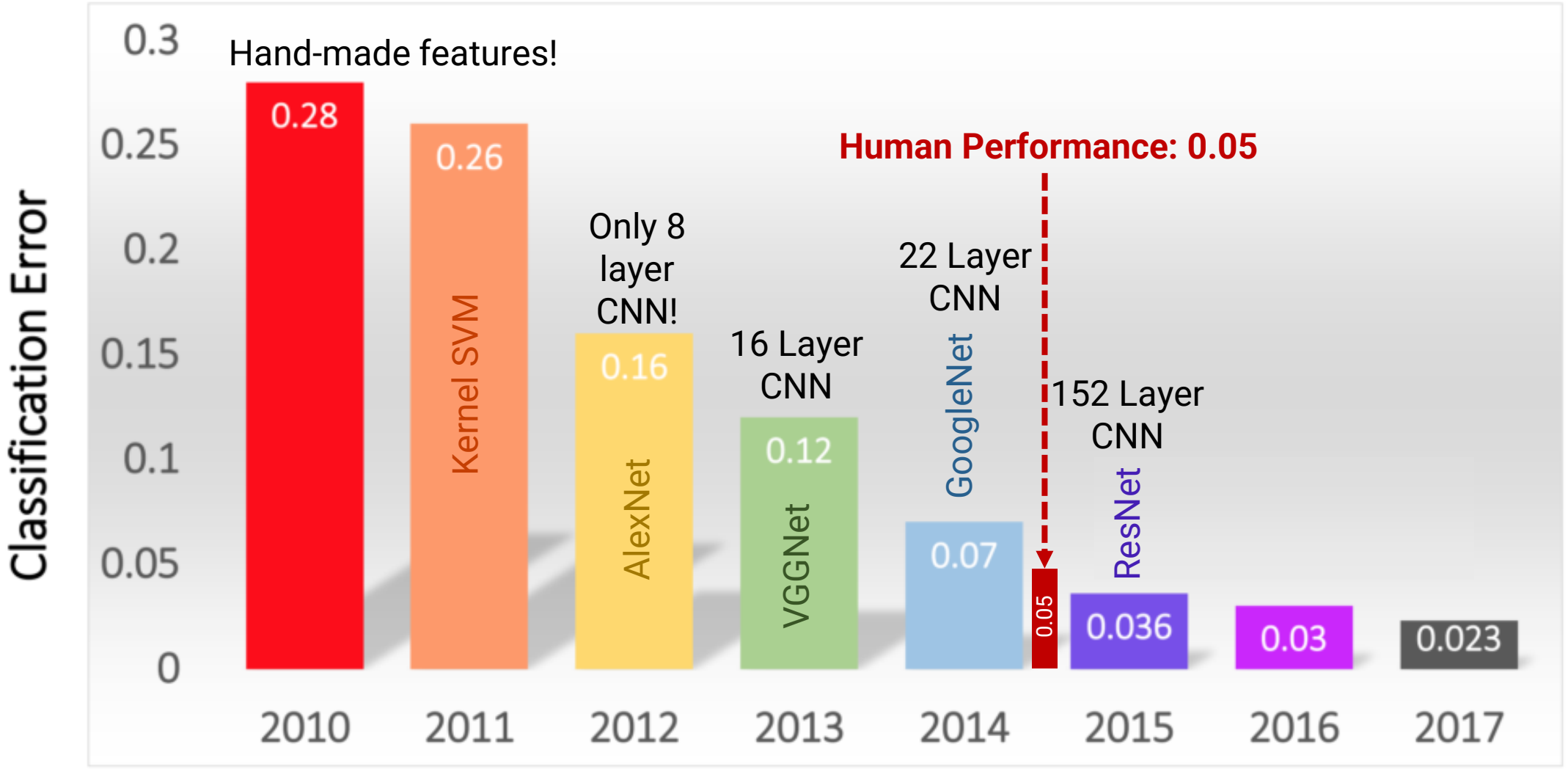
# RBE474X/595-B01-ST: Deep Learning For Perception

## Class 9: Single Pixel Attacks, Patch Based Attacks

Prof. Wei Xiao

# Do Deep Nets Generalize?




Slide idea from Stanford's CS W182/282A

# Do Deep Nets Generalize

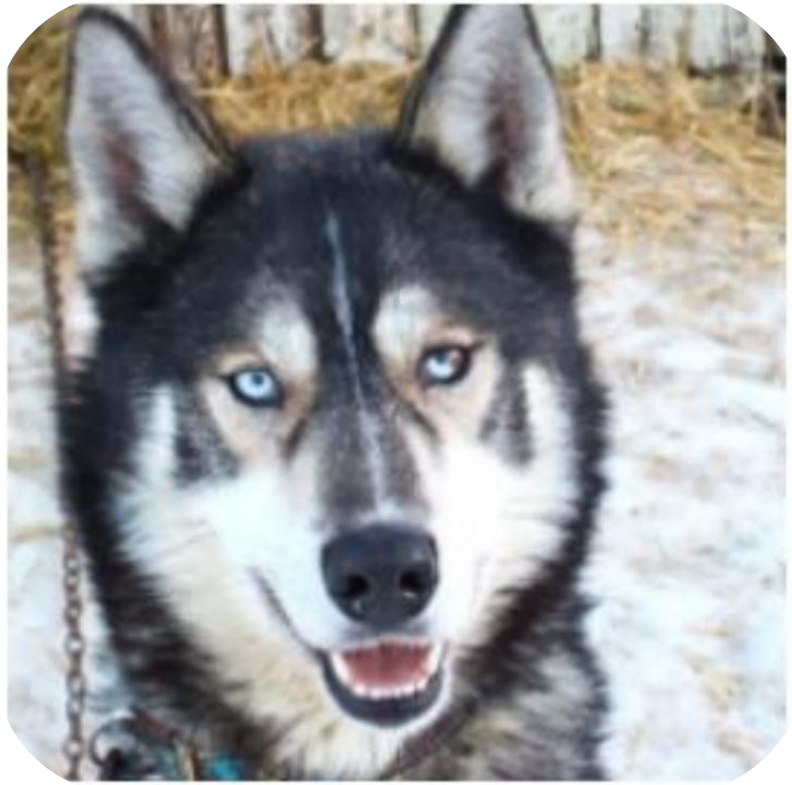
**The mistakes can make sense!**



PowerPoint Presentation (cs182sp21.github.io)

# Do Deep Nets Generalize?

**The mistakes can make sense!**



Husky classified as wolf

Explanation

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016.

# What Is The Issue?

When the training/test paradigm goes wrong!

Everything might be “working as intended”, but we might still not get what we want!

# A Major Problem!

## Distributional Shift

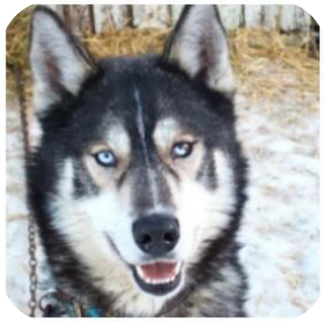



How Tesla Teaches Cars to Stop (roboflow.com)

# Calibration

Are in-distribution predictions calibrated?

Usually not, but there are many methods for improving calibration.



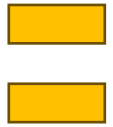
Magical Deep Learning

Husky: 0.3
Wolf: 0.7

"7/10 times, a person would say this is a wolf"

Magical Deep Learning

Husky: 0.5
Wolf: 0.5



"I don't know what this is!"

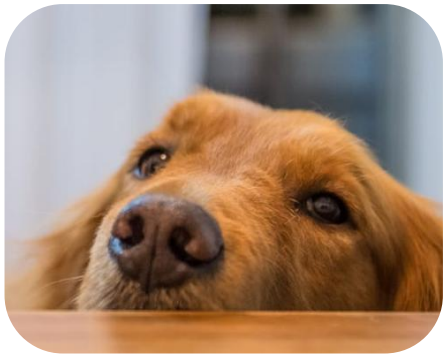
Depends on how data is generated and labelled!

Does this happen?

Usually not, such models typically give **confident** but **wrong** predictions on OOD inputs (but not always!)


Let’s Go On The Attack

# Let’s Make The Net Fail!



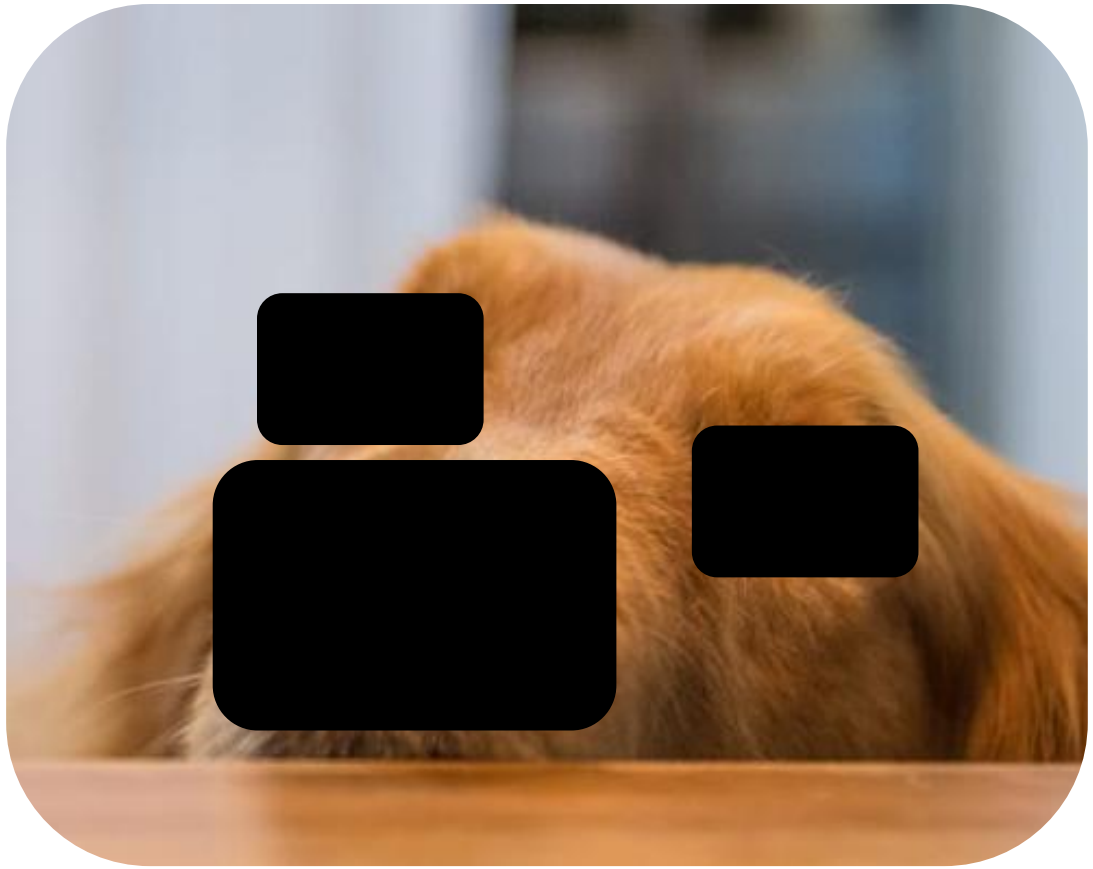
Magical Classifier

$p(dog|I)$

# Let’s Make The Net Fail!

Blackout

# Let's Make The Net Fail!



Blackout

# Let’s Make The Net Fail!

CutMix


Yun, Sangdoo, et al. "Cutmix: Regularization strategy to train strong classifiers with localizable features." Proceedings of the IEEE/CVF international conference on computer vision. 2019.

# Let’s Make The Net Fail!

Mixup


Zhang, Hongyi. "mixup: Beyond empirical risk minimization." arXiv preprint arXiv:1710.09412 (2017).

# What Is The Problem?

Too obvious to see something is wrong

Can it happen in reality?
Kind of, but rare!

# What Is This?

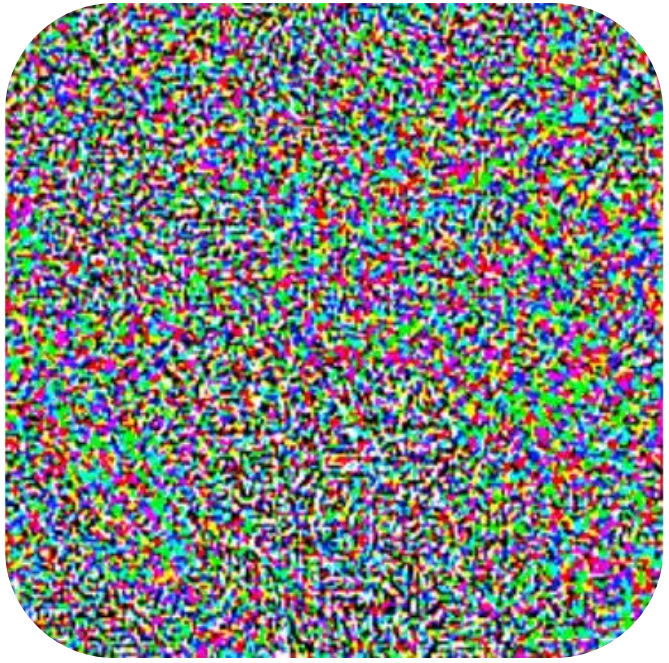
# What Is This?

Panda

−

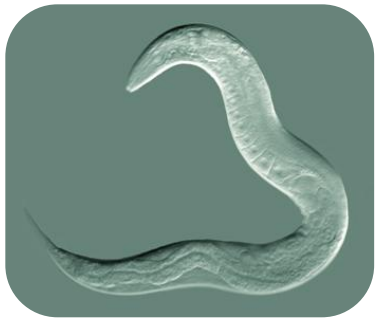
Gibbon

=



143× amplified Noise

Nematode



# Why Does This Matter?

Panda

+

0.007×

Nematode

=

Gibbon

Potential way to attack learned classifiers ☹
This is a smaller/direct issue.

What is the bigger issue?
This implies strange things about generalization!

# Adversarial Examples

Panda

+

0.007×

Nematode

=

Gibbon

We can turn **anything** to **anything else** with enough effort!

What does effort mean?

$$\|x_{adv} - x\|_p \leq \epsilon$$

Recall how we train:

$$\underset{\Theta}{\operatorname{argmin}}\, \mathcal{L}(\hat{y}, \tilde{y}, x \,|\Theta)$$

What does an adversary do?

$$\underset{\Theta}{\operatorname{argmax}}\, \mathcal{L}(\hat{y}, \tilde{y}, \hat{x} \,|\Theta)$$

But is this enough?
No! We need to have an attack budget!

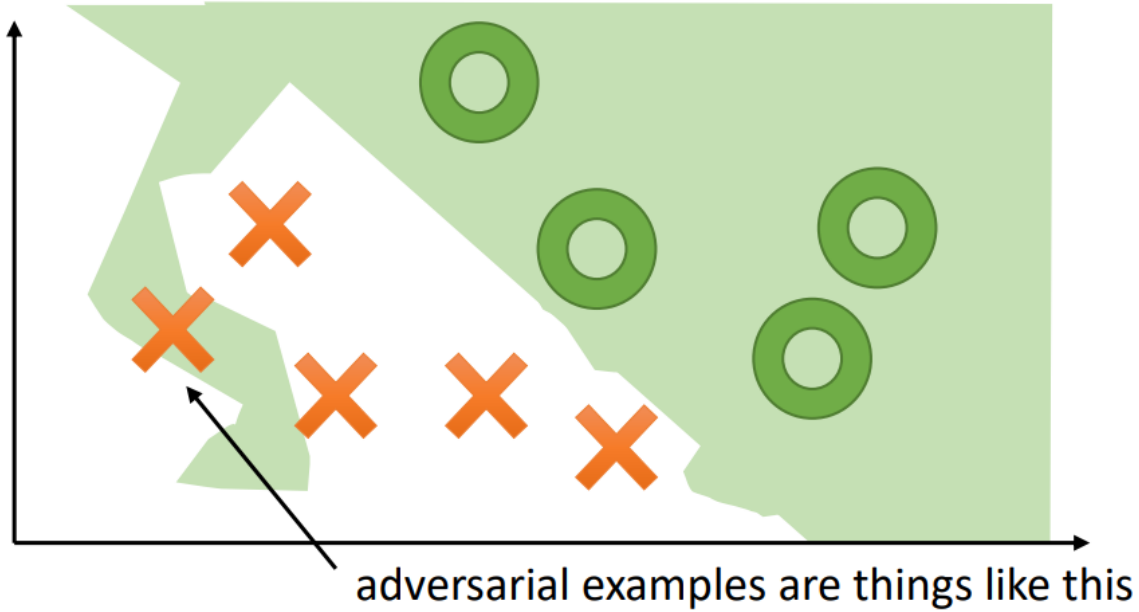
$$x_{adv} = x + \underset{\delta:\|\delta\|_p \leq \epsilon}{\operatorname{argmax}}\, \mathcal{L}(\hat{y}, \tilde{y}, x + \delta \,|\Theta)$$

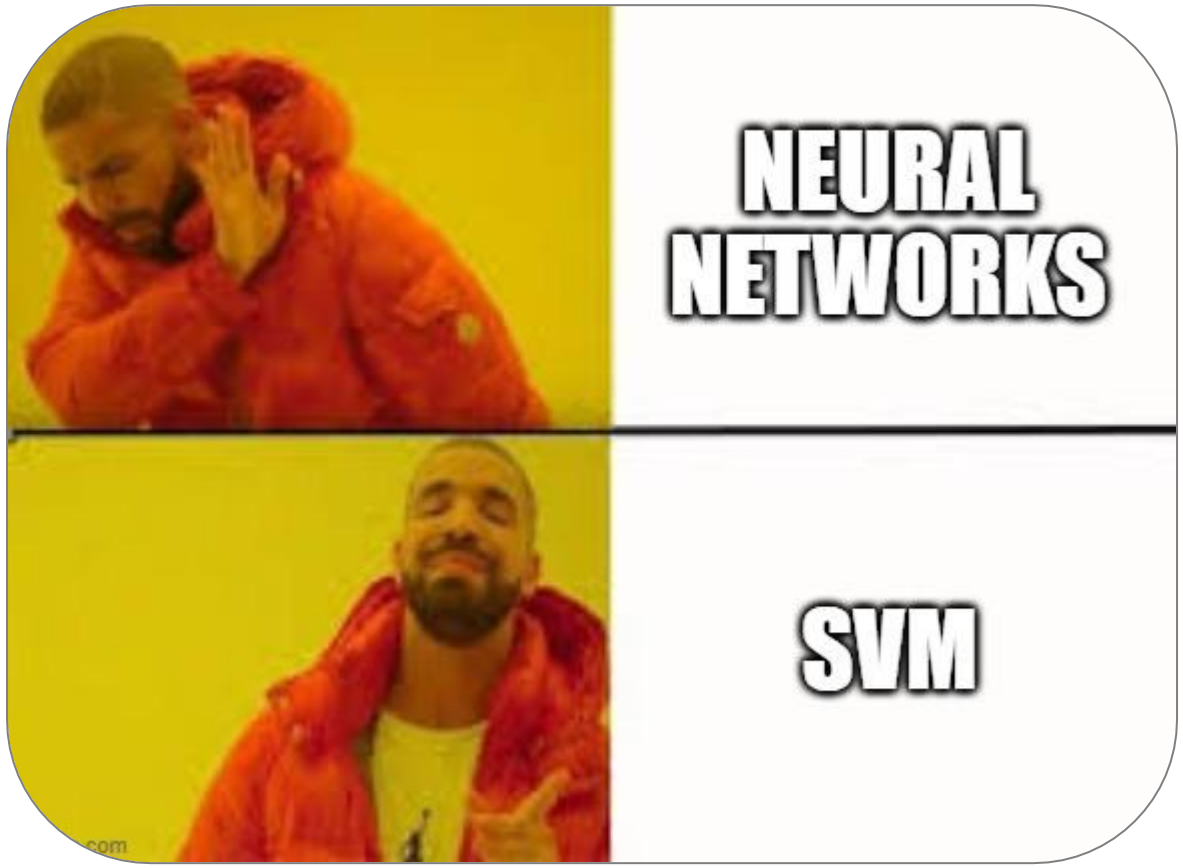
# Why Can Networks Be Attacked?

## Overfitting Hypothesis

NNs have a large number of parameters!
Hence, they overfit making them fragile!
**Stop using NNs?**

Mental model



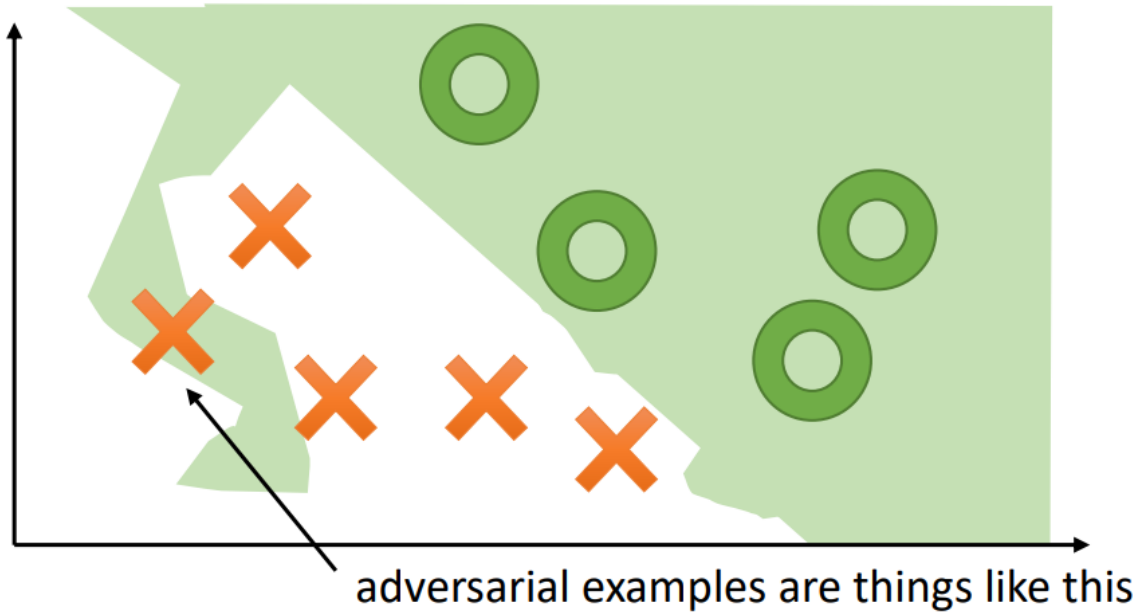
# Why Can Networks Be Attacked?

## Overfitting Hypothesis

NNs have a large number of parameters!
Hence, they overfit making them fragile!

**Stop using NNs?**

Mental model




**Most evidence suggests that this hypothesis is false!**

- If this were true, we would expect different models to have very different adversarial examples (high variance)
  - This is conclusively not the case
- If this were true, we would expect low-capacity models (e.g., linear models) not to have this issue
  - Low-capacity models also have this
- If this were true, we would expect highly nonlinear decision boundaries around adversarial examples
  - This appears to not be true

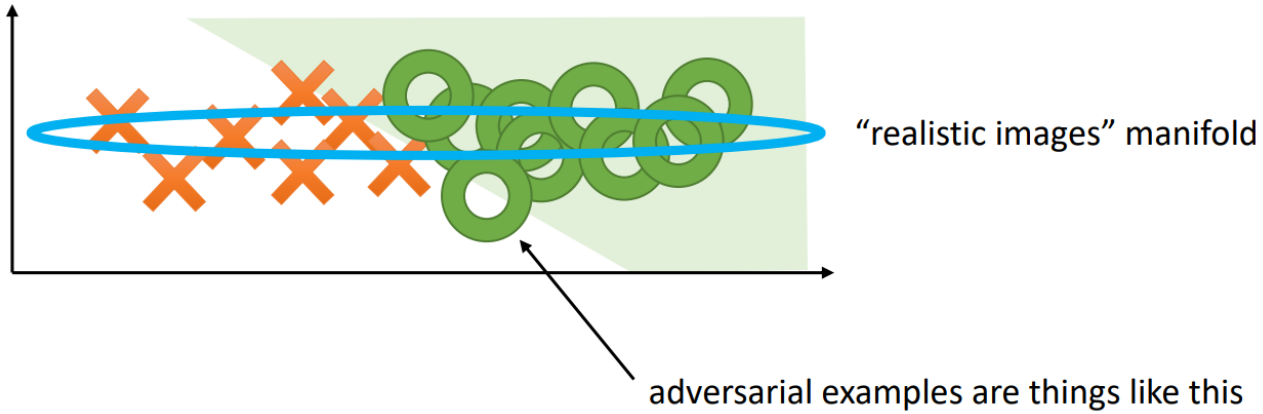
# Why Can Networks Be Attacked?

## Linear Models Hypothesis

NNs behave locally linear, they extrapolate in counterintuitive ways when **moving away** from data

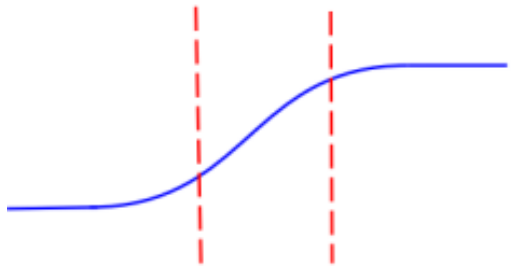
- Consistent with transferability of adversarial examples
- Reducing “overfitting” doesn’t fix the problem

Mental model



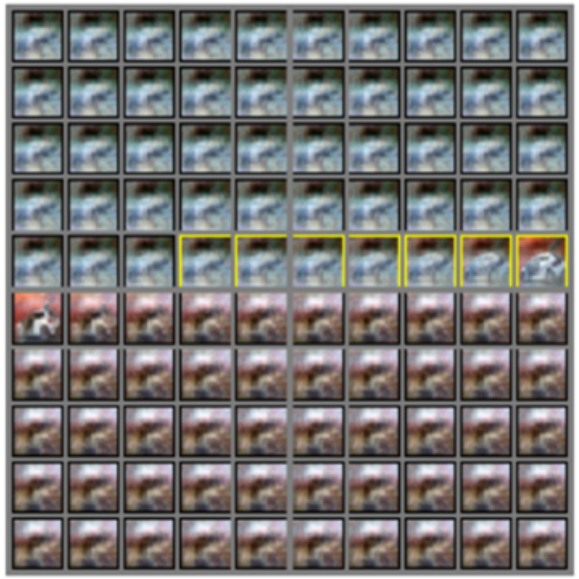

Rectified linear unit

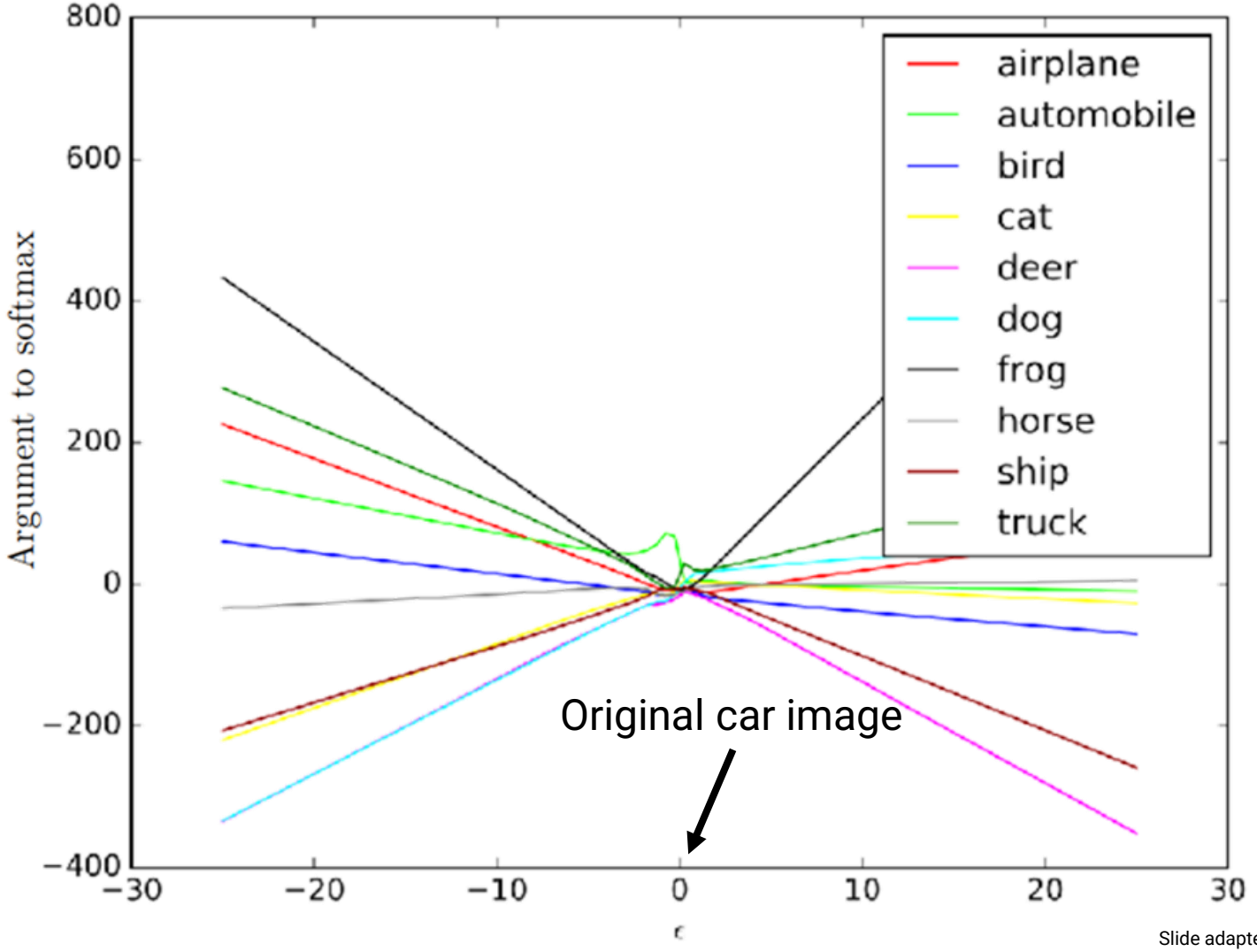
Carefully tuned sigmoid



# Why Can Networks Be Attacked?

## Linear Models Hypothesis 2

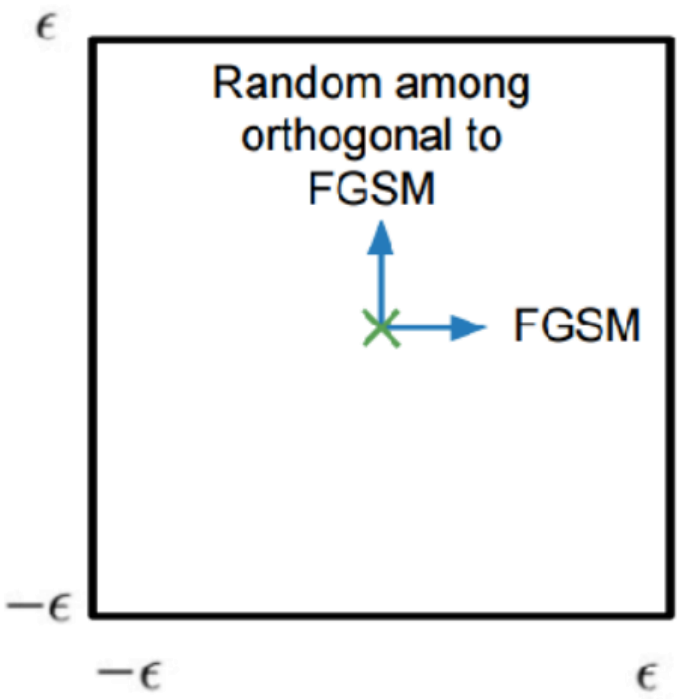




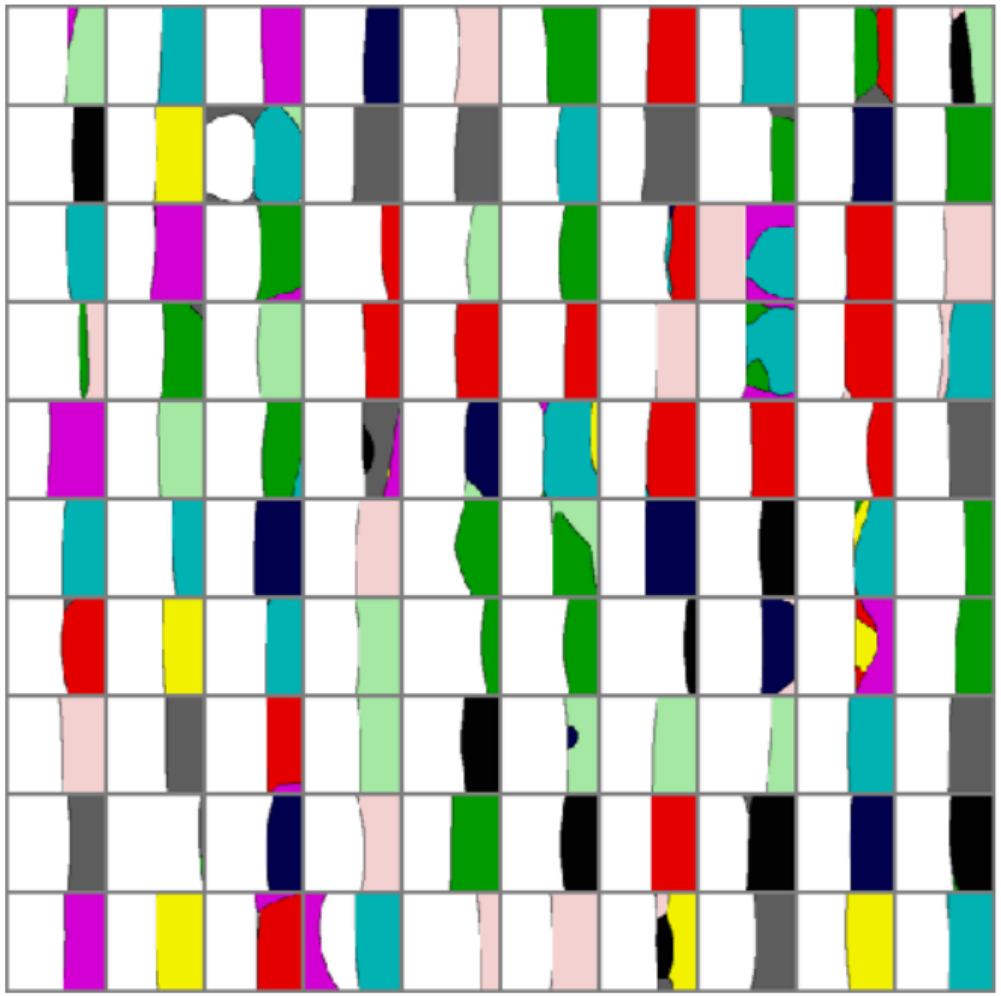

Slide adapted from Ian Goodfellow

# Why Can Networks Be Attacked?

## Linear Models Hypothesis




fast gradient sign method



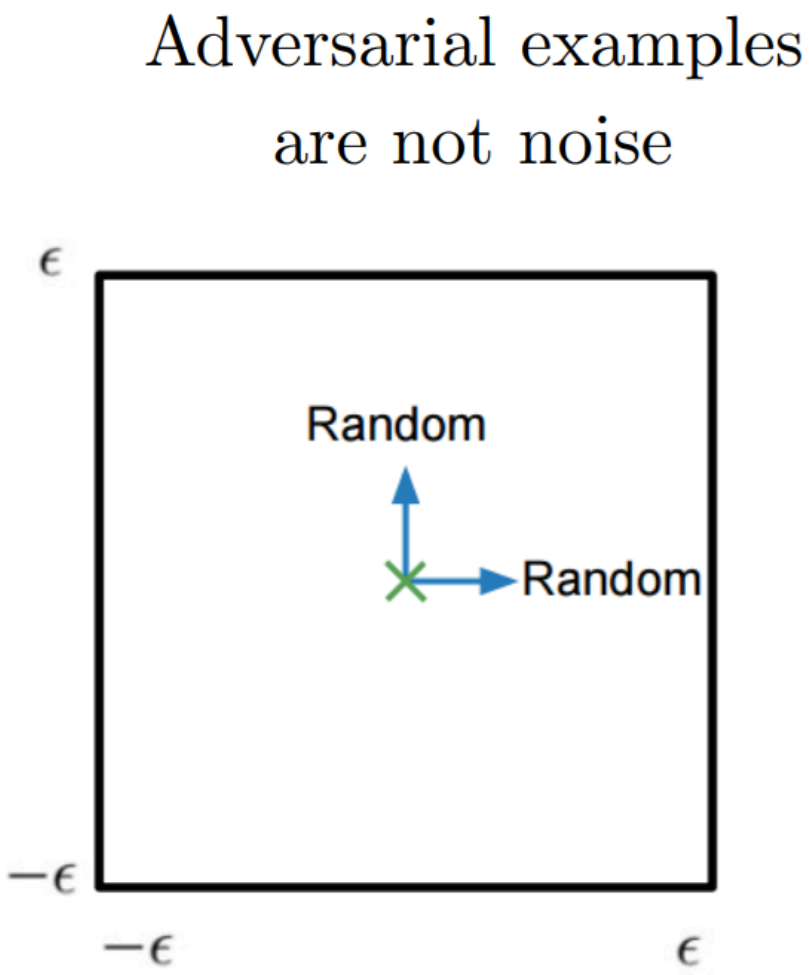
Not much variation **orthogonal** to adversarial direction!

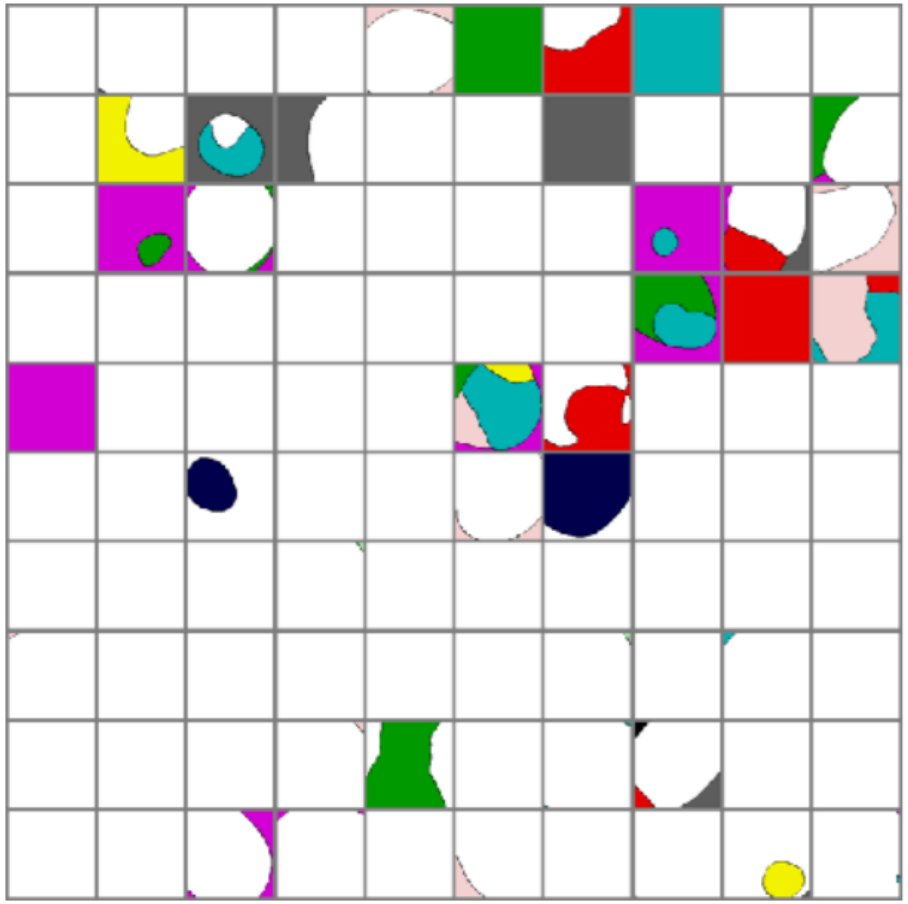
Clean "shift" on **one side** for adversarial direction, suggesting a mostly linear decision boundary

Lecture 16 | Adversarial Examples and Adversarial Training (youtube.com)

Slide adapted from Ian Goodfellow

# Can Noise Do The Same?



Lecture 16 | Adversarial Examples and Adversarial Training (youtube.com)

Slide adapted from Ian Goodfellow

# What Does This Have To Do With Generalization?

# Generalization

- Linear hypothesis is good for understanding NNs and when they will/will not work
- Classifier learns patterns not the concept
  - Extrapolation can be weird
  - Adversarial examples are features not bugs (Ilyas, Andrew, et al. "Adversarial examples are not bugs, they are features." *Advances in neural information processing systems* 32 (2019).)

- **NNs pay attention to "adversarial directions" because it helps them to get the right answer on the training data!**

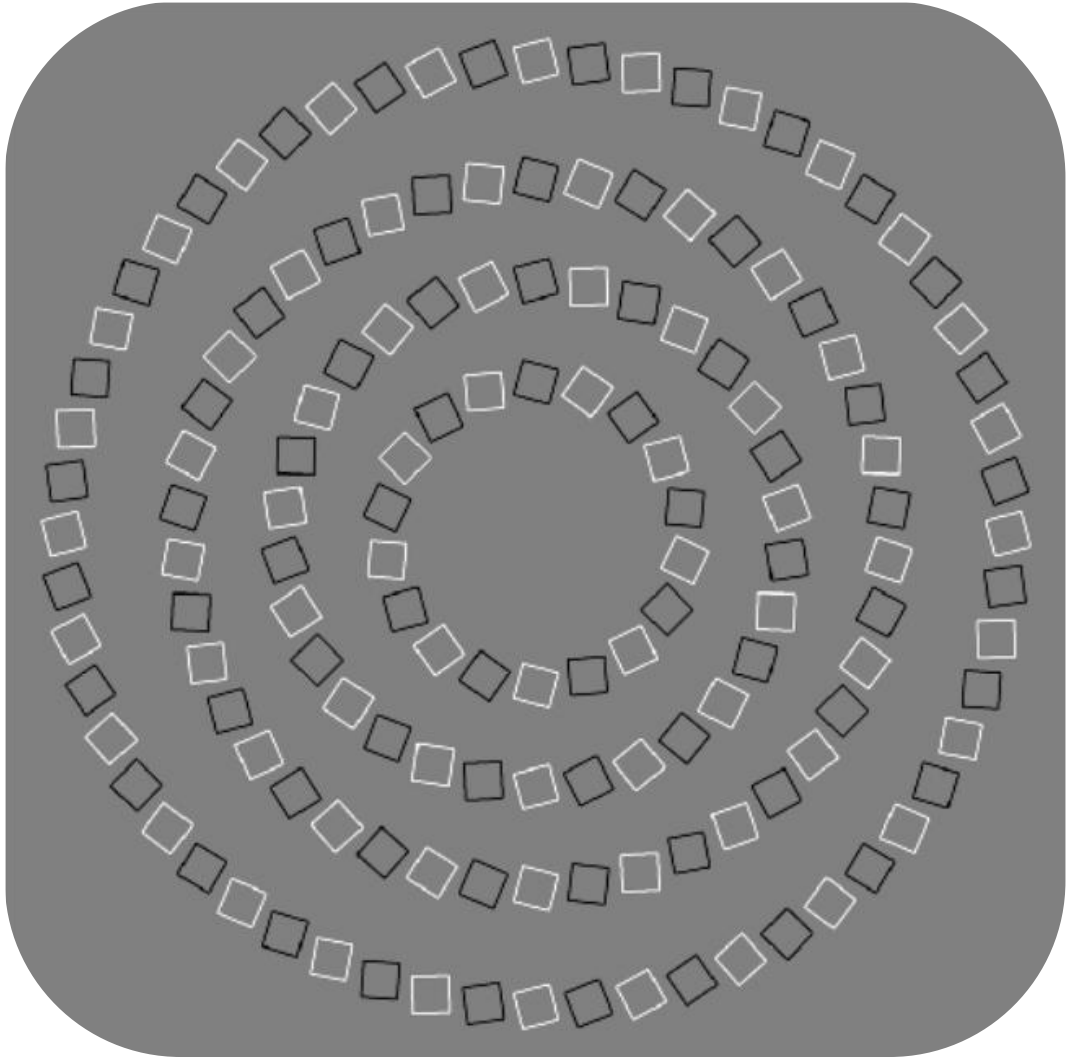
- As long as train and test distributions are same, NNs work great!
- They sometimes do this by being a **smart horse**!
  - We got to frame our problem better

- Outputs are often not well calibrated, especially OOD
- We can synthesize adversarial examples to fool an NN

- Adversaries are **not** due to overfitting
  - **locally linear boundaries**

These are concentric circles, not intertwined spirals.

# Are Humans Perfect?



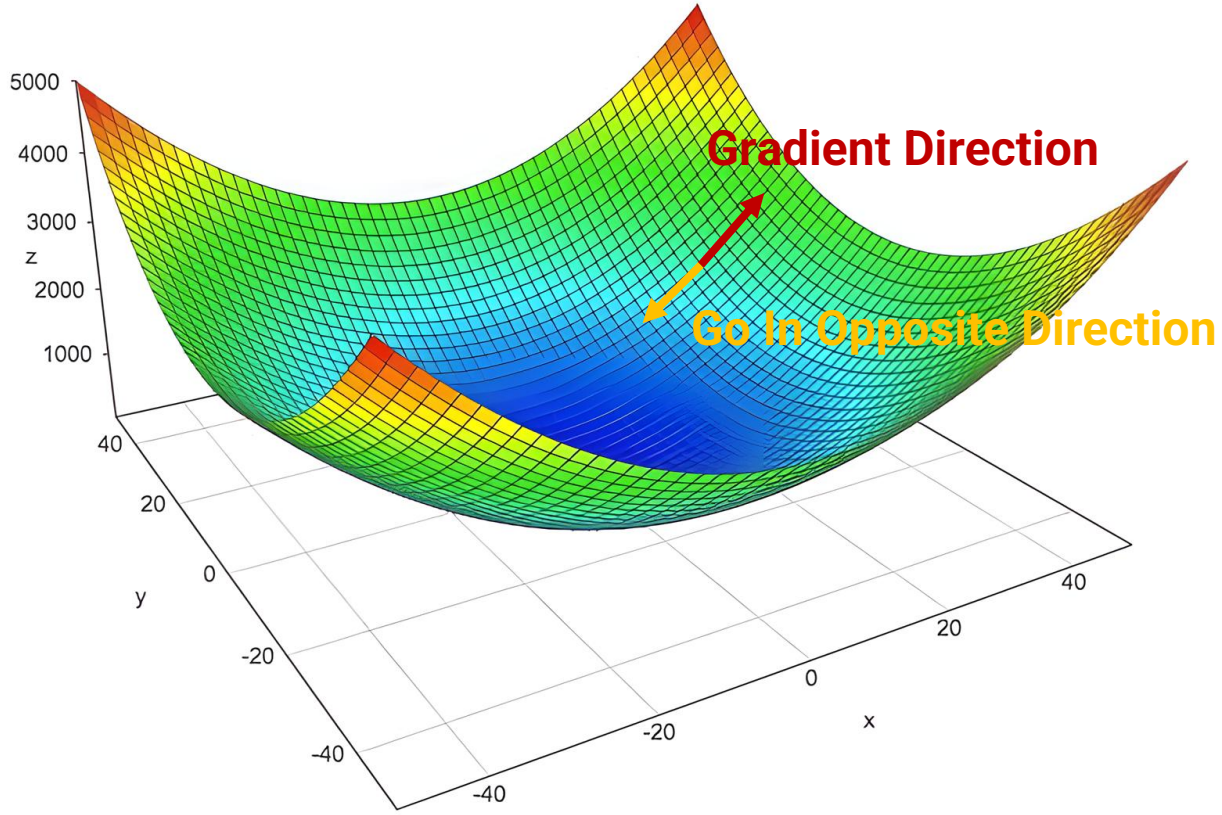
Pinna, Baingio, and Richard L. Gregory. "Shifts of edges and deformations of patterns." Perception (2002).


Back On The Attack

# Fast Gradient Sign Method

**FSGM**

**Recall Gradient Descent!**




$$x^{k+1} = x^k - \tau \nabla f\left(x^k\right)$$

Learning Rate!

# Fast Gradient Sign Method

**FSGM**

**Do Gradient Ascent!**

$$\mathcal{L}(x_{adv}, y) \approx \mathcal{L}(x, y) + (x_{adv} - x)^T \nabla_x \mathcal{L}$$
$$x_{adv} = x + \delta$$

Recall $x_{adv} = x + \underset{\delta:\|\delta\|_p \leq \epsilon}{\operatorname{argmax}} \mathcal{L}(\hat{y}, \tilde{y}, x + \delta \,|\Theta)$

Subject to $l_\infty$ attack budget of $\epsilon$ when $p = \infty$!

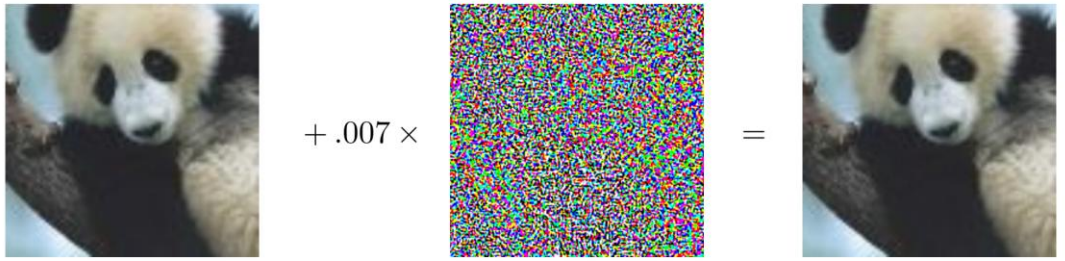
Issue is that it is dominated by one/few dimensions!
Better to use all of them ☺

$$x_{adv} = x + \epsilon \operatorname{sign}(\nabla_x \mathcal{L})$$

Why is this fast?
Only one step of Gradient Ascent!

This works very well against standard (naïve) neural nets!

It can be defeated with simple defenses, but more advanced attacks can be more resilient!




$\boldsymbol{x}$
"panda"
57.7% confidence

$\operatorname{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence

$\boldsymbol{x} + \epsilon \operatorname{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Chapter 3 - Adversarial examples, solving the inner maximization (adversarial-ml-tutorial.org)

Goodfellow, Ian J. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

# Projected Gradient Descent

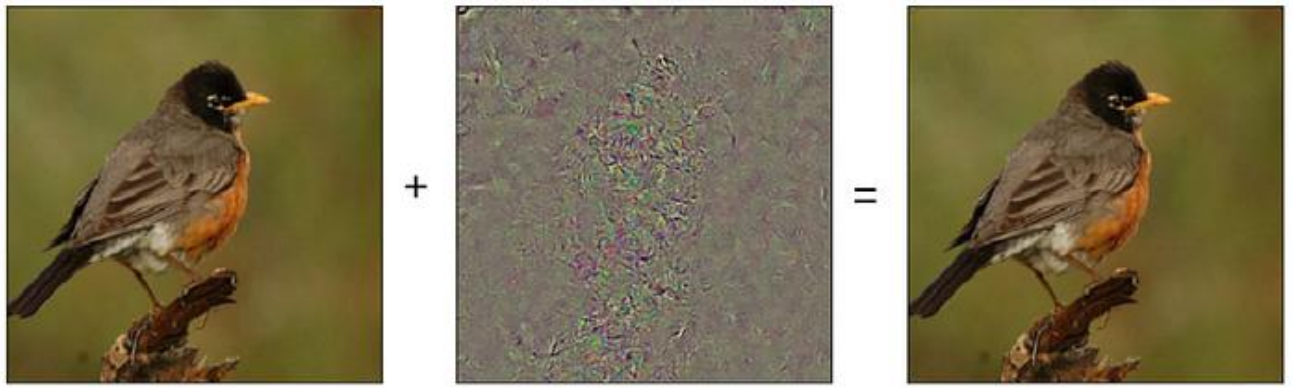
## PGD

$x_{adv} := x + n; n \sim \mathcal{U}[-\epsilon, \epsilon]$

For $t = 1, \cdots, T$:

$x_{adv} \coloneqq \mathcal{P}(x + \alpha \operatorname{sign}(\nabla_x \mathcal{L}(x + \delta, y \,|\Theta)))$

Where $\mathcal{P}(z) = \operatorname{clip}(z, x - \epsilon, x + \epsilon)$




Attack on ResNet50

xinyun_cvpr_advml_tutorial_2021 (advmlincv.github.io)

Mądry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. stat, 1050(9).


Does It Really Matter?

# How Do You Attack In Real World?

Environmental Robustness

Perturbation/Noise Robustness

# That’s Scary!

What a day. I get alerted this morning “reduced front camera visibility” msg 3rd day in a row. I am on the highway and the car applies the brakes heavily (I’m in the HOV lane) with no car in front but cars behind.

https://www.reddit.com/r/TeslaModelY/comments/1ey2mup/what_a_day_i_get_alerted_this_morning_reduced/

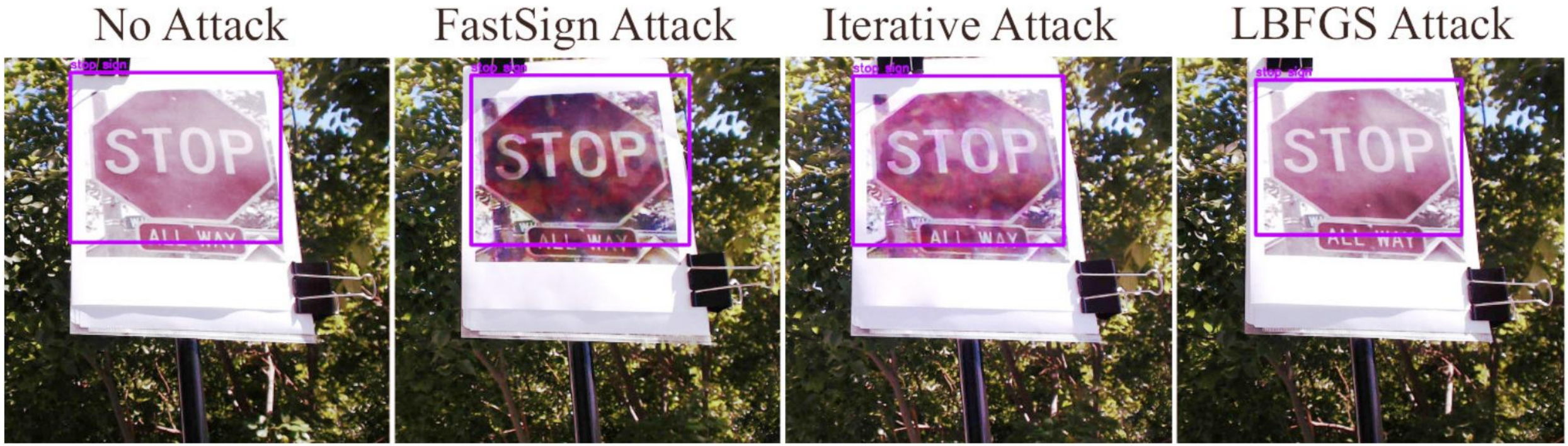
# That’s Scary!

What a day. I get alerted this morning “reduced front camera visibility” msg 3rd day in a row. I am on the highway and the car applies the brakes heavily (I’m in the HOV lane) with no car in front but cars behind.

**Then later I see why; fly in the front camera array…**

https://www.reddit.com/r/TeslaModelY/comments/1ey2mup/what_a_day_i_get_alerted_this_morning_reduced/

# How Do You Attack In Real World?




"Adversarial perturbation methods applied to **stop sign detection** only work in carefully chosen situations, and our preliminary experiment shows that we might **not need to worry** about it in many real circumstances, specifically with autonomous vehicles."

[Attacking Optical Flow (youtube.com)]

Slide adapted from Andreas Geiger

Lu, Jiajun, et al. "No need to worry about adversarial examples in object detection in autonomous vehicles." arXiv preprint arXiv:1707.03501 (2017).

# Robust Adversarial Attacks




- Demonstrate existence of **robust adversarial examples** in the physical world
- Maximize expectation over transformation $\mathcal{T}$ (EOT):

$$\underset{x_{adv}}{\operatorname{argmax}}\, \mathbb{E}_{t\sim\mathcal{T}}[\log P(y|t(x_{adv})) - \lambda\|t(x_{adv}) - x\|_2]$$

- Larger distributions require larger perturbations

Athalye, Anish, et al. "Synthesizing robust adversarial examples." International conference on machine learning. PMLR, 2018.

# Can Attacks Look Non-Obvious?


**Robust adversarial example** designed to mimic “graffiti”

Eykholt, Kevin, et al. "Robust physical-world attacks on deep learning visual classification." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

This paper appears at CVPR 2018

# Robust Physical-World Attacks on Deep Learning Visual Classification

Kevin Eykholt*[1], Ivan Evtimov*[2], Earlence Fernandes[2], Bo Li[3],
Amir Rahmati[4], Chaowei Xiao[1], Atul Prakash[1], Tadayoshi Kohno[2], and Dawn Song[3]

[1]University of Michigan, Ann Arbor
[2]University of Washington
[3]University of California, Berkeley
[4]Samsung Research America and Stony Brook University



## Abstract

*Recent studies show that the state-of-the-art deep neural networks (DNNs) are vulnerable to adversarial examples, resulting from small-magnitude perturbations added to the input. Given that that emerging physical systems are using DNNs in safety-critical situations, adversarial examples could mislead these systems and cause dangerous situations. Therefore, understanding adversarial examples in the physical world is an important step towards developing resilient learning algorithms. We propose a general attack algorithm, Robust Physical Perturbations ($RP_2$), to generate robust visual adversarial perturbations under different physical conditions. Using the real-world case of road sign classification, we show that adversarial examples generated using $RP_2$ achieve high targeted misclassification rates against standard-architecture road sign classifiers in the physical world under various environmental conditions, including viewpoints. Due to the current lack of a standardized testing method, we propose a two-stage evaluation methodology for robust physical adversarial examples consisting of lab and field tests. Using this methodology, we evaluate the efficacy of physical adversarial manipulations on real objects. With a perturbation in the form of only black and white stickers, we attack a real stop sign, causing targeted misclassification in 100% of the images obtained in lab settings, and in 84.8% of the captured video frames obtained on a moving vehicle (field test) for the target classifier.*

## 1. Introduction

Deep Neural Networks (DNNs) have achieved state-of-the-art, and sometimes human-competitive, performance on many computer vision tasks [11, 14, 36]. Based on these successes, they are increasingly being used as part of control pipelines in physical systems such as cars [8, 17], UAVs [4, 24], and robots [40]. Recent work, however, has demonstrated that DNNs are vulnerable to adversarial perturbations [5, 9, 10, 15, 16, 22, 25, 29, 30, 35]. These carefully crafted modifications to the (visual) input of DNNs can cause the systems they control to misbehave in unexpected and potentially dangerous ways.

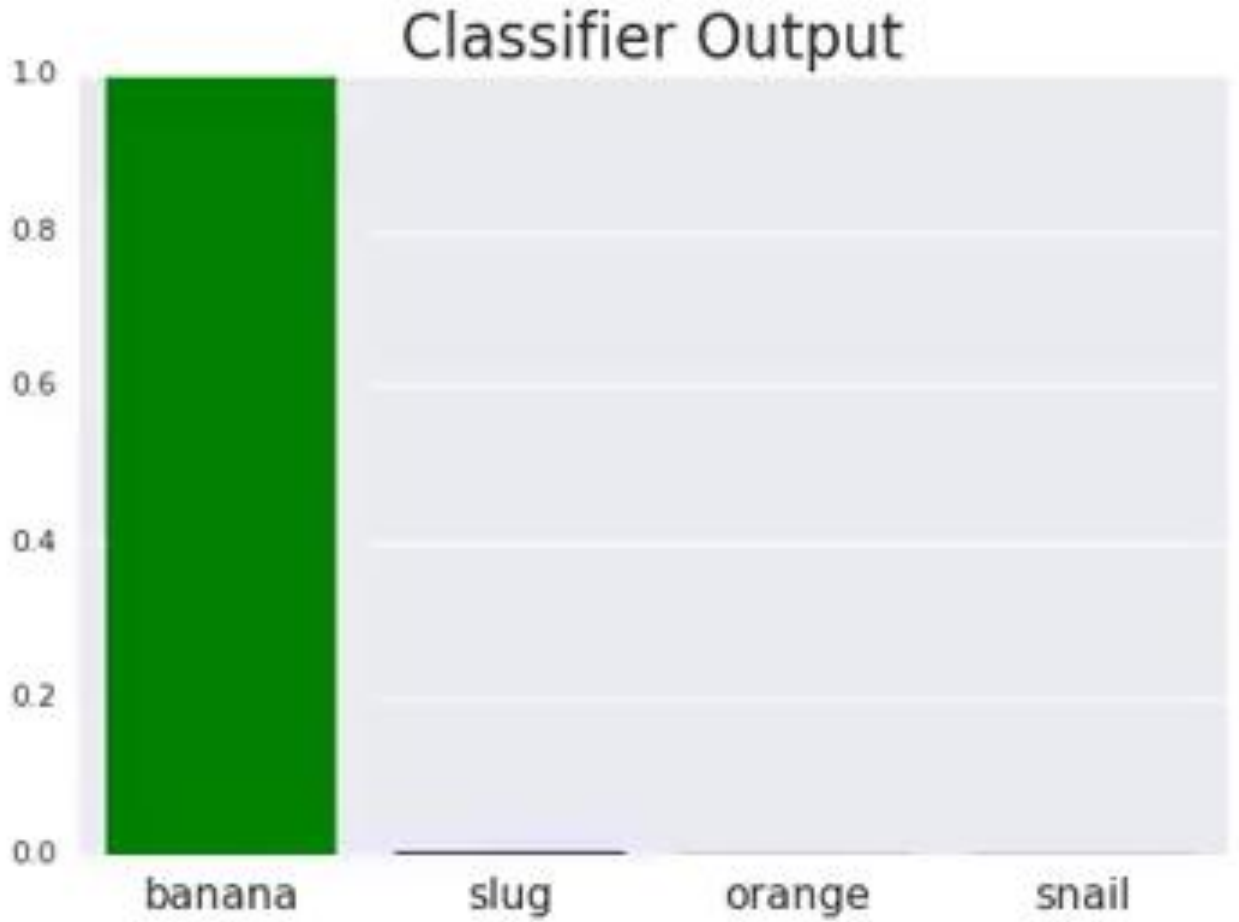
This threat has gained recent attention, and work in computer vision has made great progress in understanding the space of adversarial examples, beginning in the digital domain (*e.g.* by modifying images corresponding to a scene) [9, 22, 25, 35], and more recently in the physical domain [1, 2, 13, 32]. Along similar lines, our work contributes to the understanding of adversarial examples when perturbations are physically added to the *objects themselves*. We choose road sign classification as our target domain for several reasons: (1) The relative visual simplicity of road signs make it challenging to hide perturbations. (2) Road signs exist in a noisy unconstrained environment with changing physical conditions such as the distance and angle of the viewing camera, implying that physical adversarial perturbations should be robust against considerable environmental instability. (3) Road signs play an important role in transportation safety. (4) A reasonable threat model for transportation is that an attacker might not have control over a vehicle's systems, but is able to modify the objects in the physical world that a vehicle might depend on to make crucial safety decisions.

The main challenge with generating robust physical perturbations is environmental variability. Cyber-physical systems operate in noisy physical environments that can destroy perturbations created using current digital-only algorithms [19]. For our chosen application area, the most dynamic environmental change is the distance and angle of

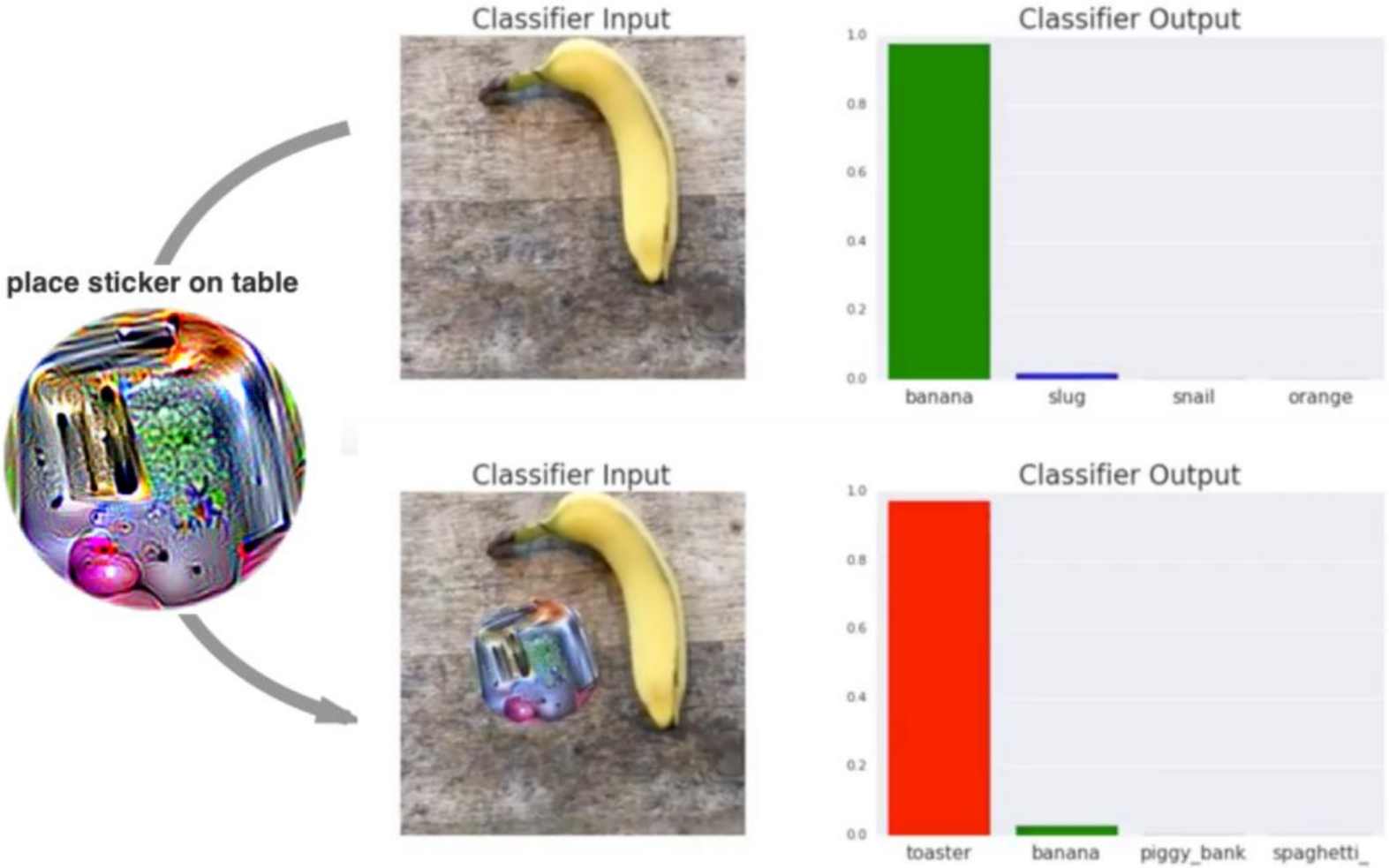
*These authors contributed equally.

# Microwaves As Phones!

# Patch-Based Attacks

# Patch-Based Attacks




- Use EOT (Expectation over Transformation) idea but generalize across images!
- Easy to apply in real-world settings (attaching patch to an object)

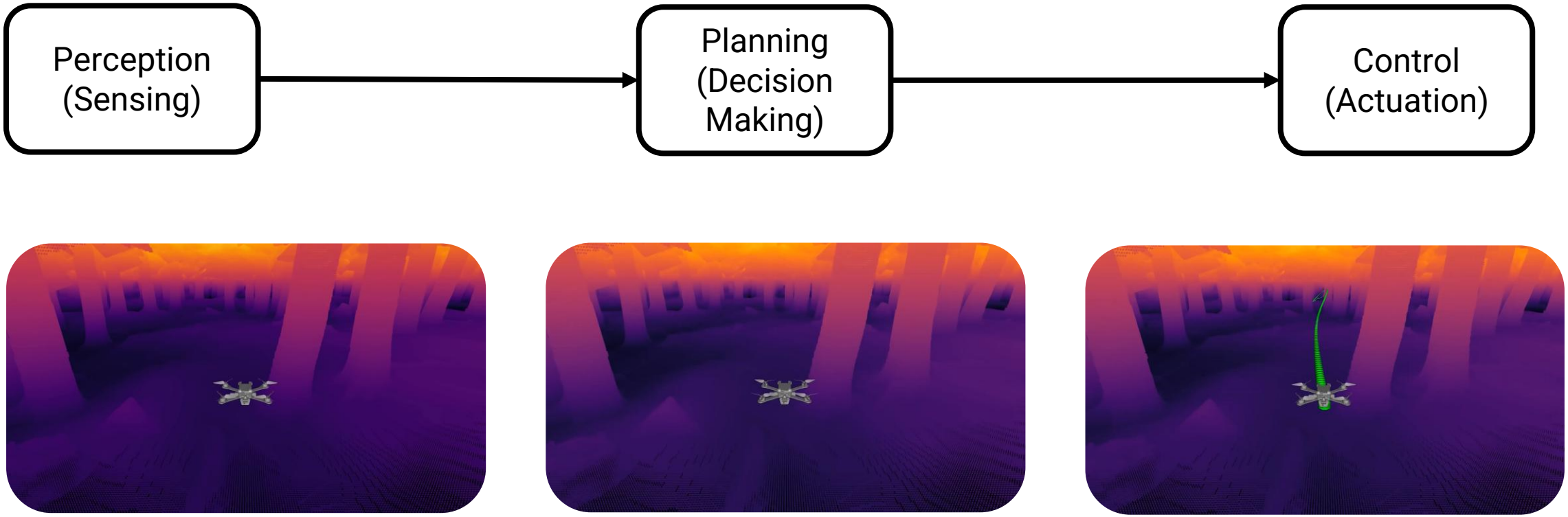
Brown, T. B., Mané, D., Roy, A., Abadi, M., & Gilmer, J. (2017). Adversarial patch. arXiv preprint arXiv:1712.09665.

# Patch-Based Attacks

A(  ,  , location, rotation, scale,... ) =

# Robot

**Ph.D. version**




Loquercio, Antonio, Elia Kaufmann, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. "Learning high-speed flight in the wild." *Science Robotics* 6, no. 59 (2021).

# You Have To Recursively Perceive!

## Consider A Toy Example

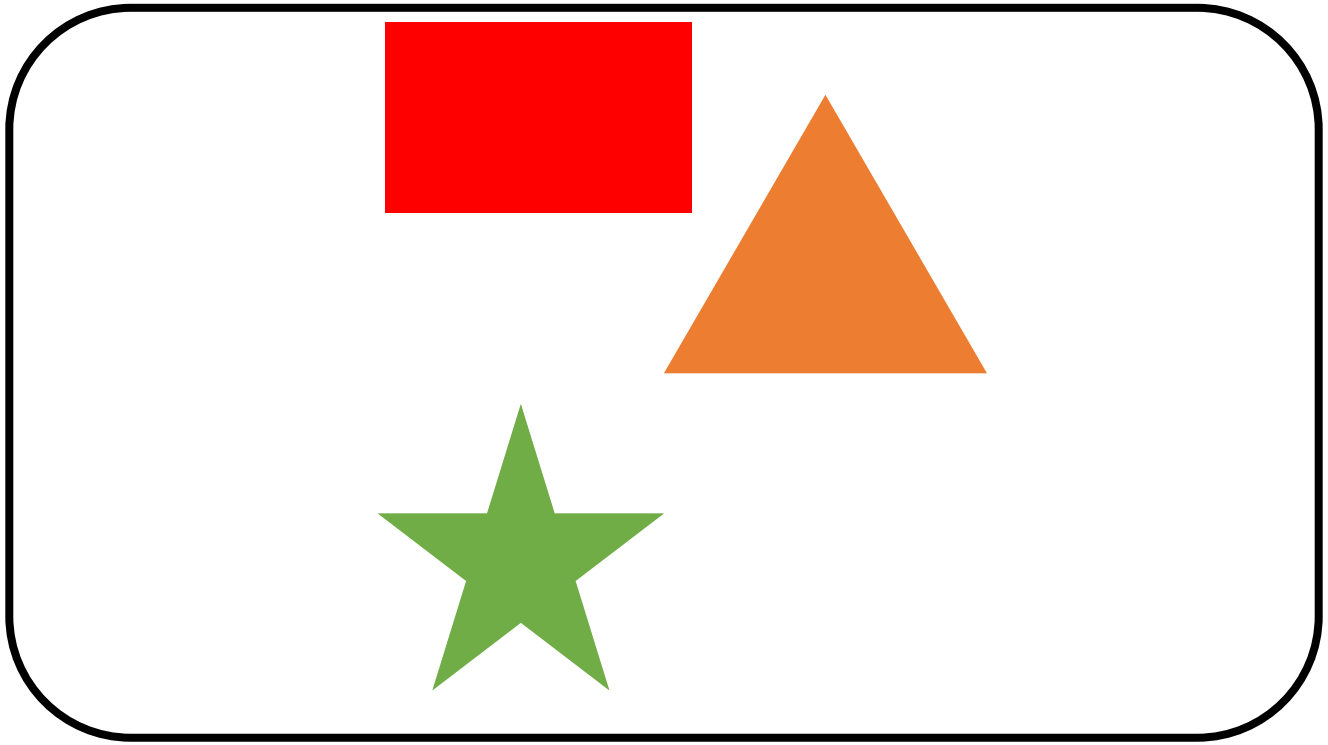
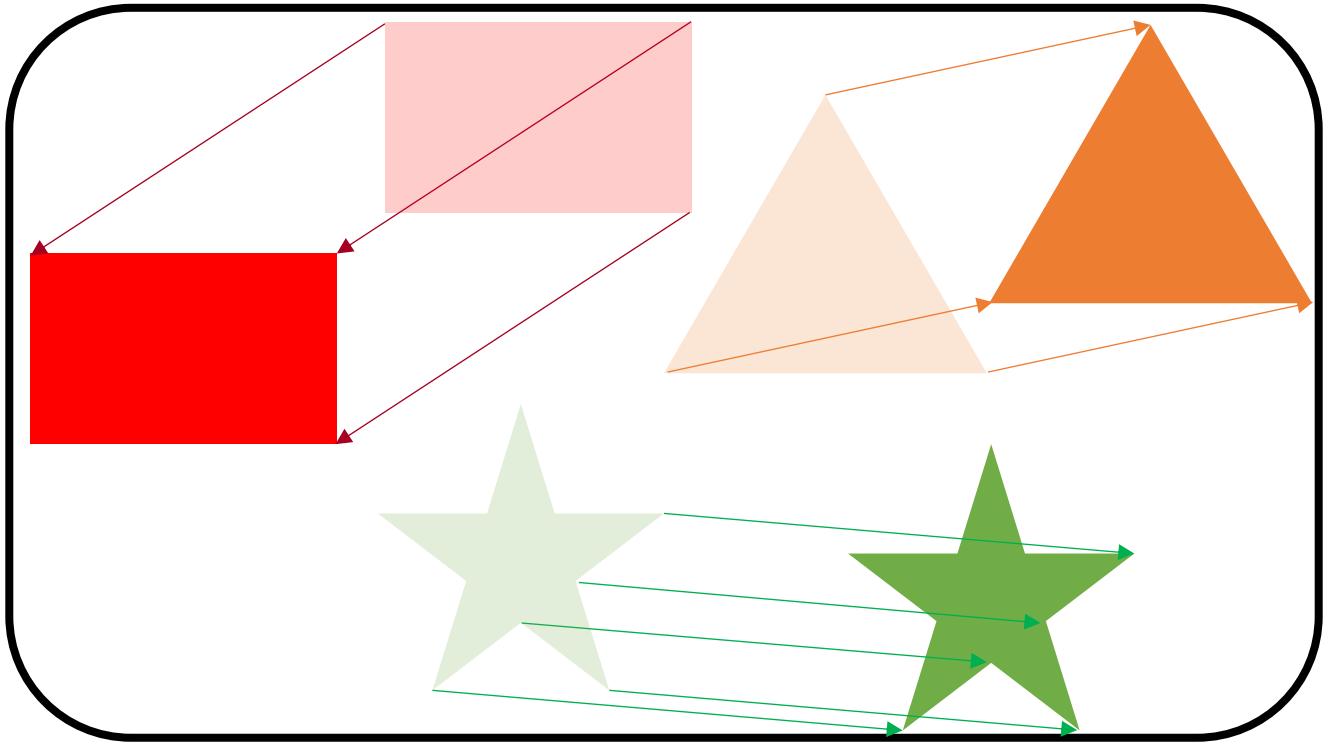


Image Plane

Thanks Chahat for these slides!

# Moving Objects/Scene/Camera



# Moving Objects/Scene/Camera

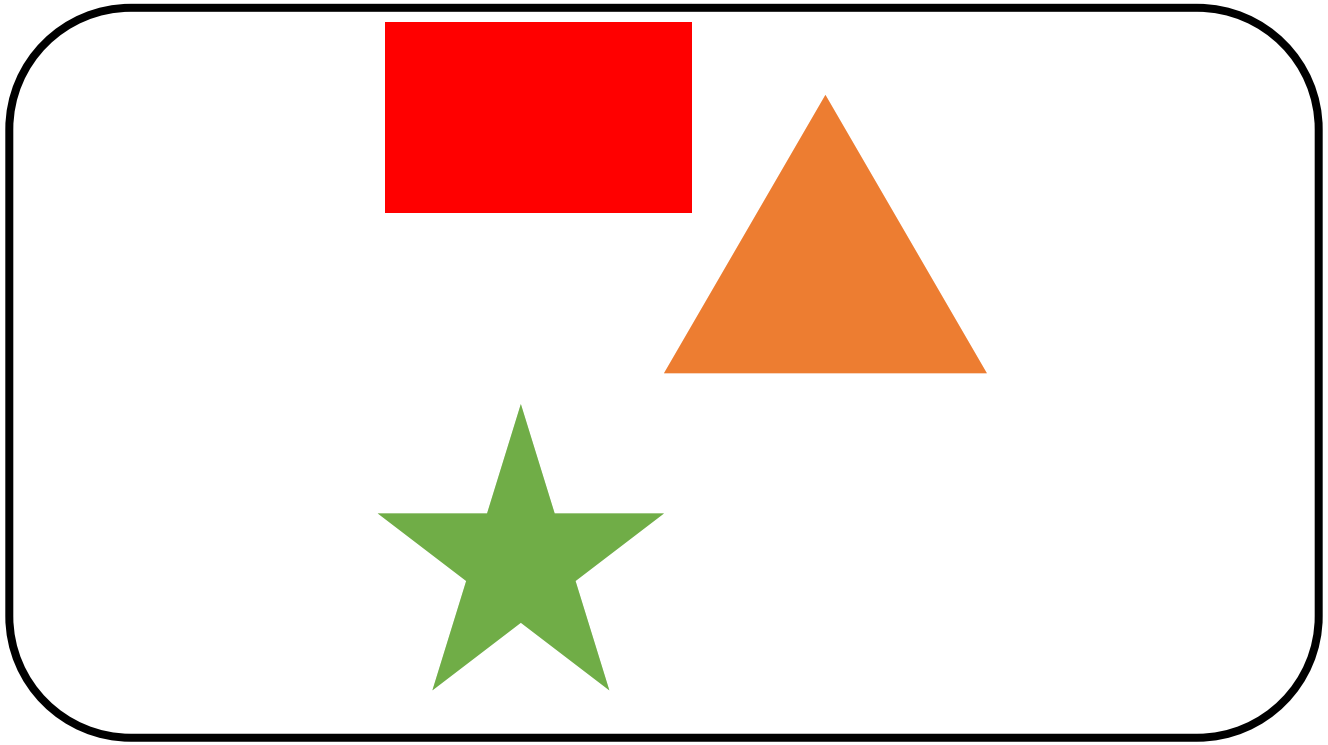
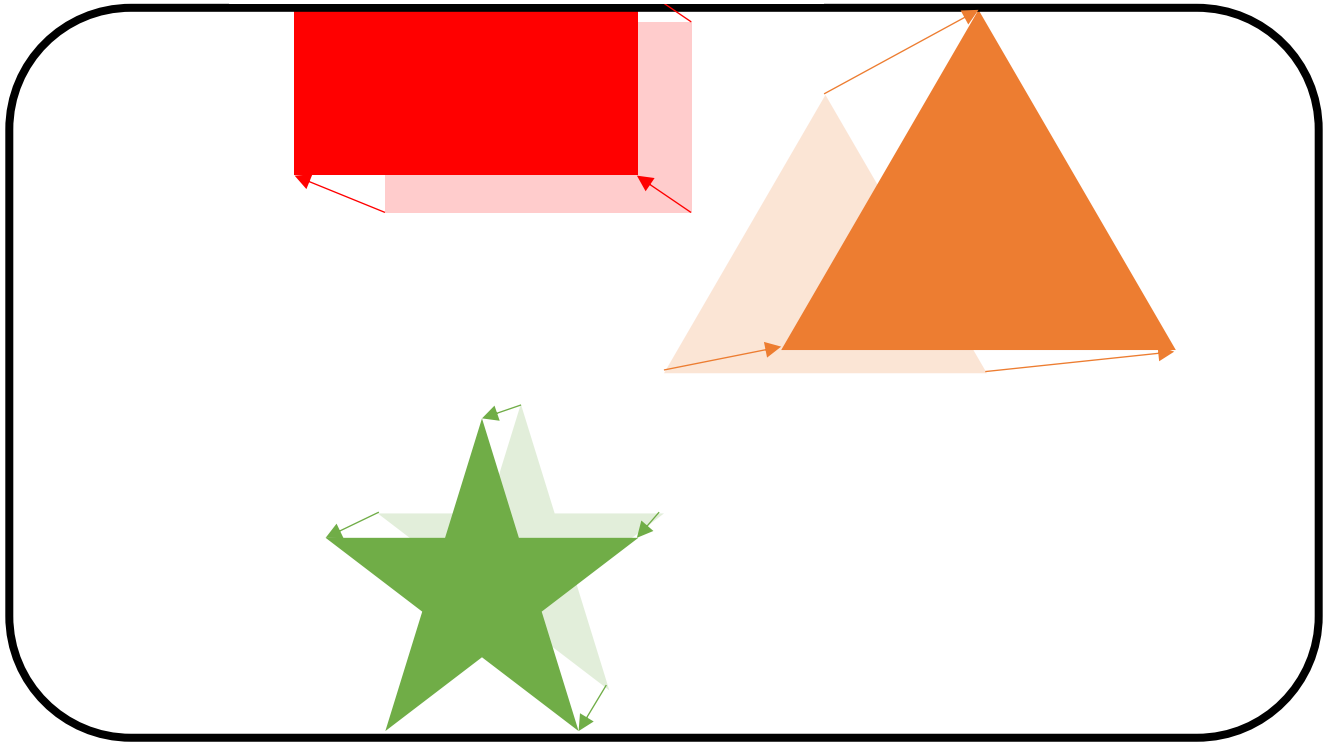
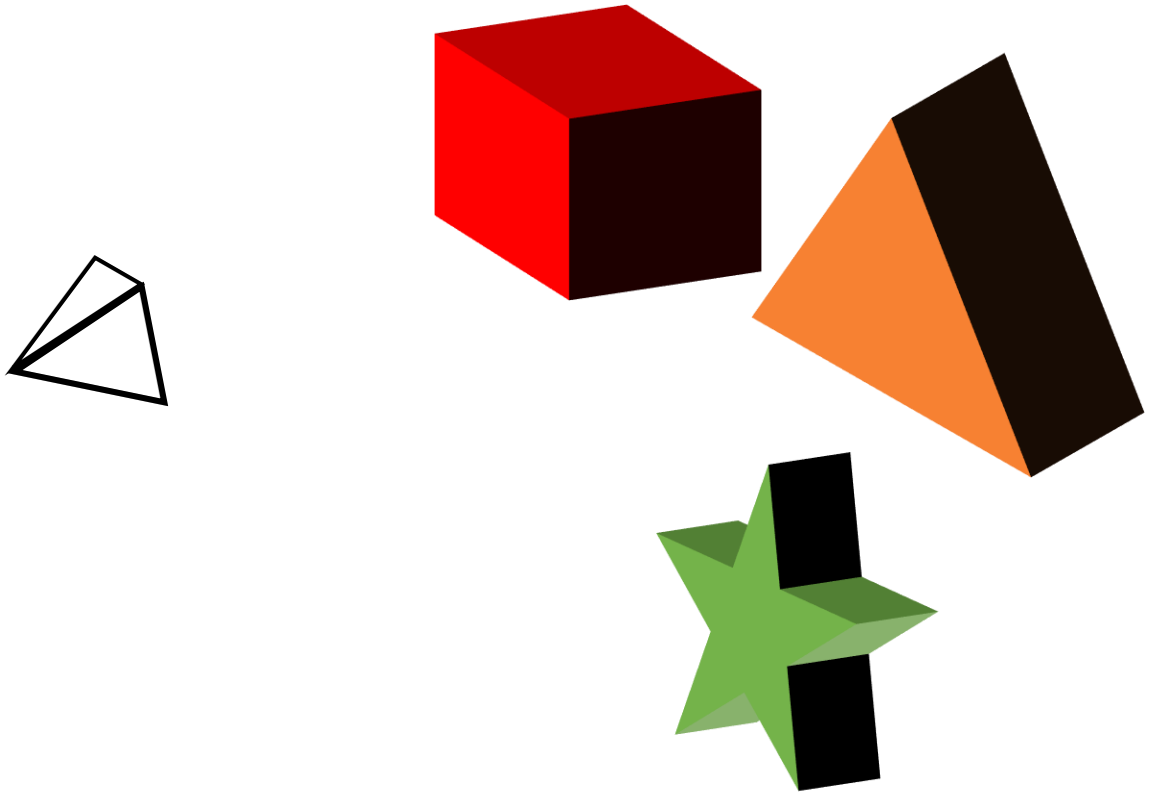


Image Plane

# Moving Objects/Scene/Camera



# Let's Look At It In 3D



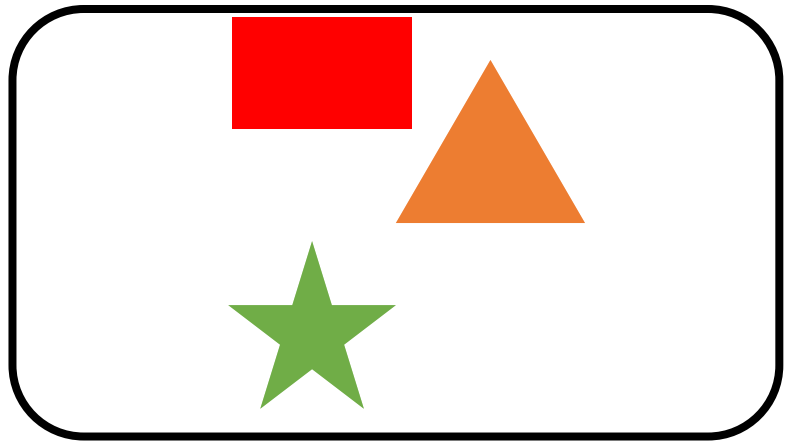
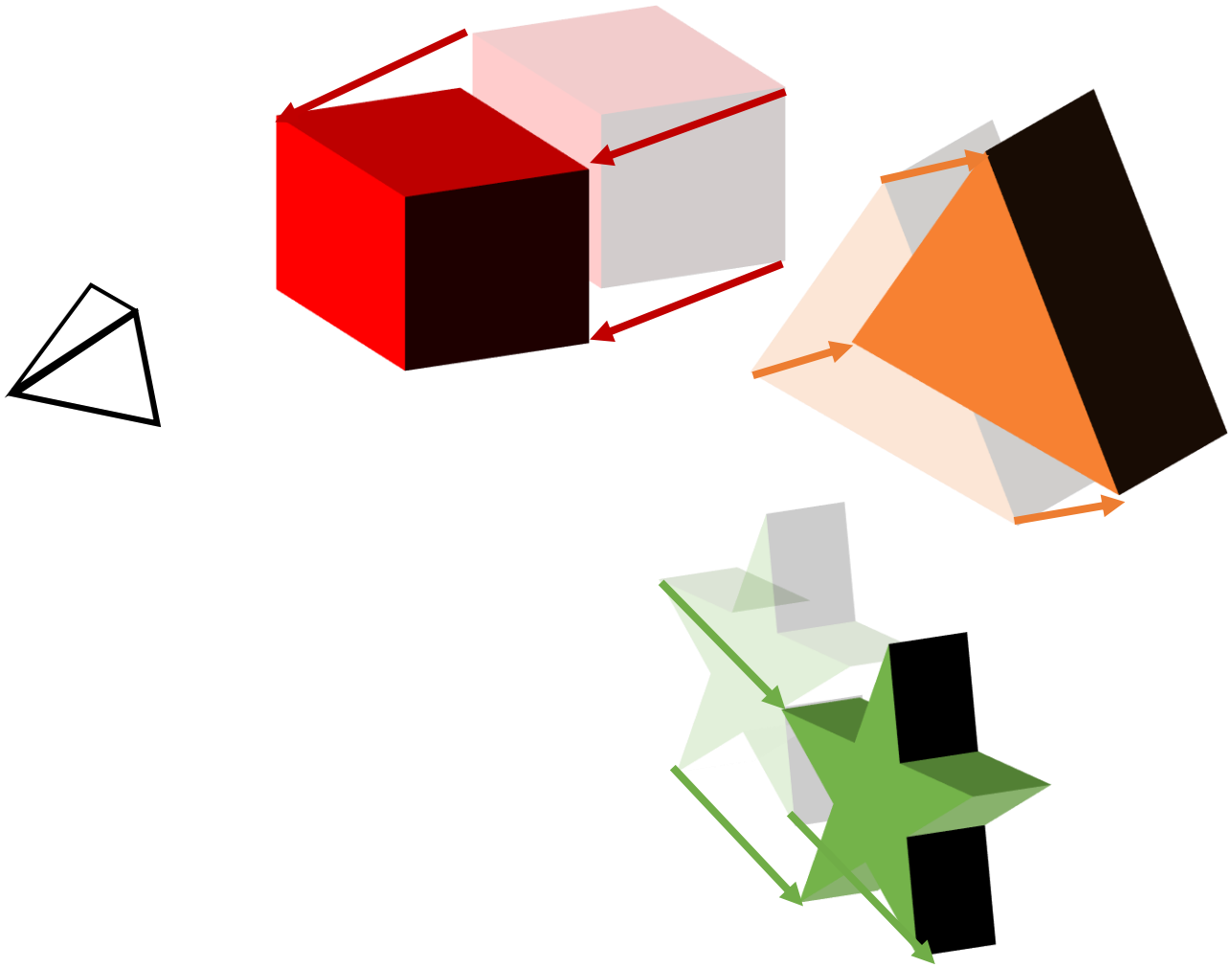


Image Plane

# Let's Look At It In 3D



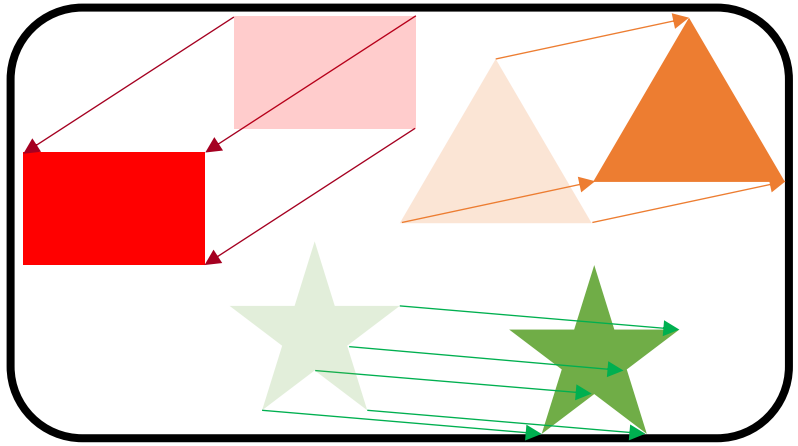
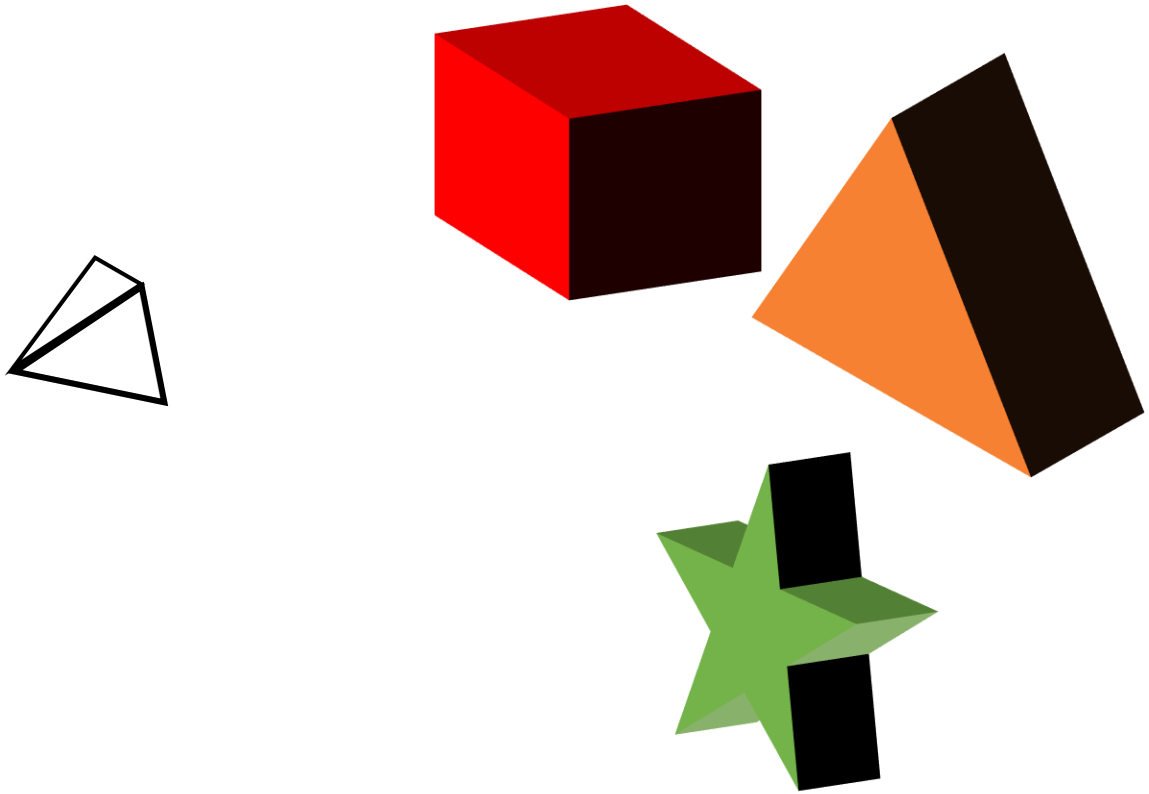


Image Plane

**Objects Could be Moving**

# Let's Look At It In 3D



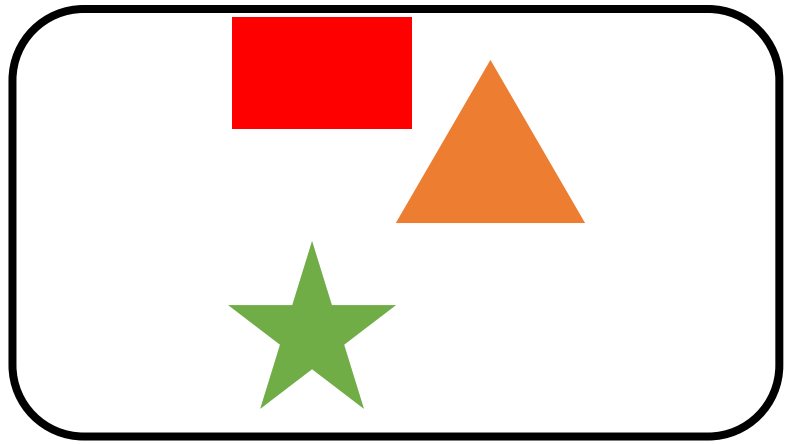
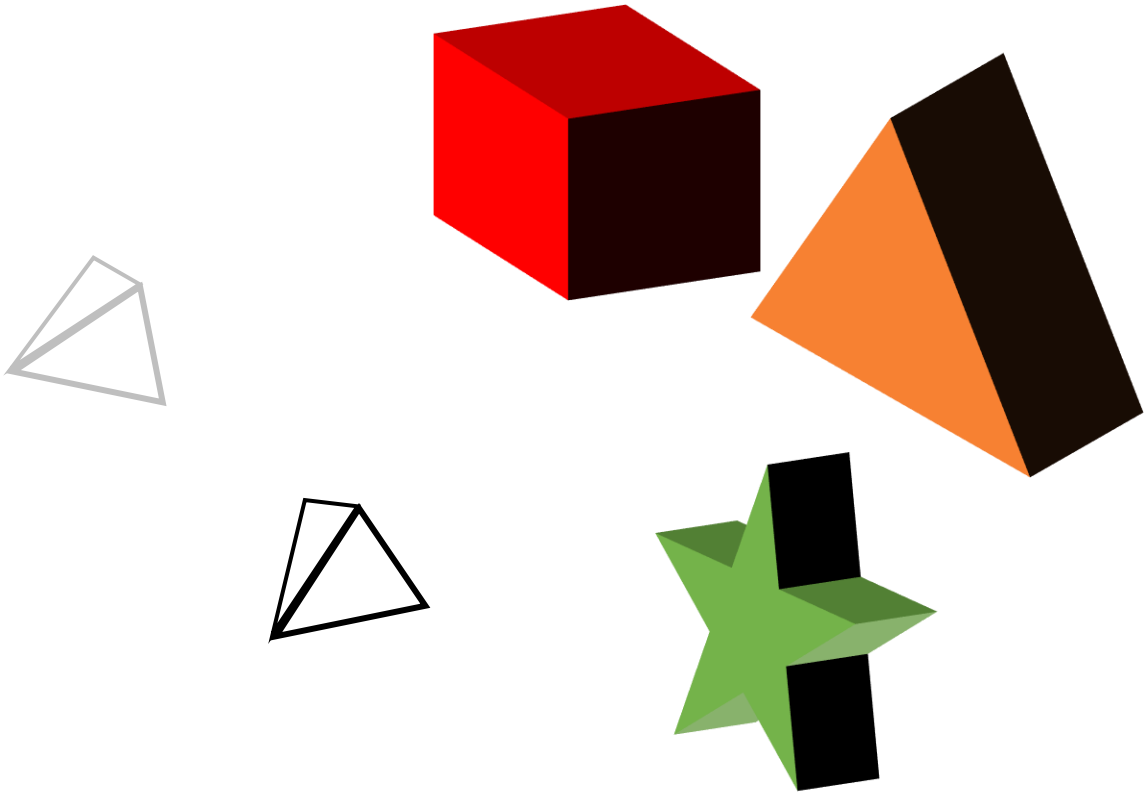


Image Plane

# Let's Look At It In 3D



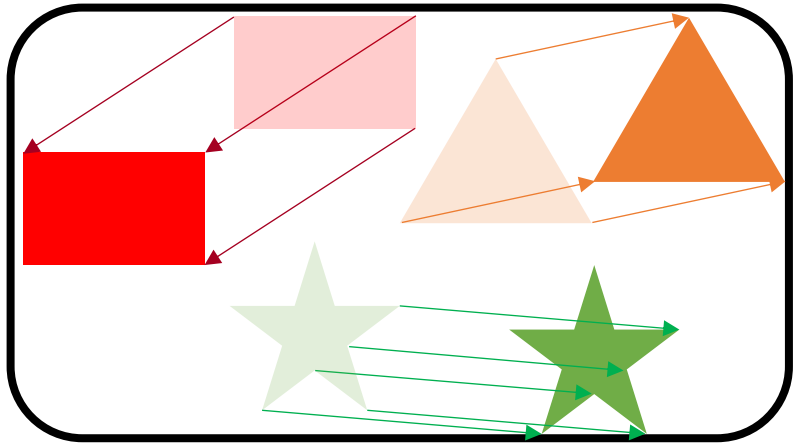
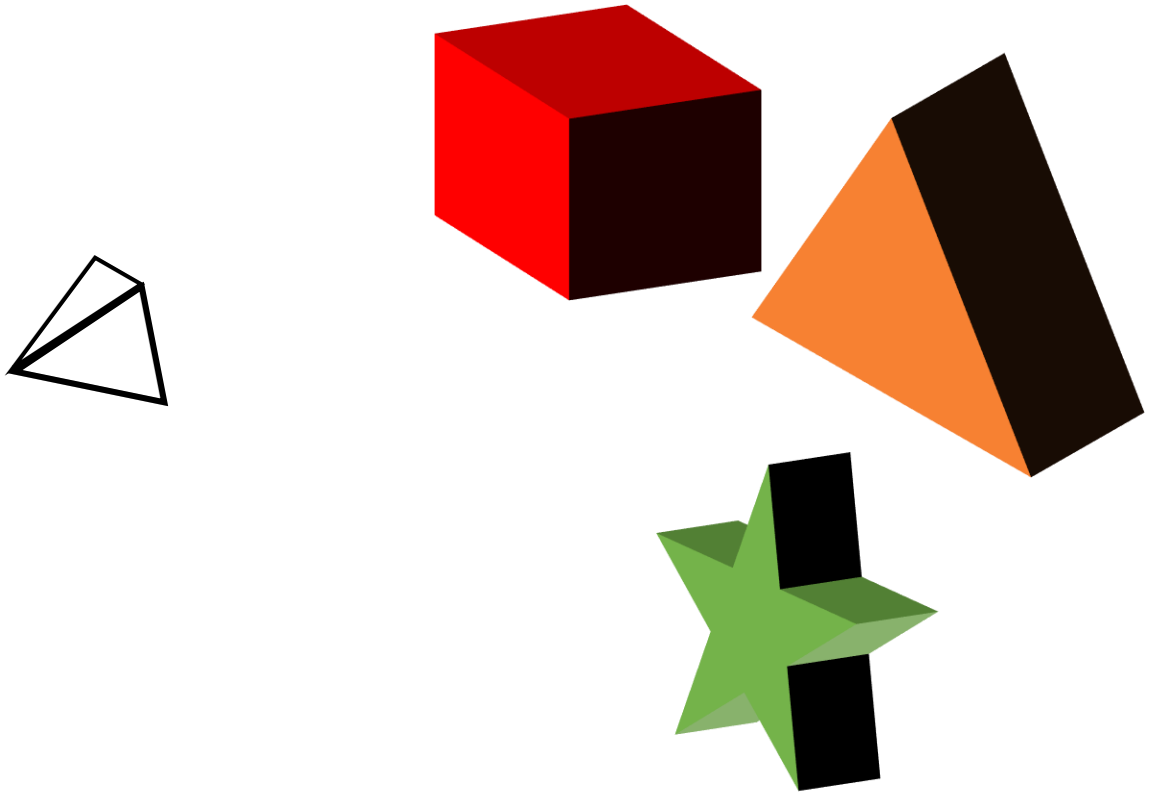


Image Plane

**Camera Could be Moving**

# Let's Look At It In 3D



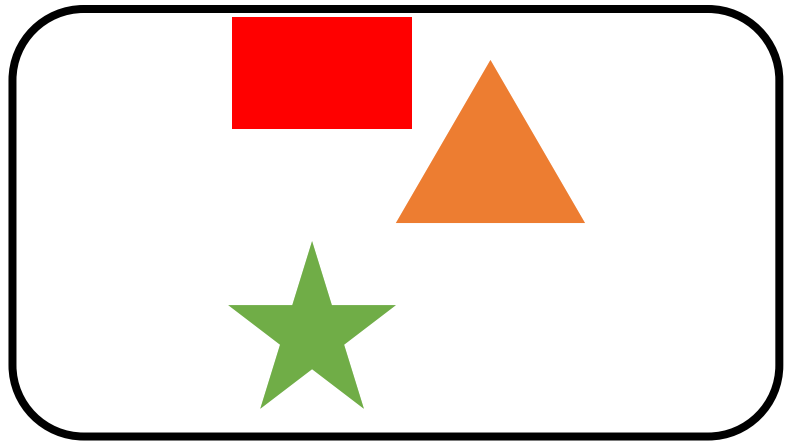
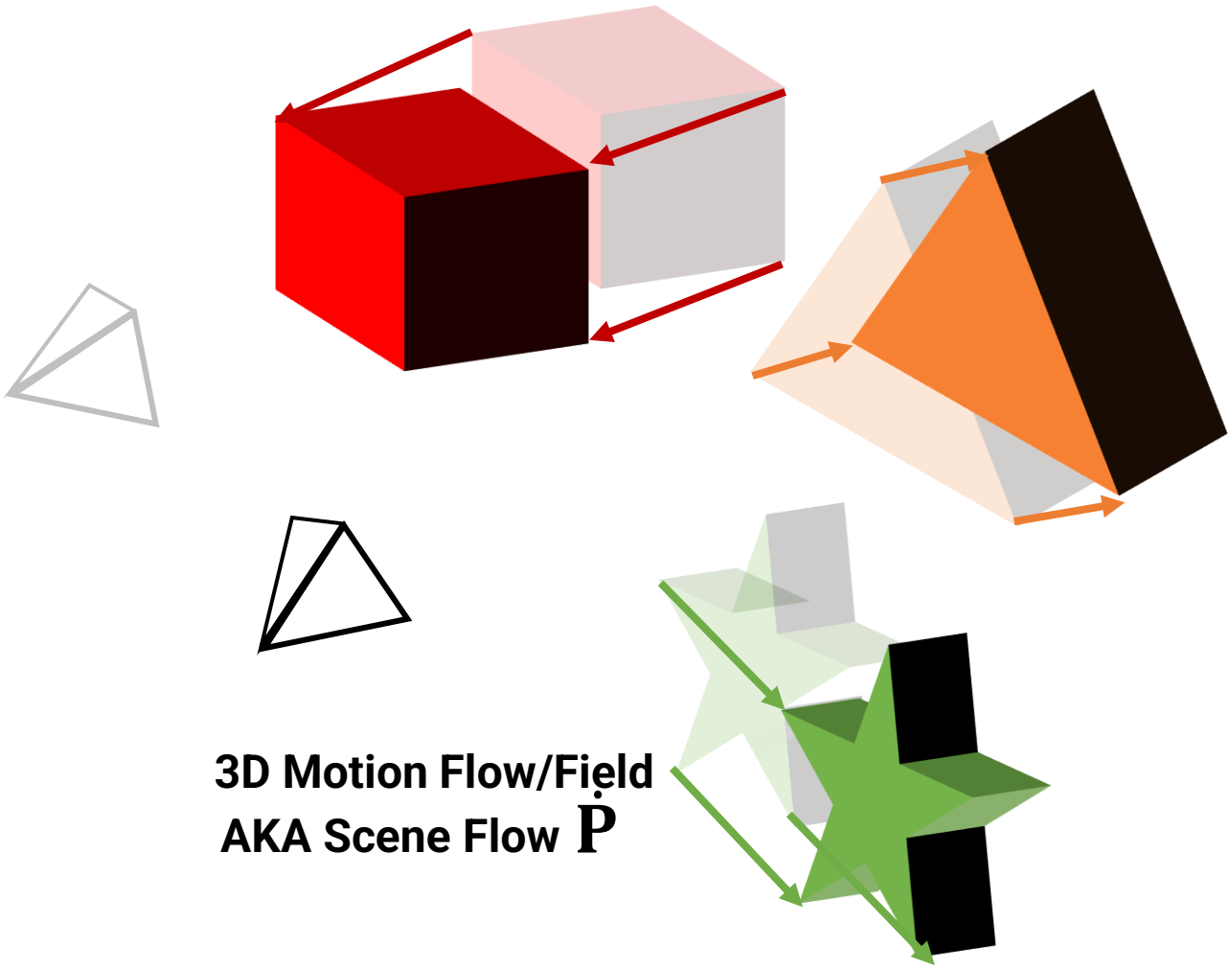


Image Plane

# Let's Look At It In 3D



**3D Motion Flow/Field**
**AKA Scene Flow $\dot{\mathbf{P}}$**

**Optical Flow $\dot{\mathbf{p}}$**

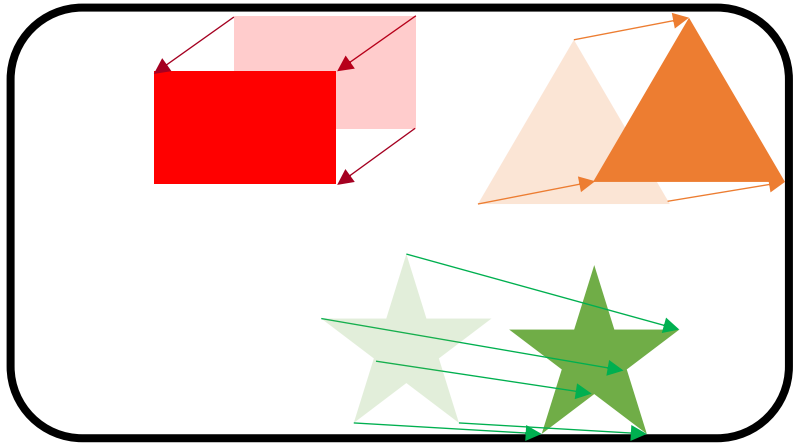


Image Plane

**Both Objects and Camera Could be Moving**

Think of imaging the 3D Motion Flow!
This is called **Optical Flow**!

# Types Of Optical Flow


Sparse

For some pixels

- Can be every $N^{th}$ pixel (Block matching)
- Can be at keypoints (Like SIFT)

# Types Of Optical Flow


Sparse

For some pixels

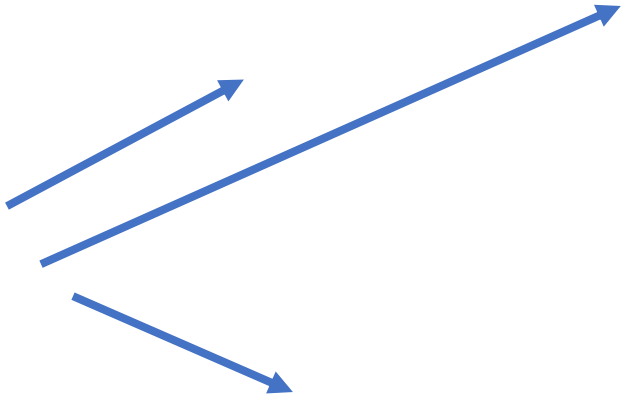
- Can be every $N^{th}$ pixel (Block matching)
- Can be at keypoints (Like SIFT)

Dense

For every pixel

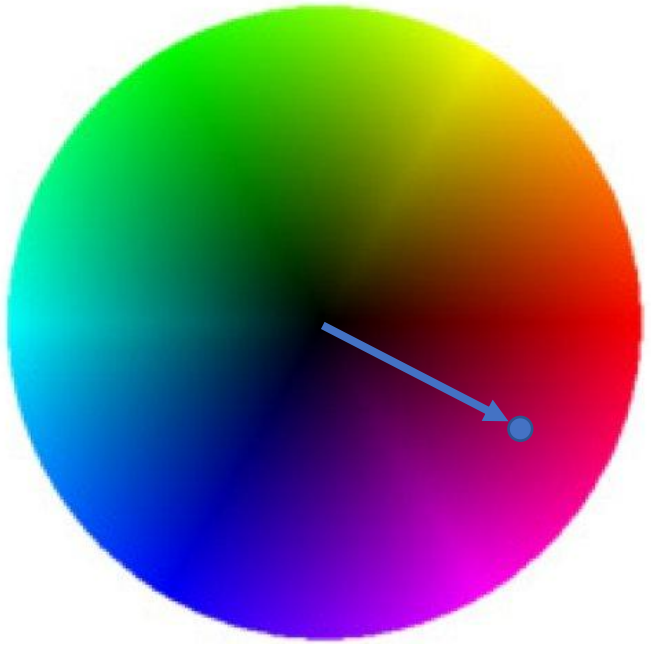
- Generally Block matching

# How Do You Represent/Visualize $\dot{p}$?



Quiver

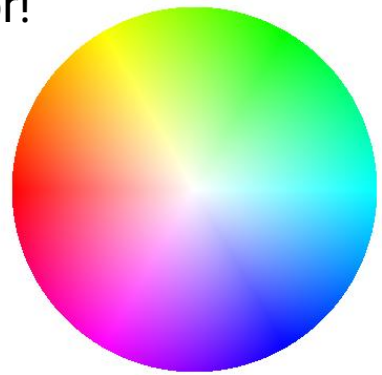
Similar flow has similar arrows!



This is the value mapped to a color!

Colormap

Similar flow has similar color!



Center can be white too!

# How Do You Represent/Visualize $\dot{p}$?

# Let’s Attack Flow

# Let’s Attack Flow

# Let’s Attack Flow

# Let’s Attack Flow

# Attacking Flow

Let $F(I, I')$ denote an optical flow network and $\mathcal{I}$ a dataset of frame pairs.

Let $A(I, p, t, l)$ denote image $I$ with patch $p$ transformed by $t$ inserted at location $l$.

Let $\mathcal{T}$ denote a distribution over affine 2D transformations.

Let $\mathcal{L}$ denote a (uniform) distribution over the image domain.

**Goal:** Find a patch $\hat{p}$ such that

$$\hat{p} = \underset{p}{\operatorname{argmin}}\, \mathbb{E}_{(I,I')\sim\mathcal{I},\, t\sim\mathcal{T},\, l\sim\mathcal{L}} \left[ \frac{(u, v) \cdot (\tilde{u}, \tilde{v})}{\|(u, v)\|_2 \cdot \|(\tilde{u}, \tilde{v})\|_2} \right]$$

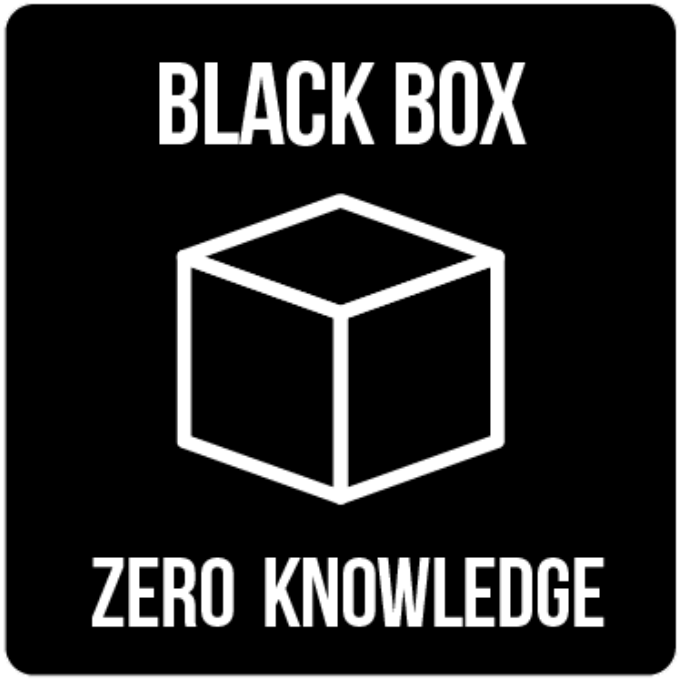
with $$(u, v) = F(I, I')$$

$$(\tilde{u}, \tilde{v}) = F(A(I, p, t, l), A(I', p, t, l))$$

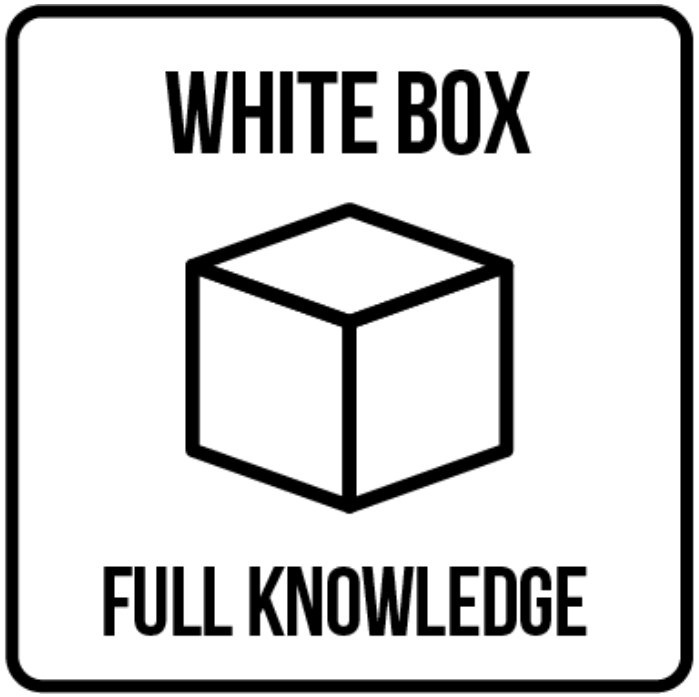
**Intuition:** Find patch $\hat{p}$ that reverses the direction of the optical flow.

Ranjan, Anurag, et al. "Attacking optical flow." Proceedings of the IEEE/CVF international conference on computer vision. 2019.

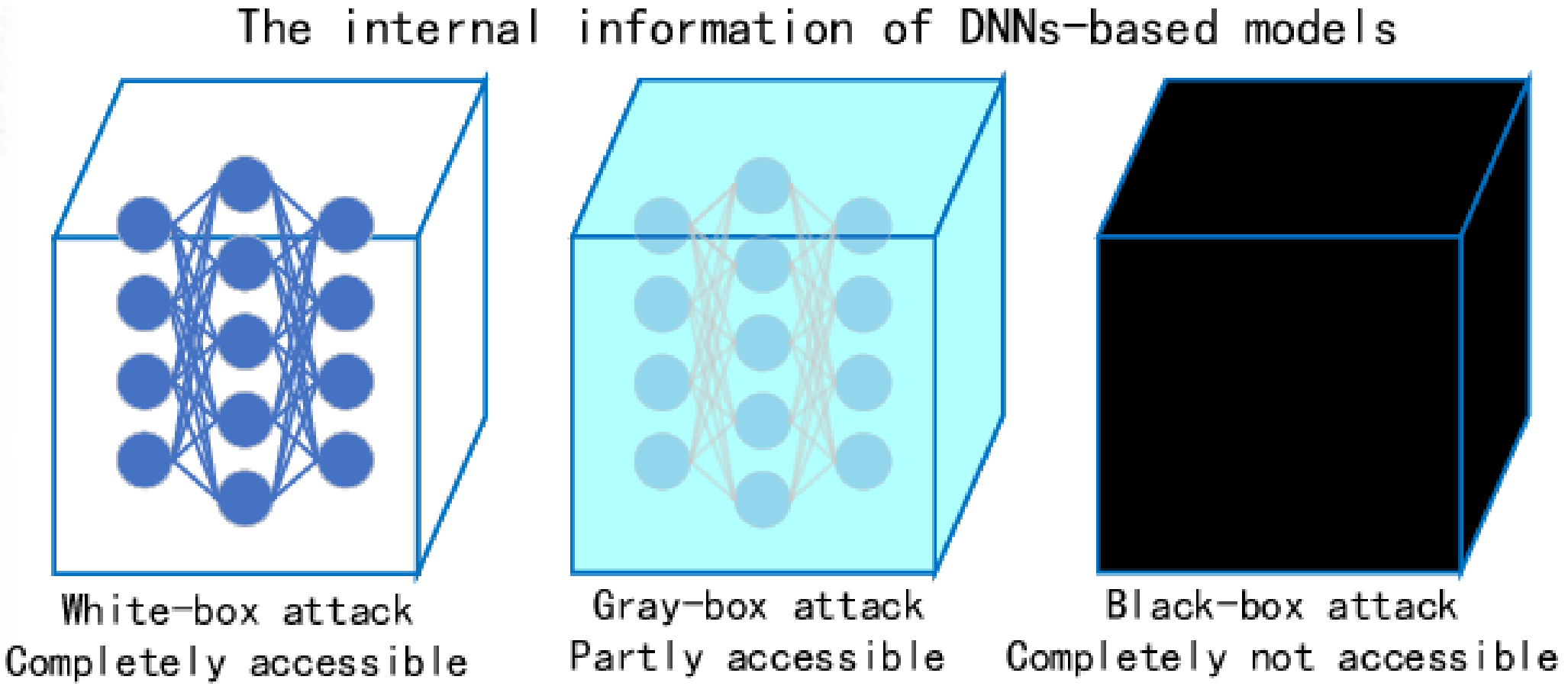
# Black Vs White Box Attack





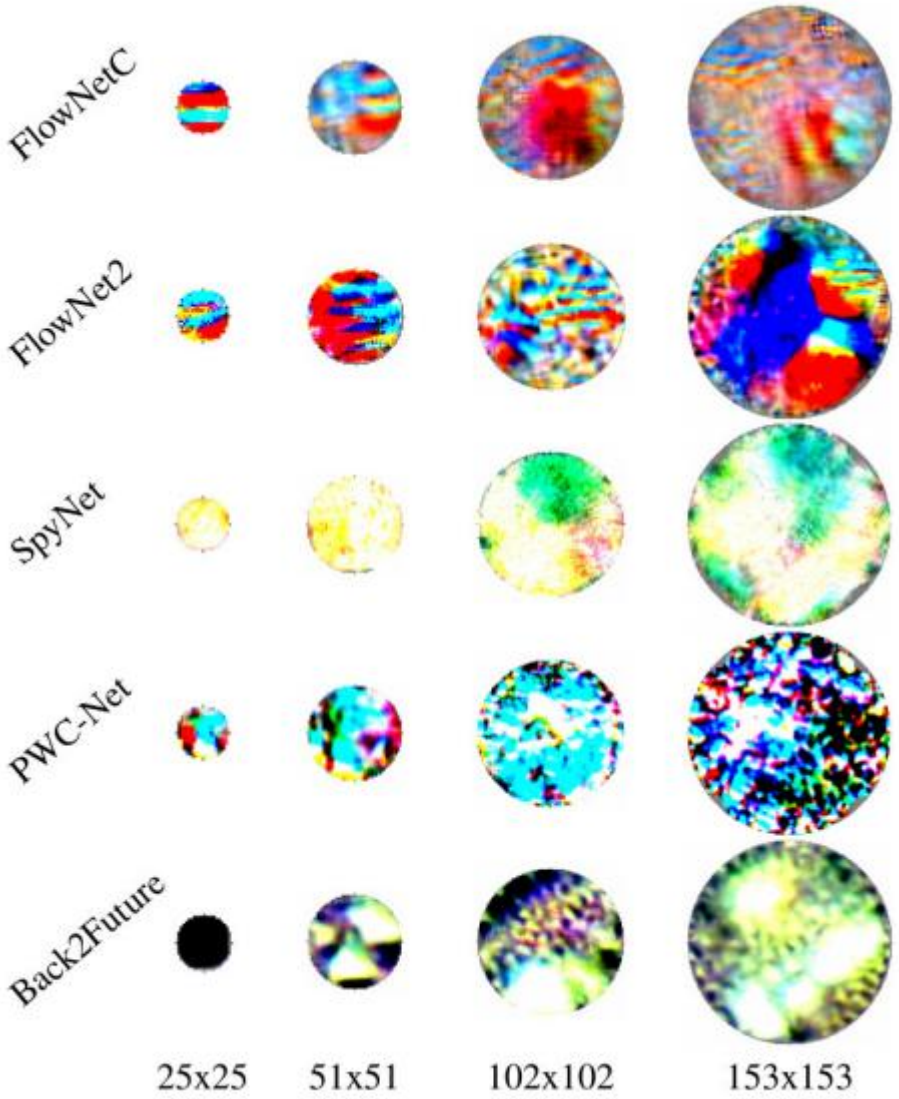


# Black Vs White Box Attack

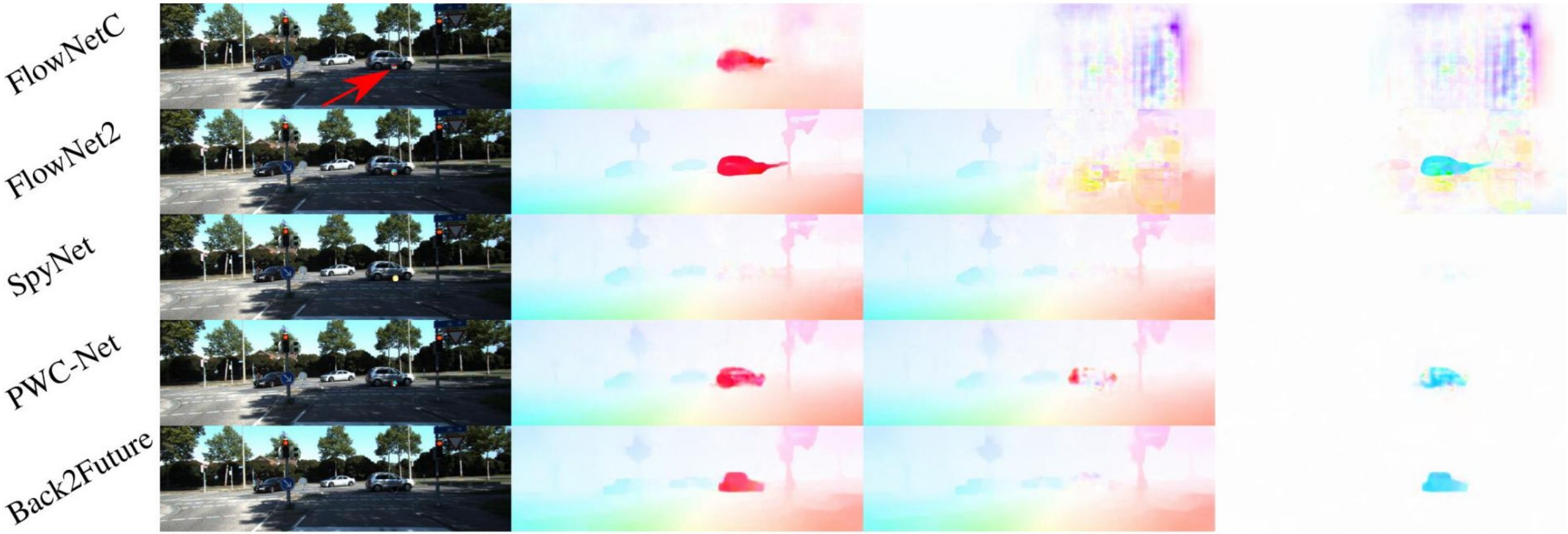


# White Box Attacks

- Attack each network in separation
- Location sampled uniformly within image
- Scale: ±5%
- Rotation: ±10%
- Optimize on 32K unlabeled KITTI frames
  - Flow predictions as pseudo ground truth
- Learn patches of 4 different sizes

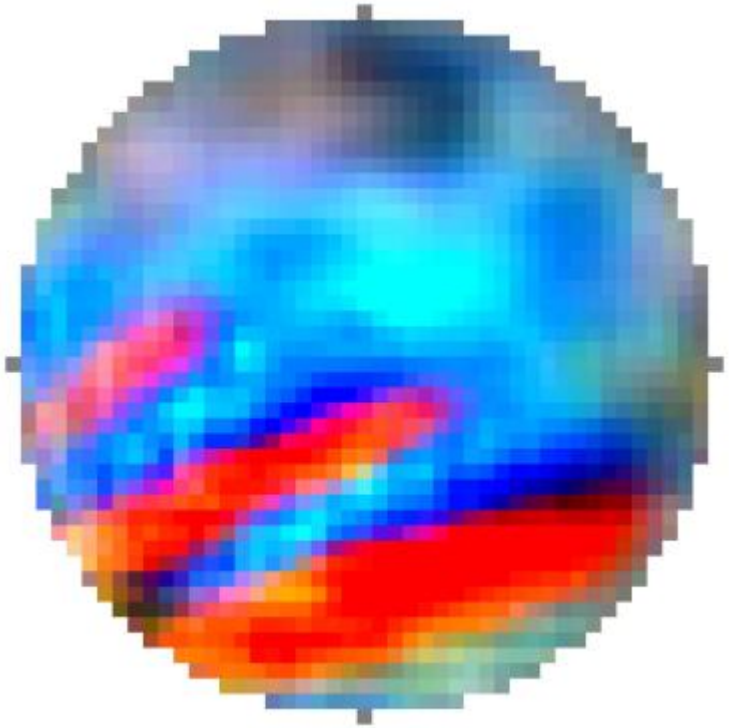


# White Box Attacks

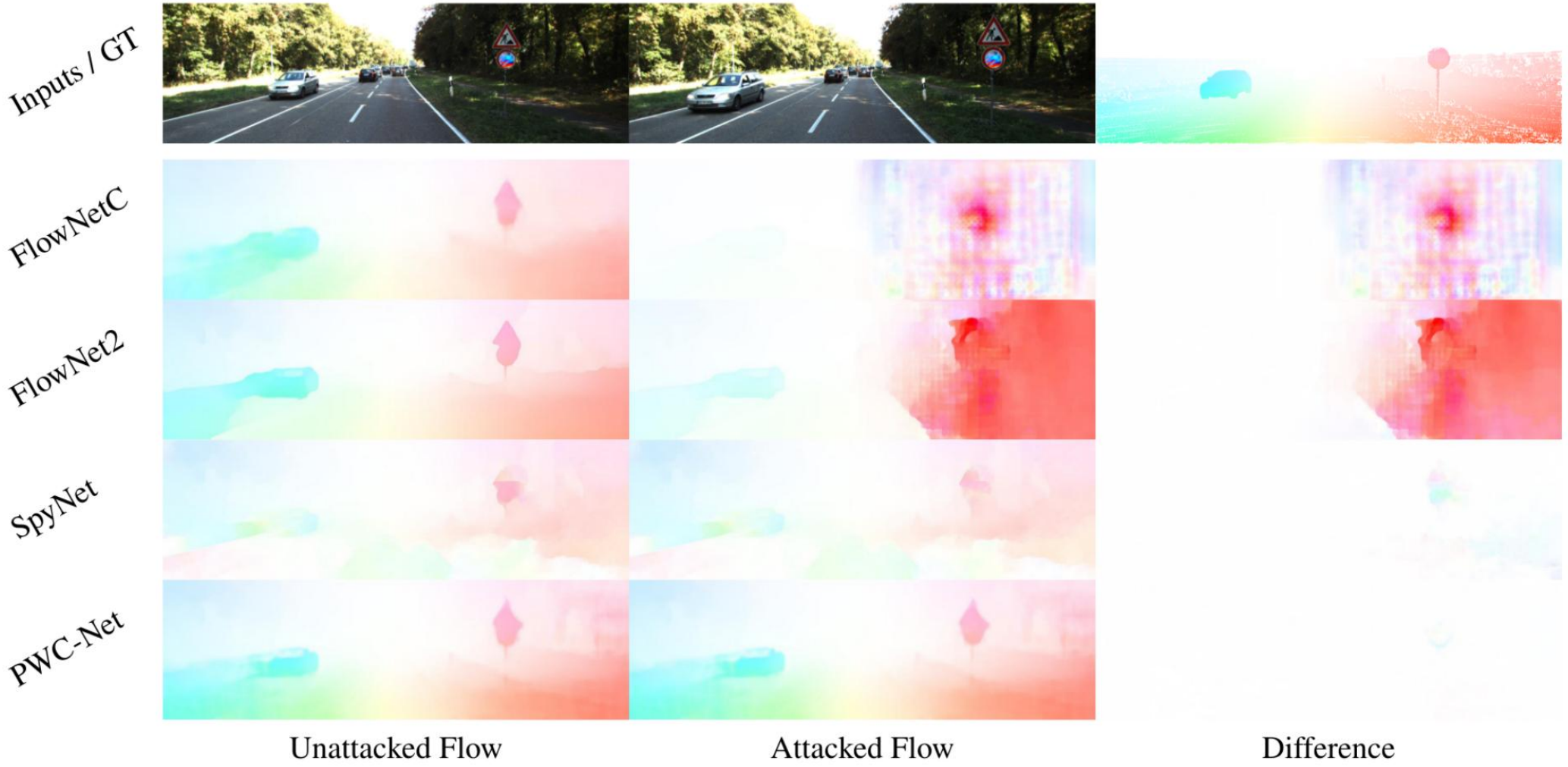


# Black Box Attacks

- Patch is optimized over several networks (e.g., FlowNet2 and PWCNet)
- Patch is used to attack all networks
- Patch is moved as if it was part of the scene



# Black Box Attacks

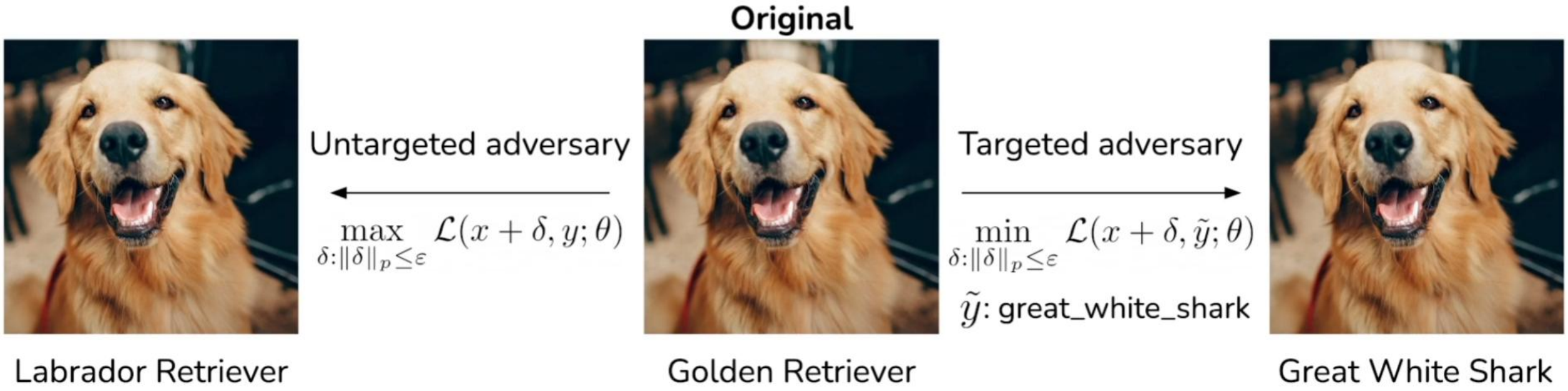


# Attacks in the real world

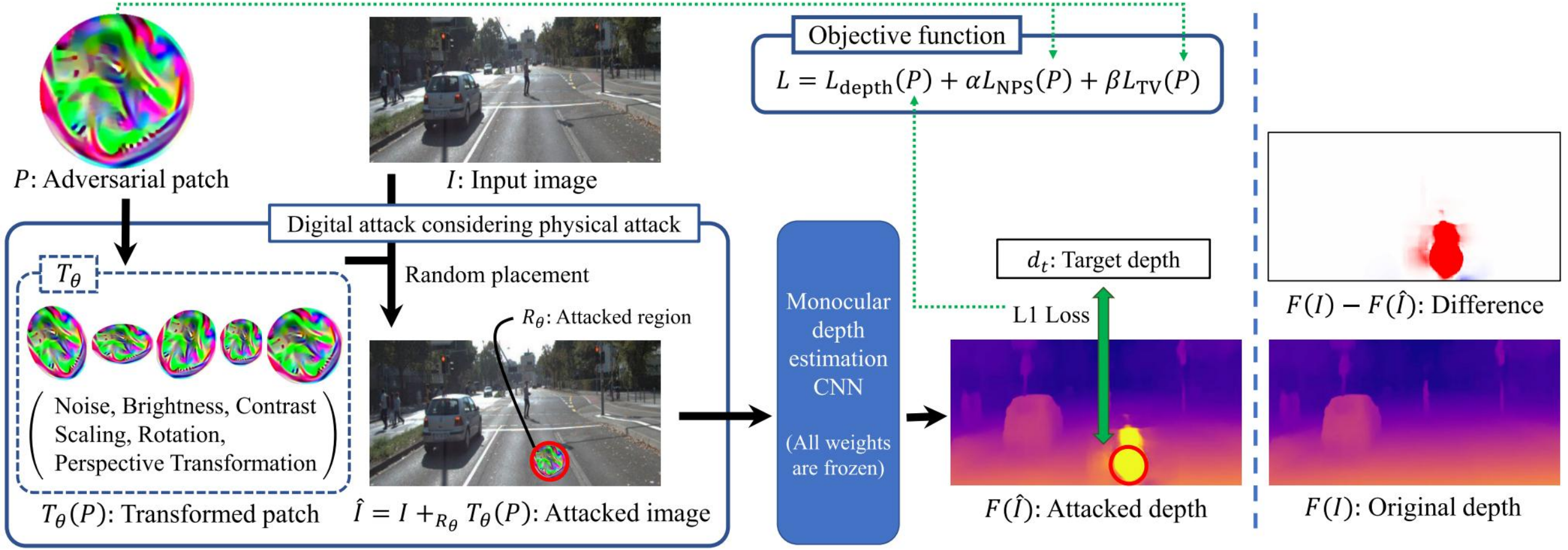
## Simple Translation

# Untargeted Vs Targeted Attacks

Untargeted: Maximize loss
Targeted: Misclassify as a particular thing!



# Attacking Monocular Depth




Yamanaka, K., Matsumoto, R., Takahashi, K., & Fujii, T. (2020). Adversarial patch attacks on monocular depth estimation networks. IEEE Access, 8, 179094-179104.

# All Hail The Losses!

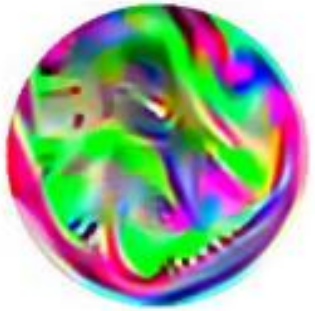
$$L = L_{\text{depth}}(P) + \alpha L_{\text{NPS}}(P) + \beta L_{\text{TV}}(P)$$

$$L_{\text{depth}}(P) = \sum_{(i,j)\in R_\theta} \left| d_t - F_{i,j}(I +_{R_\theta} T_\theta(P)) \right|$$

$$L_{\text{NPS}}(P) = \sum_{i,j} \min_{\vec{c}\in C} \left\| \vec{p}_{i,j} - \vec{c} \right\|_1$$

$$L_{\text{TV}}(P) = \sum_{i,j} \sqrt{(p_{i,j} - p_{i+1,j})^2 + (p_{i,j} - p_{i,j+1})^2}$$

# White Box Attack Patches

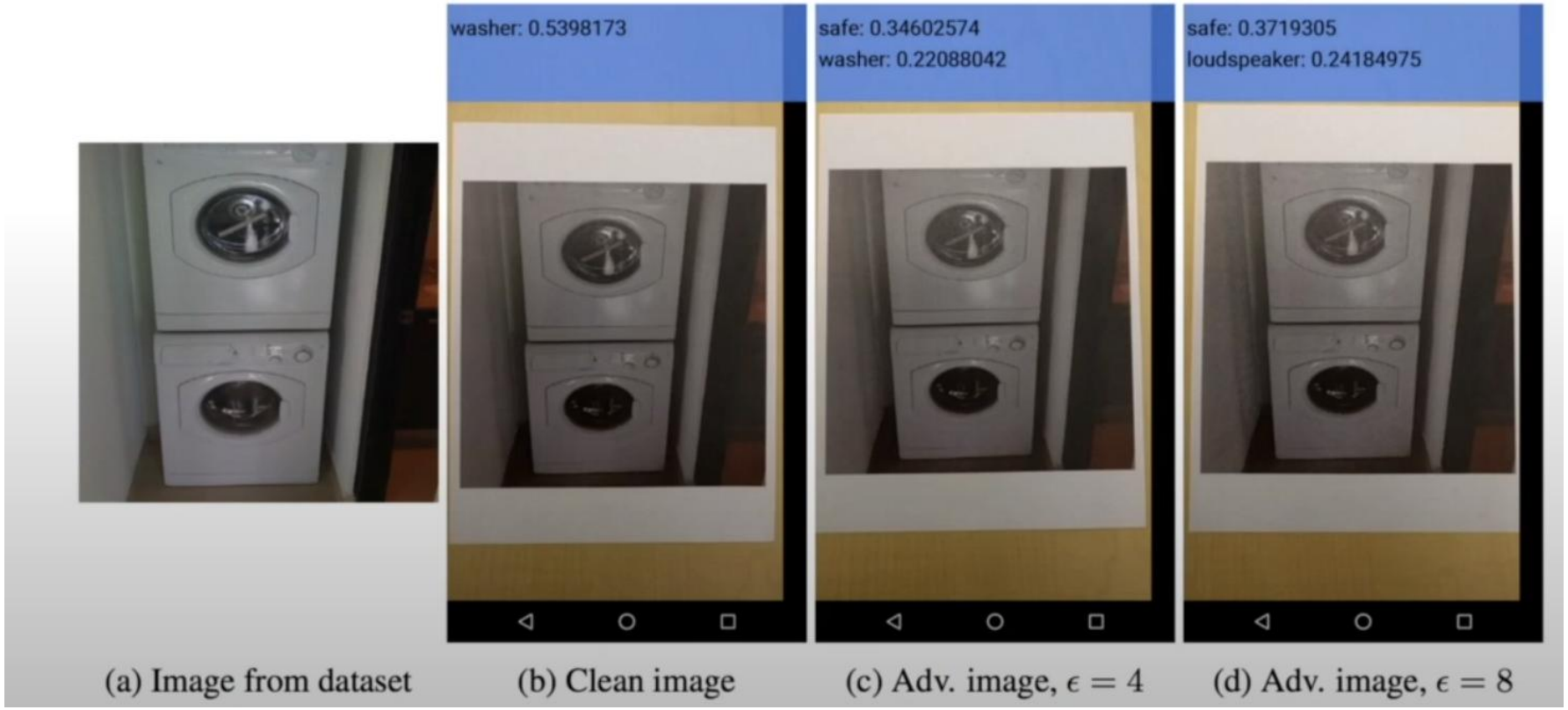
| Target depth | Target methods | | | | | |
|---|---|---|---|---|---|---|
| | Guo et al. [7] | | Lee et al. [8] | | Guo et al. [7], Lee et al. [8] | |
| $d_t = 3$ [m] |  | $P_n^1$ |  | $P_n^2$ |  | $P_n^*$ |
| $d_t = 150$ [m] |  | $P_f^1$ |  | $P_f^2$ |  | $P_f^*$ |

# Experiment in the real scene

(Attack on Guo et al.’s method)

# Transferability

- Craft an attack on similar models and attacks can transfer!
- Black box attack methods are generally better transferred



# What Happened Here?

Teapot

# What Happened Here?

Joystick

Su, Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks." IEEE Transactions on Evolutionary Computation 23.5 (2019): 828-841.

# What Happened Here?

## Single Pixel Attack

$$\underset{e(\mathbf{x})^*}{\text{maximize}} \quad f_{\text{adv}}(\mathbf{x} + e(\mathbf{x}))$$
$$\text{subject to} \quad \|e(\mathbf{x})\|_0 \leq d$$

Joystick 

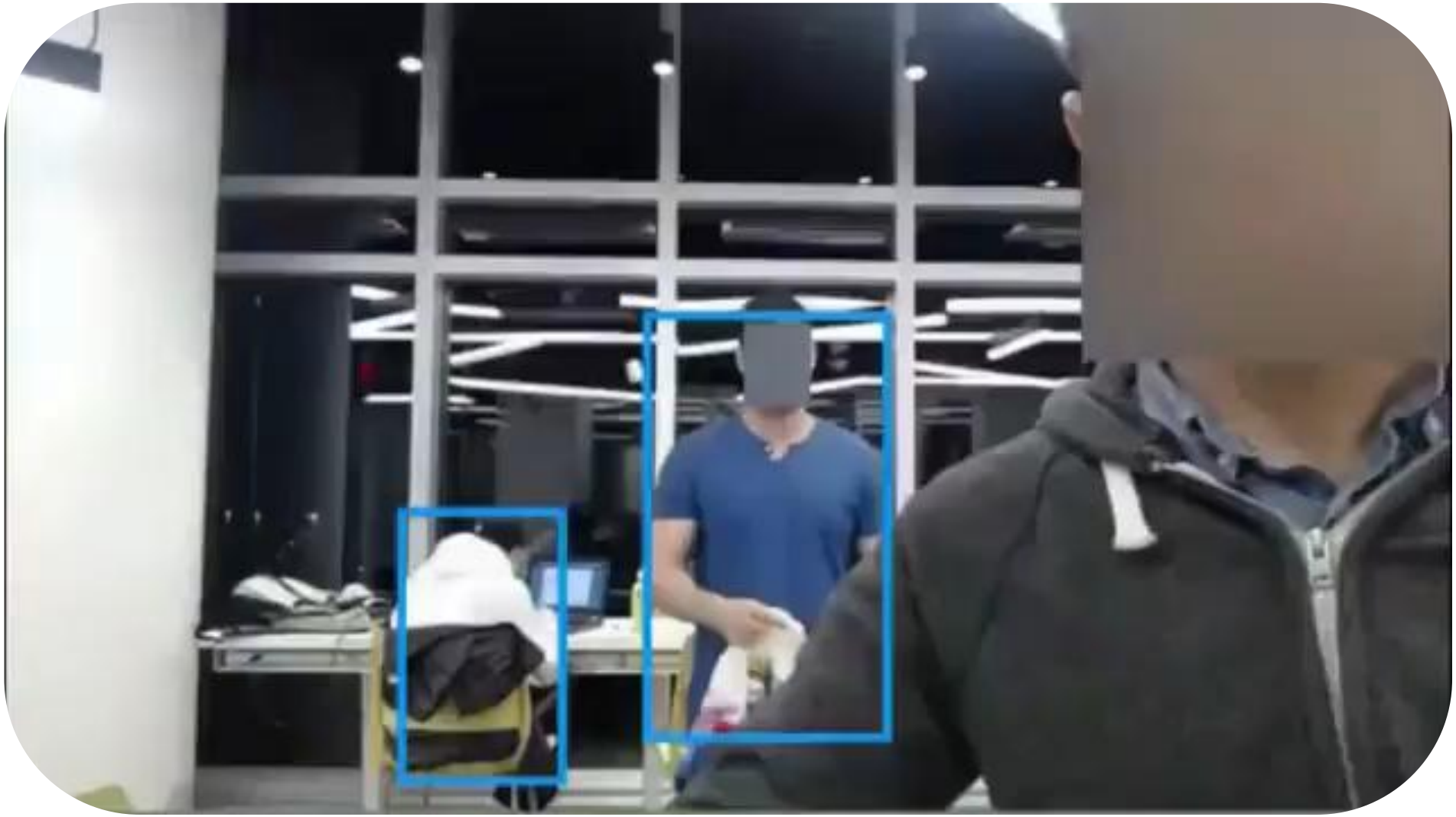
Su, Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks." IEEE Transactions on Evolutionary Computation 23.5 (2019): 828-841.

# Does It Matter?

"Hot pixels"

You can, well almost!

# I Want To Be Invisible!



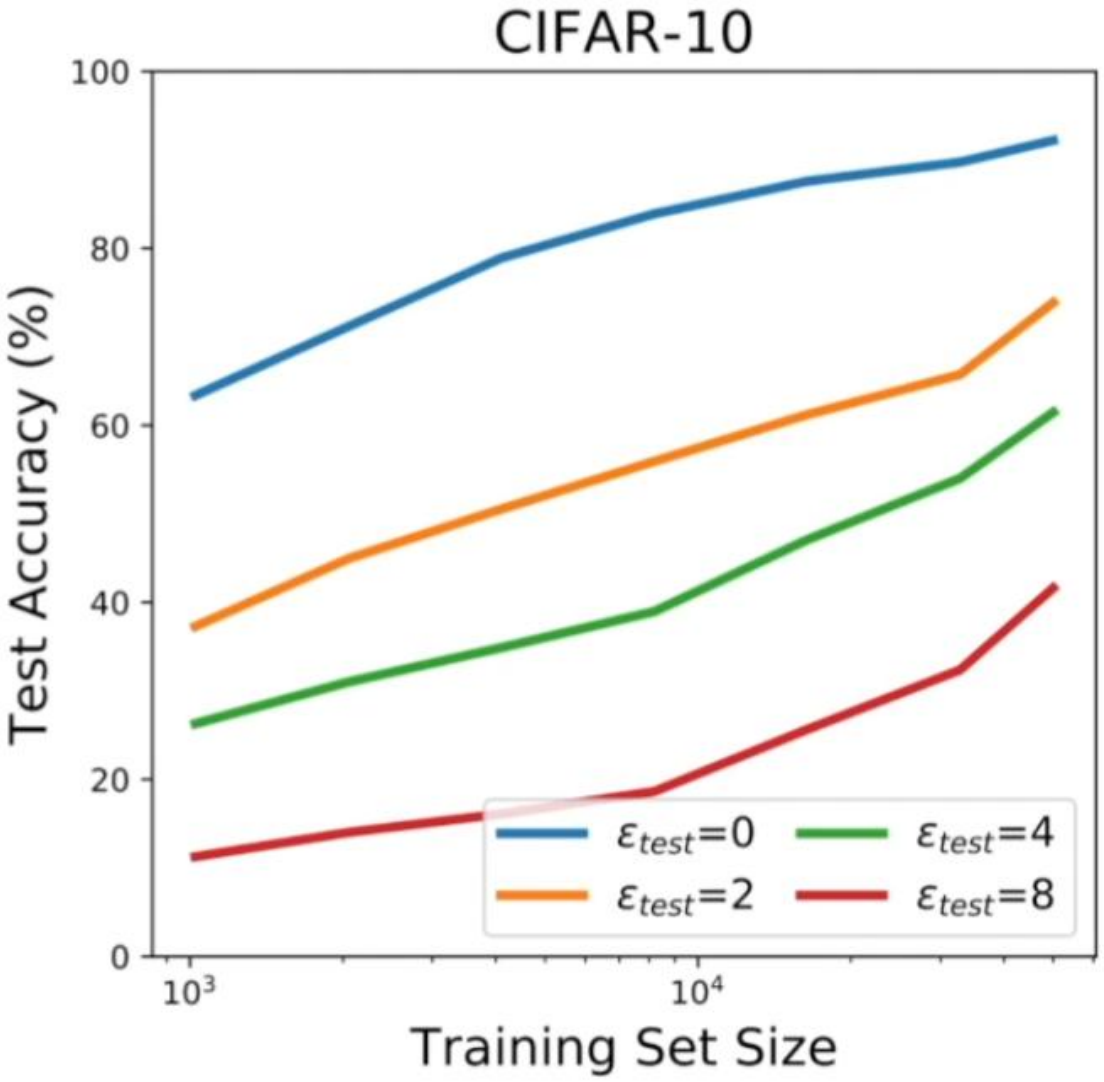
# Improving Robustness

More data = better!

Adversarially pre-train on larger similar datasets and fine-tune on yours!

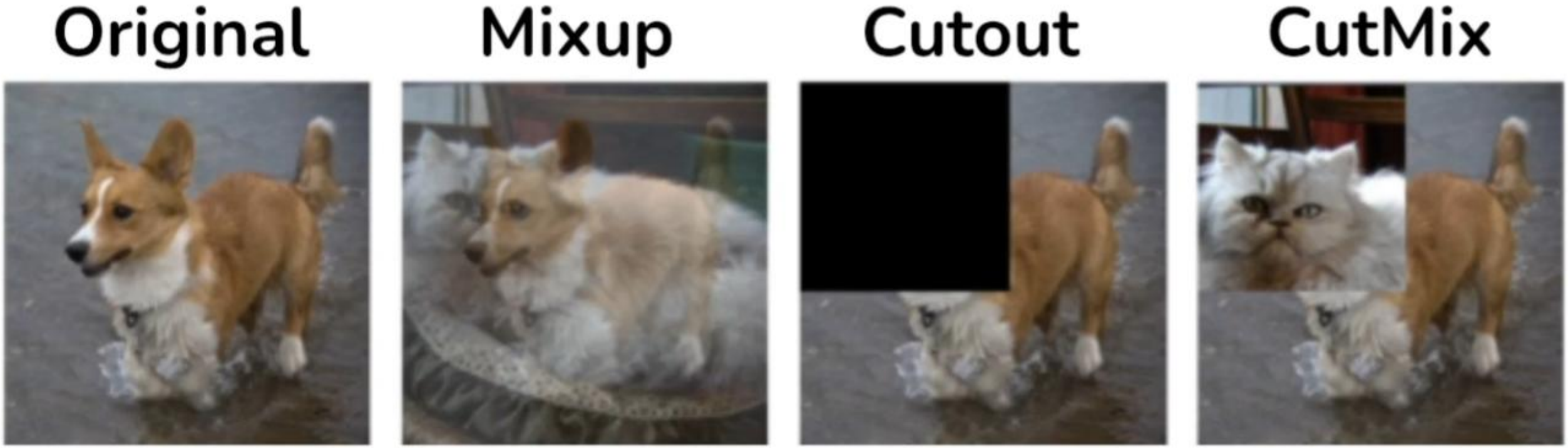
Adversarial robustness transfers to similar tasks ☺

| | CIFAR-10 | | CIFAR-100 | |
|---|---|---|---|---|
| | Clean | Adversarial | Clean | Adversarial |
| Normal Training | 96.0 | 0.0 | 81.0 | 0.0 |
| Adversarial Training | 87.3 | 45.8 | 59.1 | 24.3 |
| Adv. Pre-Training and Tuning | 87.1 | 57.4 | 59.2 | 33.5 |



# Improving Robustness

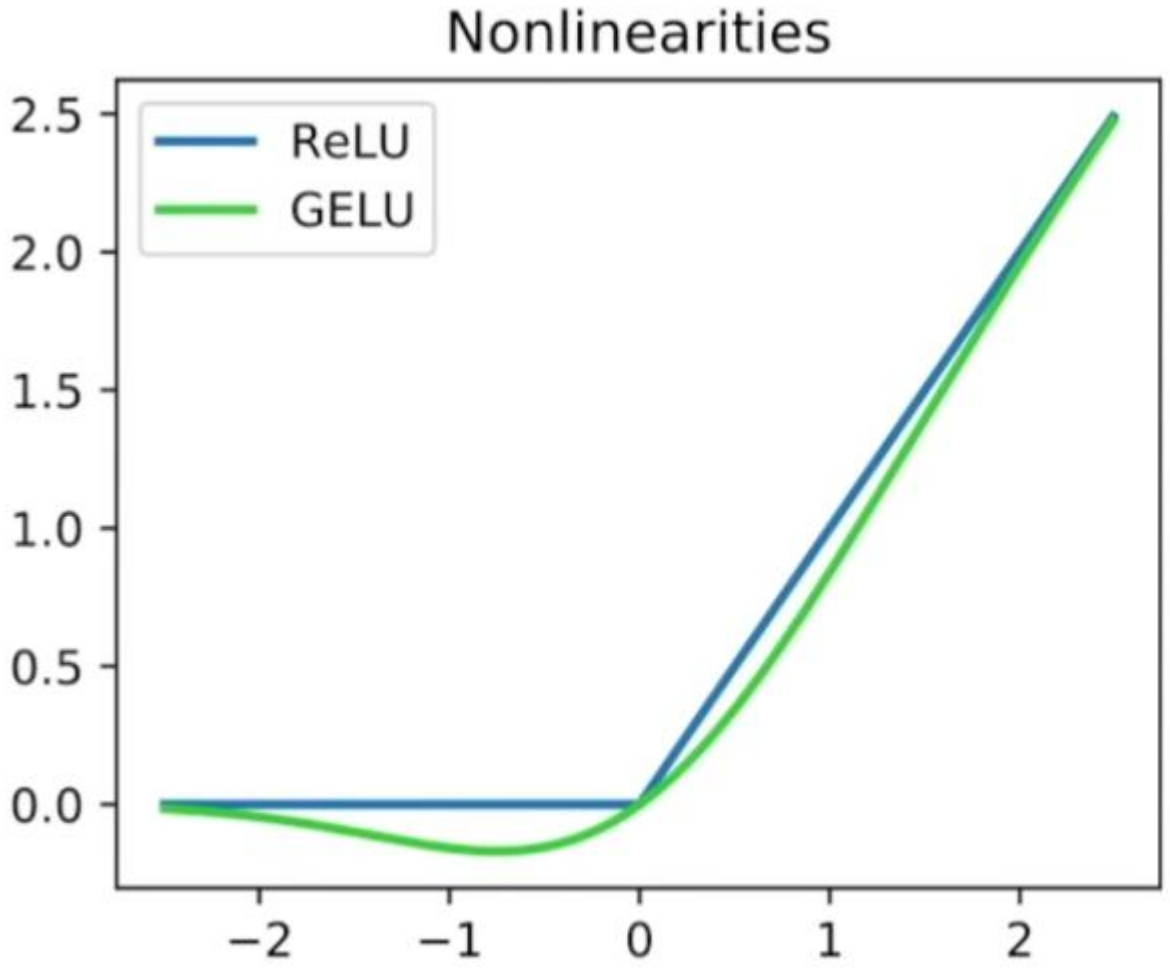
## Better Data Augmentation



# Improving Robustness

## Smoother Activations

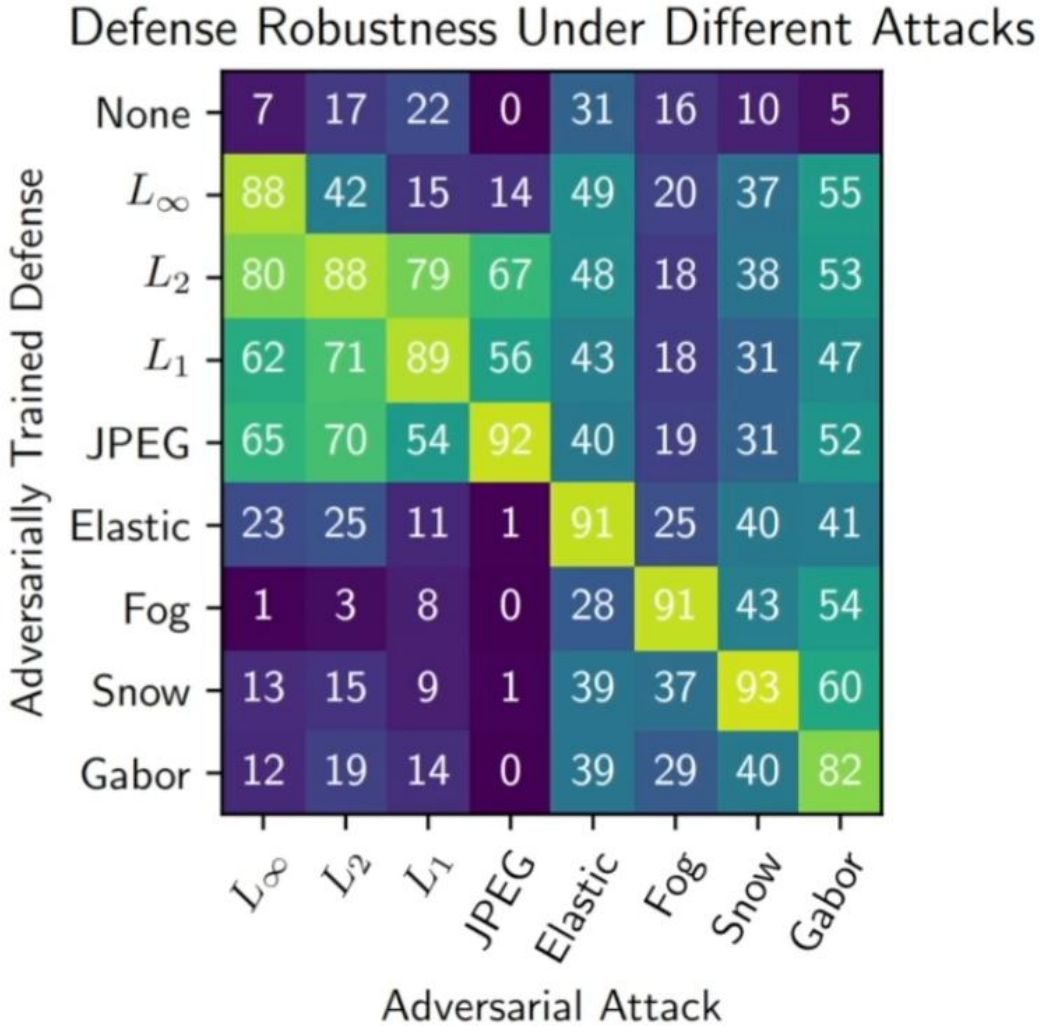
GELU > ReLU for adversarial training!




| Model | ImageNet Adversarial Accuracy |
|---|---|
| ResNet-50 with ReLUs | 26.41% |
| ResNet-50 with GELUs | 35.51% |

# Adversaries In Real-life?

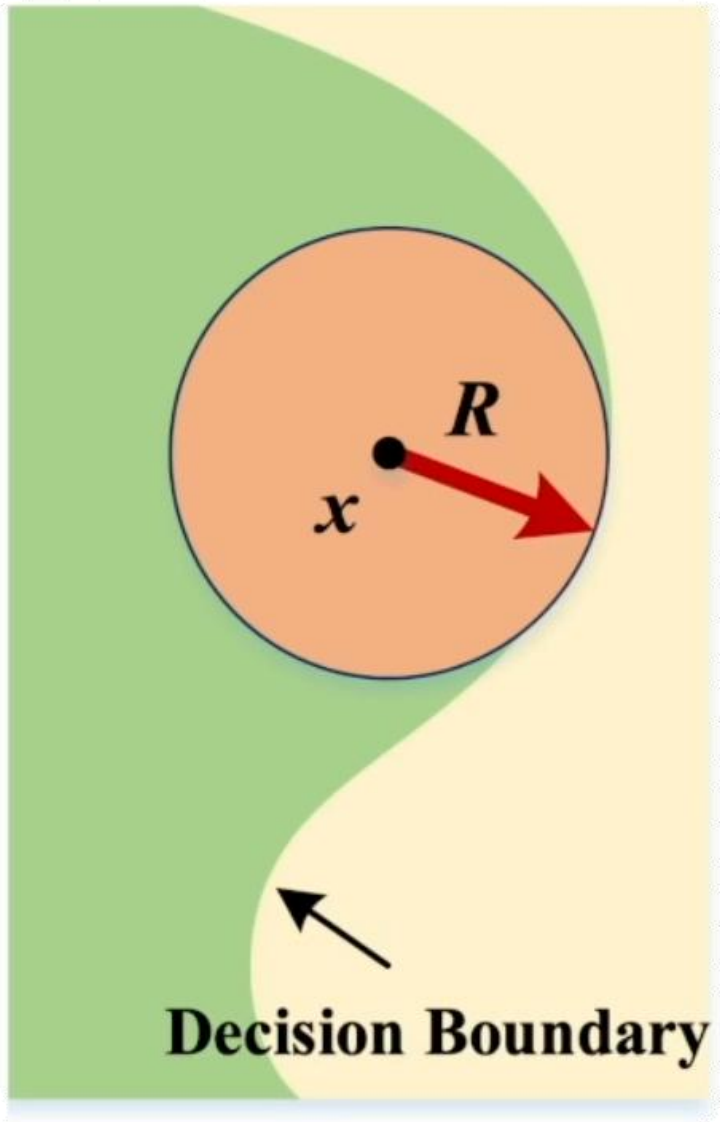
**Can They Be Unforeseen?**


# Is It Just For Fooling?

# Certifiability

- Create provable guarantees “certificates” for how a model behaves
  - AKA Driver’s License test for NNs
- You can sort of verify around a $p$-norm ball around training samples
- You can also create uncertainty models!

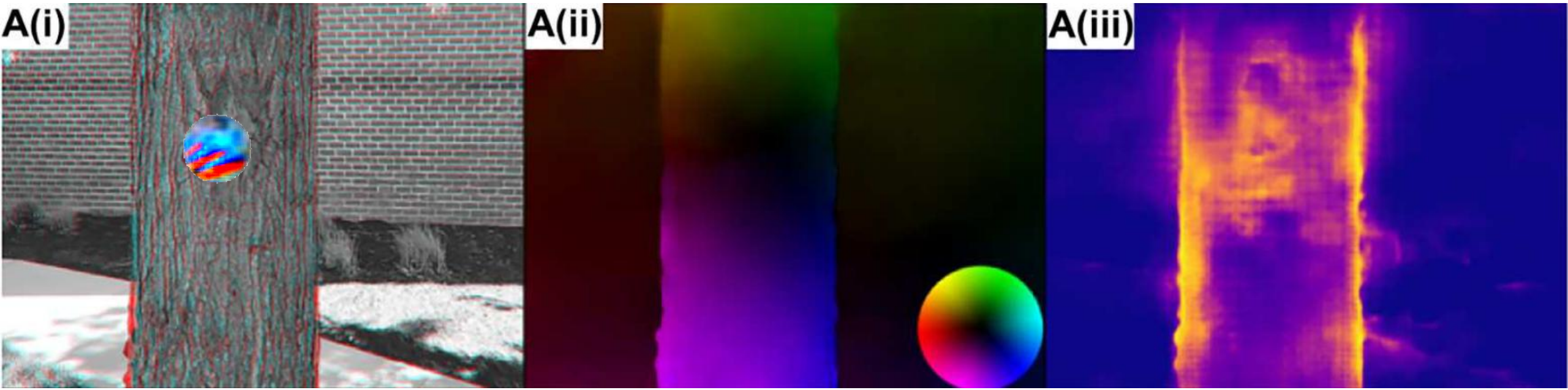



Certified Radius: $R$

Adversarial Robustness (youtube.com)

# Can’t Plan For Everything!

# Robustness Comes Also From Uncertainty!




Sanket, Nitin J., et al. "Ajna: Generalized deep uncertainty for minimal perception on parsimonious robots." Science Robotics 8.81 (2023): eadd5139.

# Next Class!

**Generative Models: VAEs, GANs, Attacking GANs**