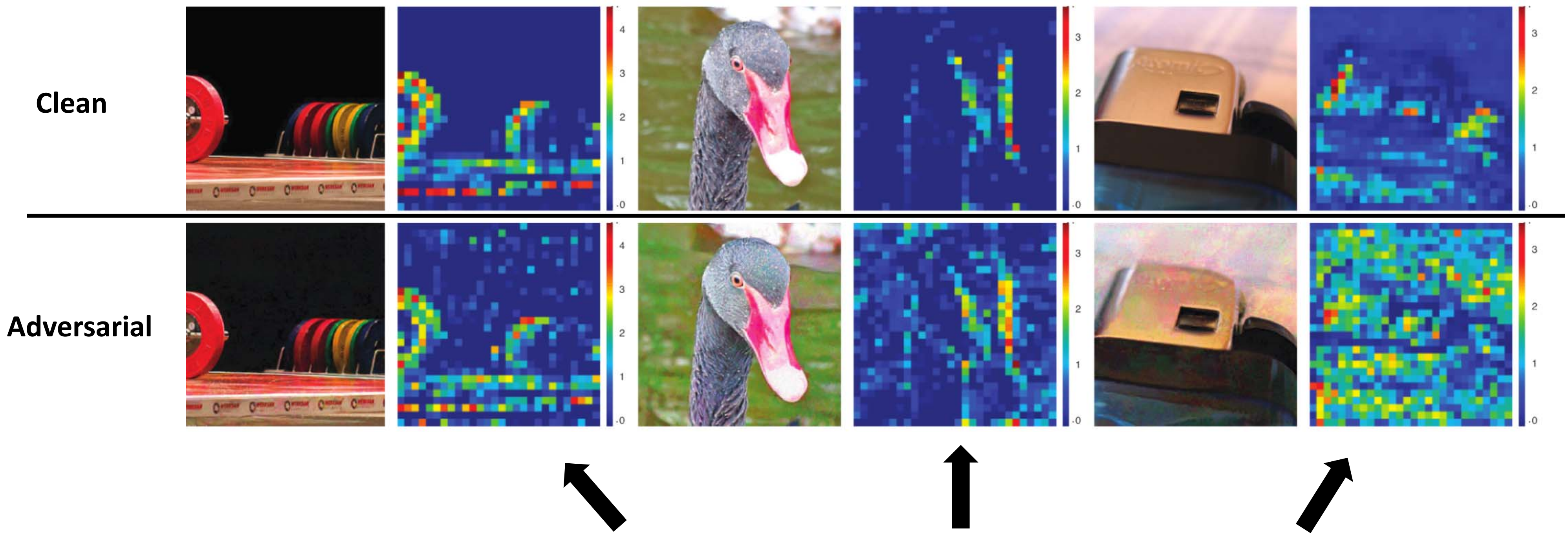
Adversarial



Observation: Adversarial perturbations are BIG on the feature space



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:

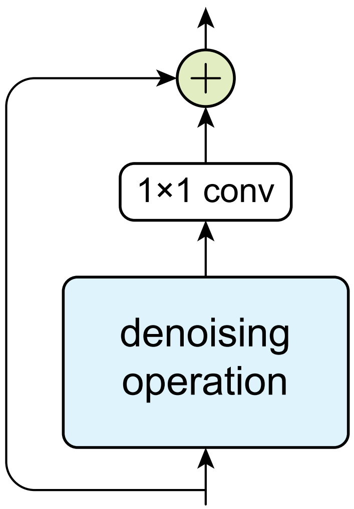
Local filters (predefine a local region $\Omega(i)$ for each pixel i):

- 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

- Implementation: distributed on 128 GPUs, 32 images per GPU
  (since finding adversarial examples is computationally expensive)

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)

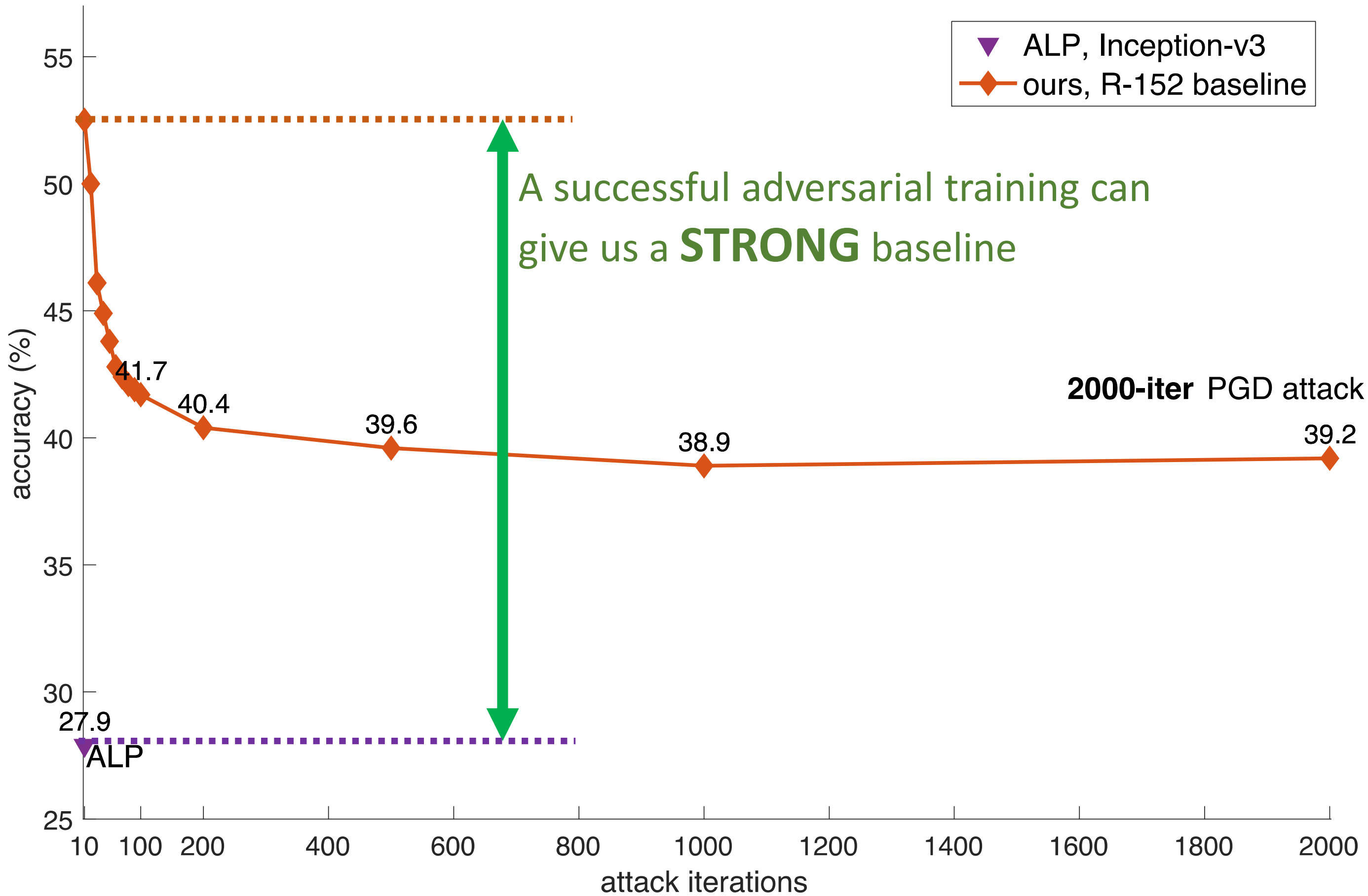
Defending Against Blind Attacks

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

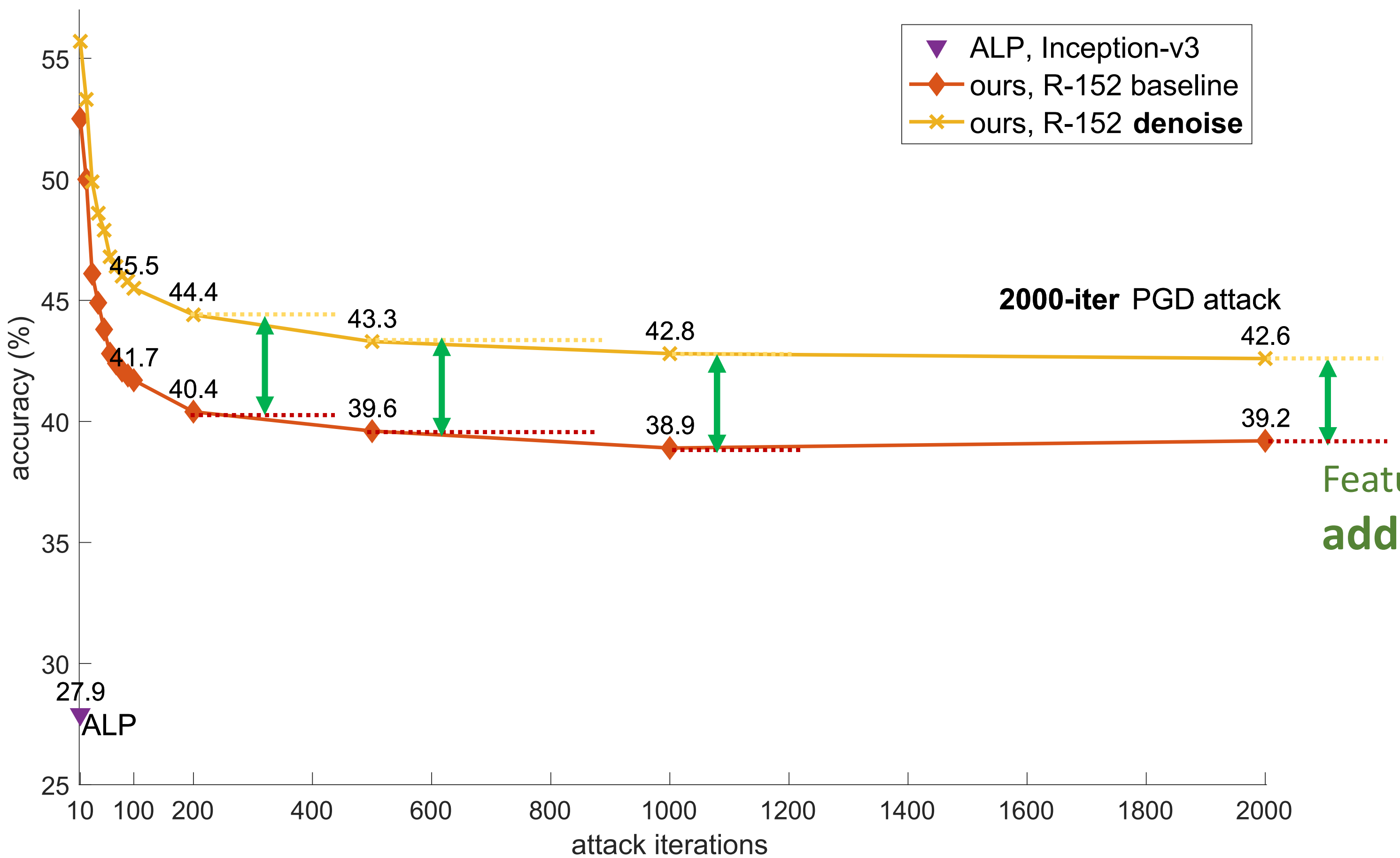
Defending Against White-box Attacks

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

Defending Against White-box Attacks – Part I

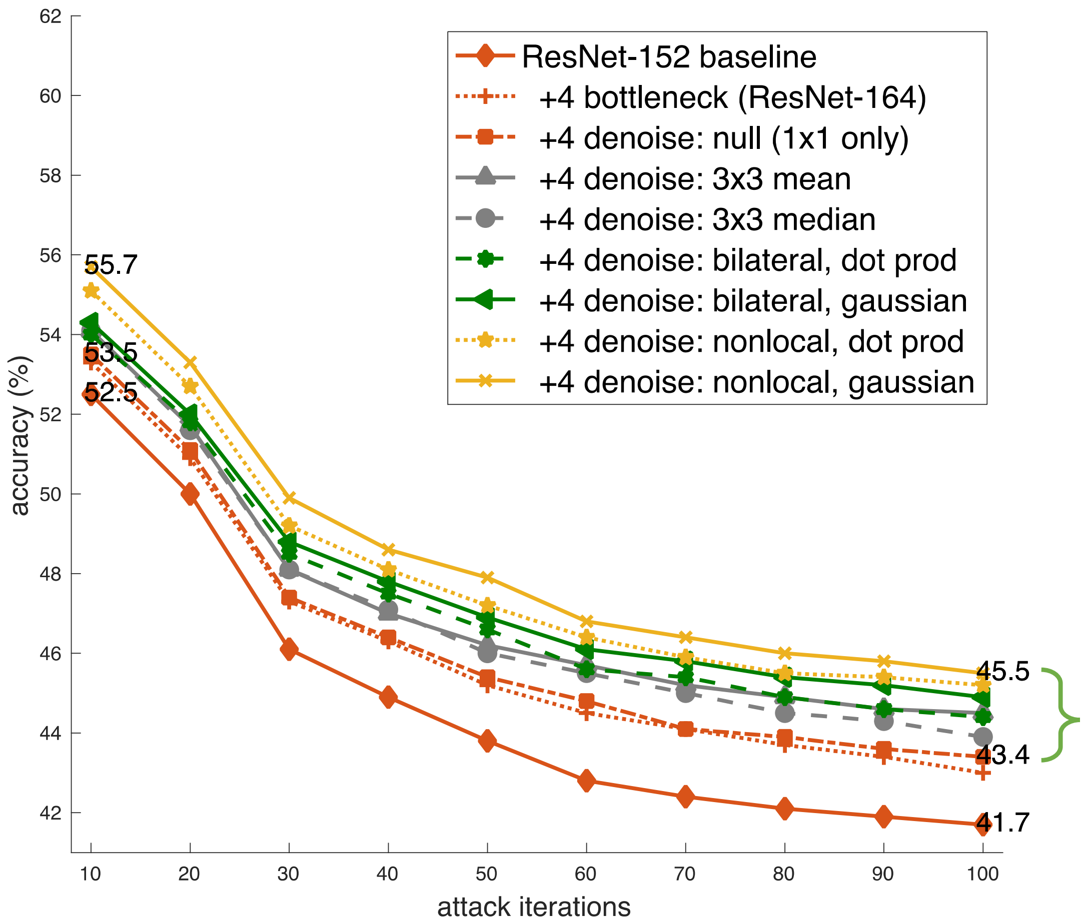


Defending Against White-box Attacks – Part I



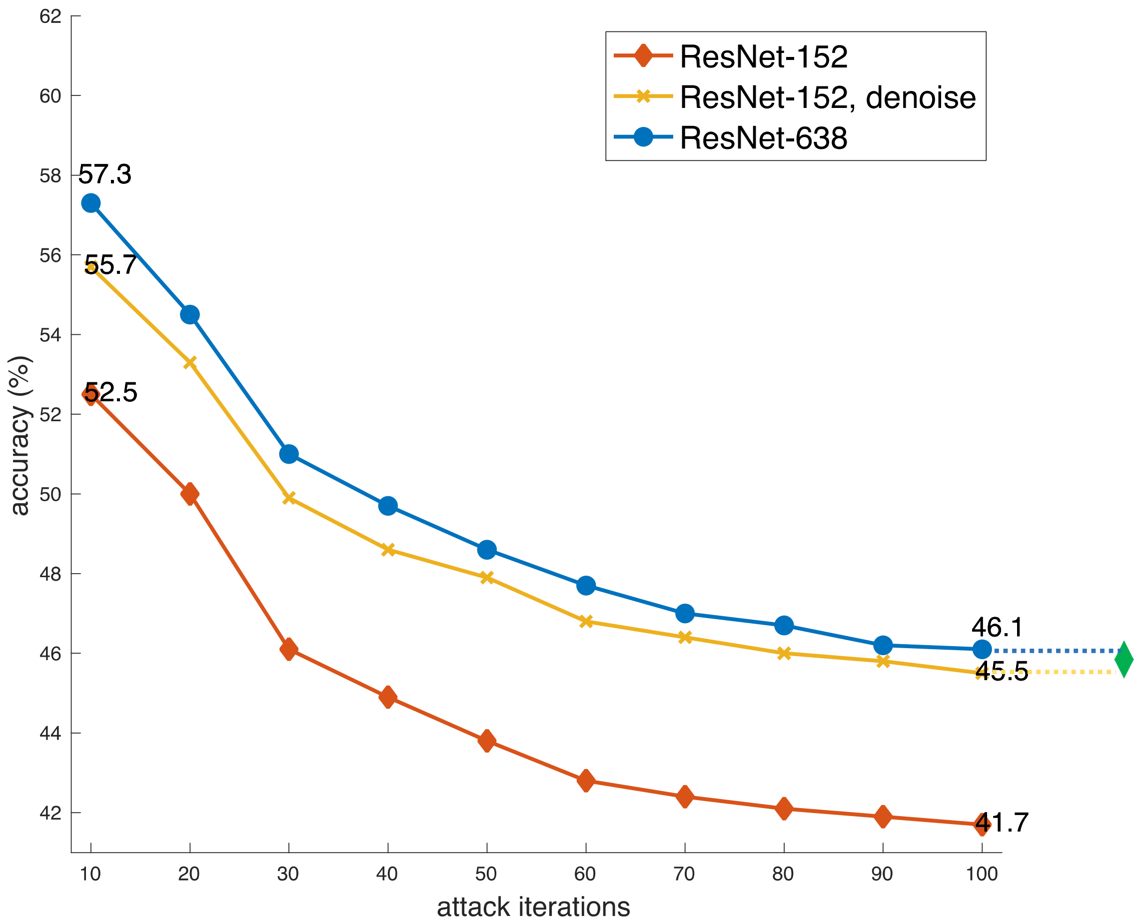
Feature Denoising can give us **additional** benefits

Defending Against White-box Attacks – Part II



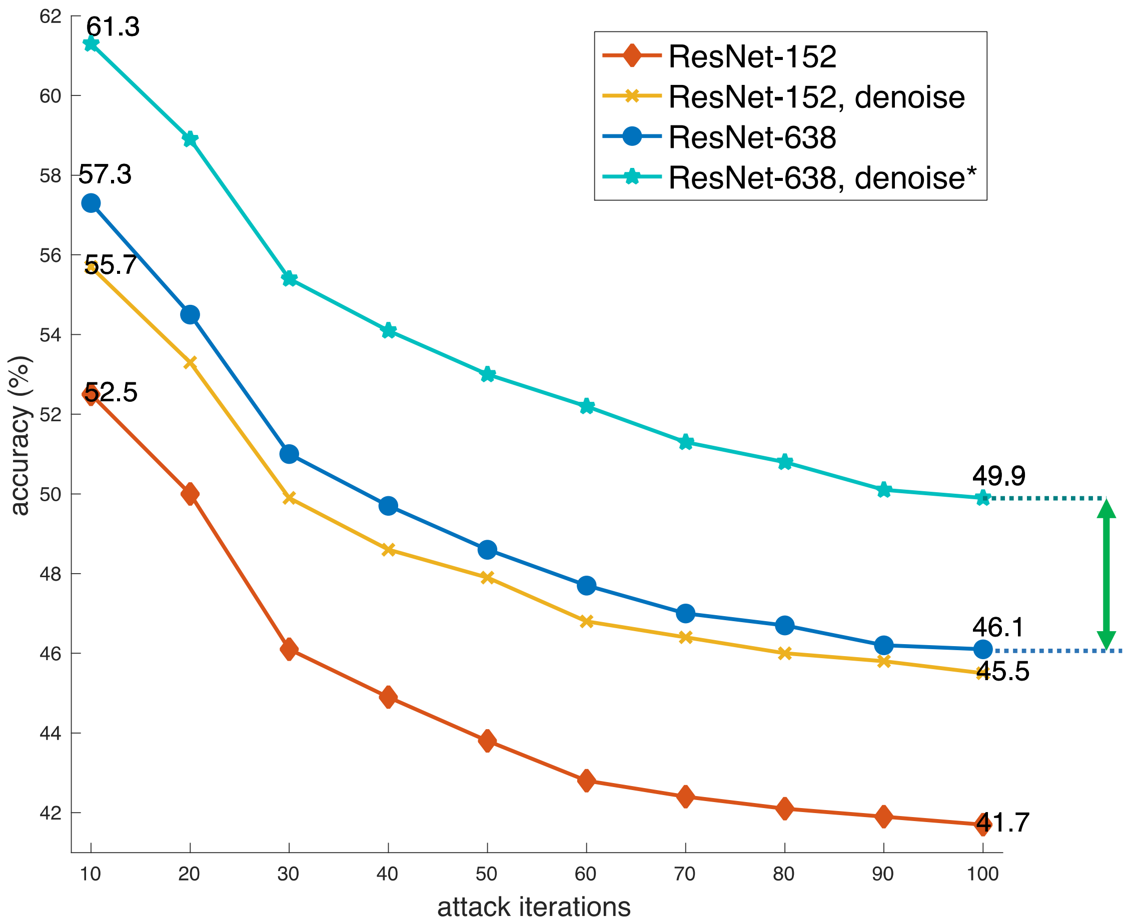
All denoising operations can help

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**

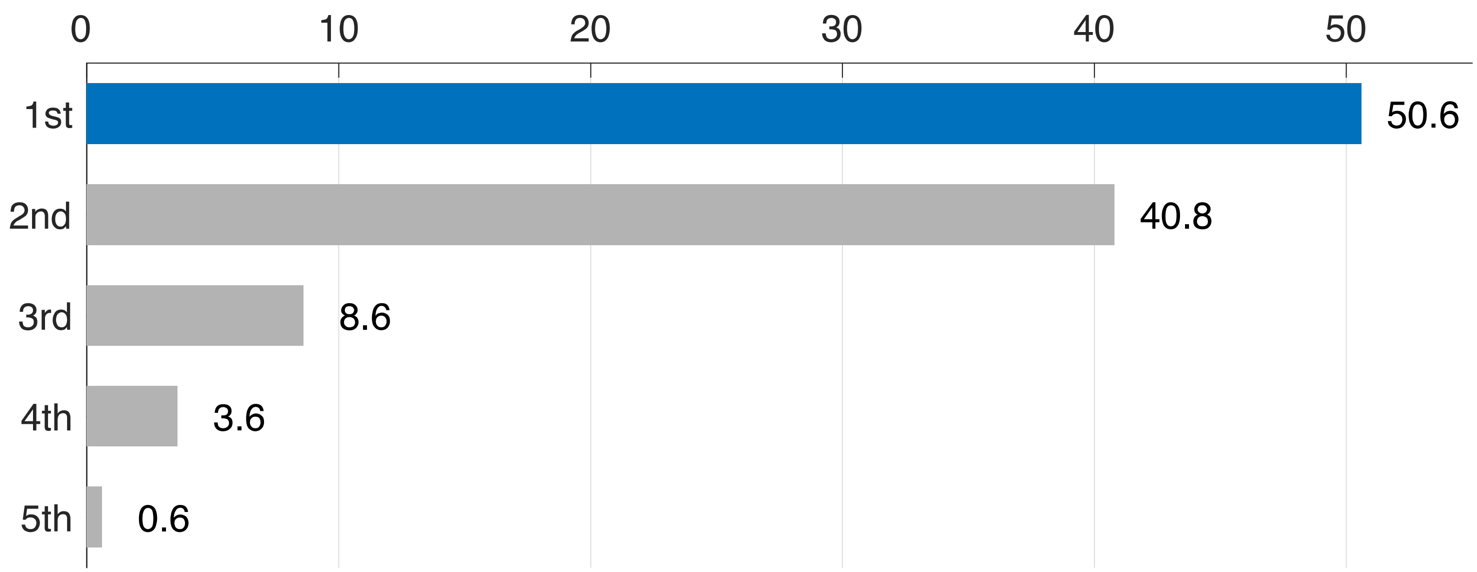
Defending Against Blind Attacks --- CAAD 2017 Offline Evaluation

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**

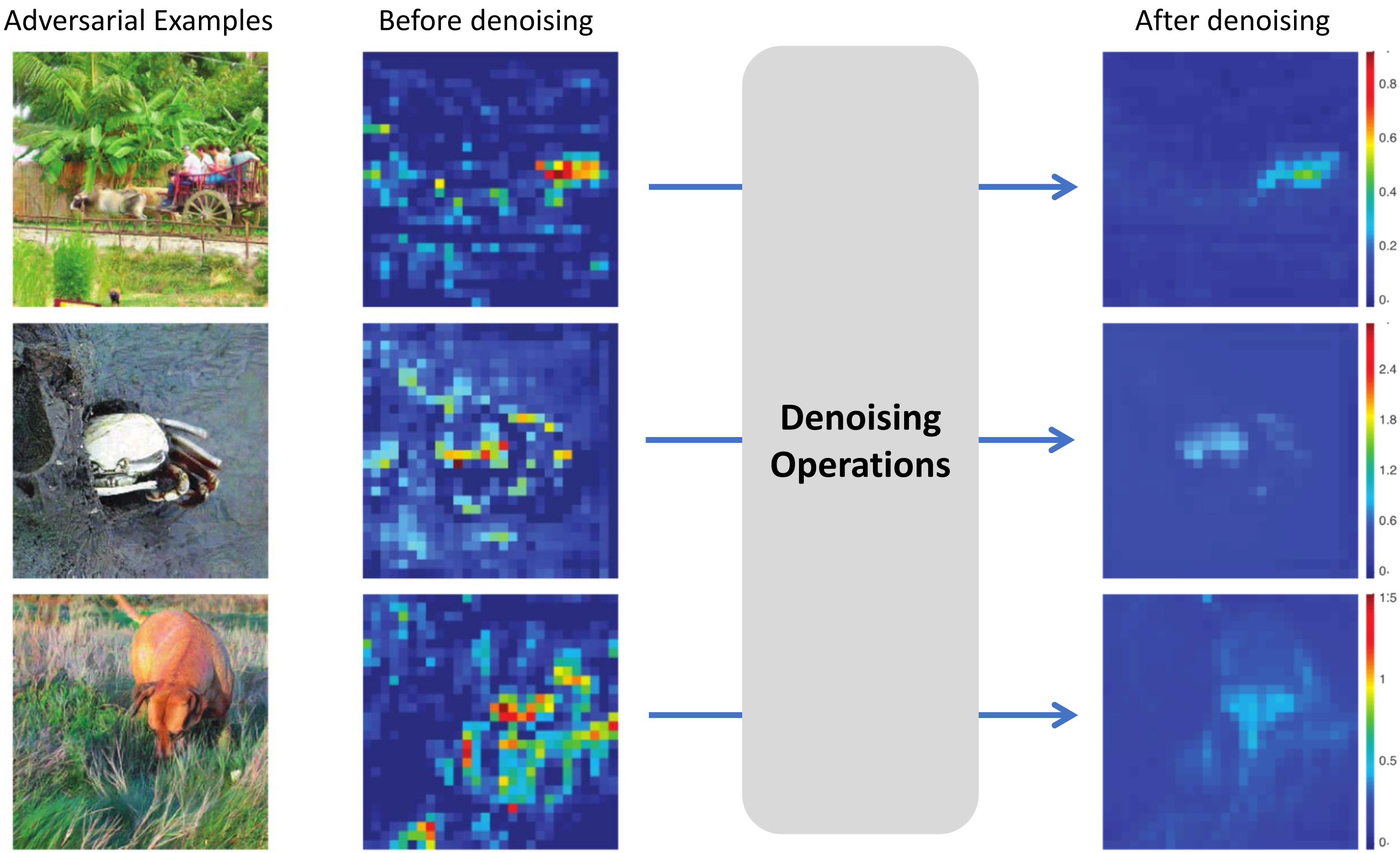
Defending Against Blind Attacks --- CAAD 2017 Offline Evaluation

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**

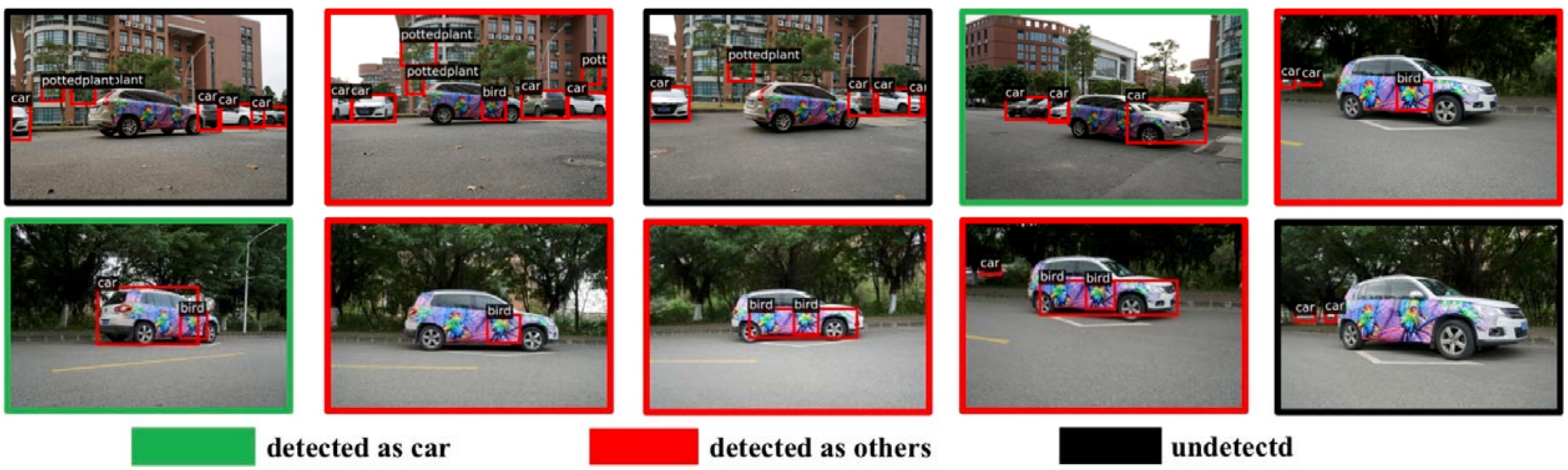
Defending Against Blind Attacks --- CAAD 2018 Online Competition

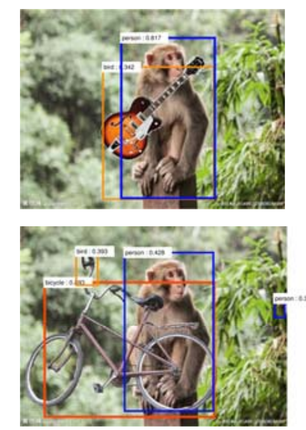


Visualization



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





Questions?