Defending Against Multiple and Unforeseen Adversarial Videos

Shao-Yuan Lo 羅紹元

Johns Hopkins University

https://shaoyuanlo.github.io

January 10, 2022 at NCTU

About Me

- 2013-2017　EECS, NCTU（交大電資學士班）

- 2017-2019　EE, NCTU（交大電子所）

  Advisor: Prof. Hsueh-Ming Hang（杭學鳴教授）

- 2019-now　ECE, Johns Hopkins University (JHU)

  Advisor: Prof. Vishal M. Patel

- 2021 summer　Applied Scientist Intern at Amazon Lab126

 Manager: Jim Thomas & Cheng-Hao Kuo

Prof. Hsueh-Ming Hang

Prof. Vishal M. Patel

Dr. Jim Thomas

Research Areas

- Adversarial robustness (at JHU)
  - Adversarially robust video recognition
  - Adversarially robust novelty detection
  - Adversarially robust domain adaptation
  - Multi-perturbation robustness

- Domain adaptation (at Amazon)
  - Domain adaptive monocular depth estimation

- Semantic segmentation (at NCTU)
  - Real-time semantic segmentation
  - RGB-D semantic segmentation
  - Compressed domain semantic segmentation
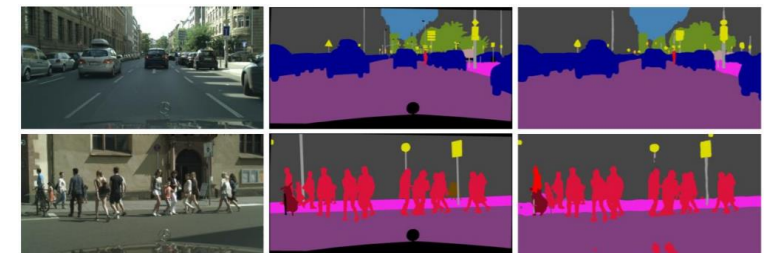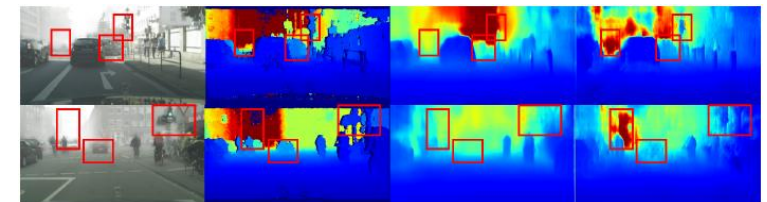
Outline

- Introduction to **adversarial examples**

- Our latest publication in **IEEE T-IP** (2022):
  "Defending Against Multiple and Unforeseen Adversarial Videos"

- Our other related works

What's Adversarial Example?

$$x_{adv} = x + \textcolor{red}{\delta}$$

$$f(\boldsymbol{x}_{adv}) \neq y$$

What's Adversarial Example?

- Adversarial examples are visually **similar** to **human** but can **fool** well-trained **deep networks**.

- Deep networks are **vulnerable** to adversarial examples.

$$x$$
"panda"
57.7% confidence

$$+ .007 \times$$

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

$$=$$

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

[Goodfellow et al. ICLR'15]

Generate Adversarial Examples

- Train a model
  - **min Loss(f(x), y; θ)**
  - **Minimize** the loss function w.r.t. **model parameters θ**

- Generate adversarial examples
  - Most common method: Gradient-based method, e.g., FGSM.
  - **max Loss(f(x+δ), y; θ)**
  - **Maximize** the loss function w.r.t. **adversarial perturbation δ**

Generate Adversarial Examples

- Generate adversarial examples
  - Most common method: Gradient-based method, e.g., FGSM.
  - **max Loss(f(x+δ), y; θ)**
  - **Maximize** the loss function w.r.t. **adversarial perturbation δ**

- Perturbation budget $\|δ\|$
  - Constrain the **magnitude** of perturbation, e.g., **Lp-norm**.
  - Constrain the **region** of perturbation, e.g., **patch attack**.

Adversary's Knowledge

- White-box attack
- Black-box attack
- Gray-box attack

Low — Adversary's Knowledge — High
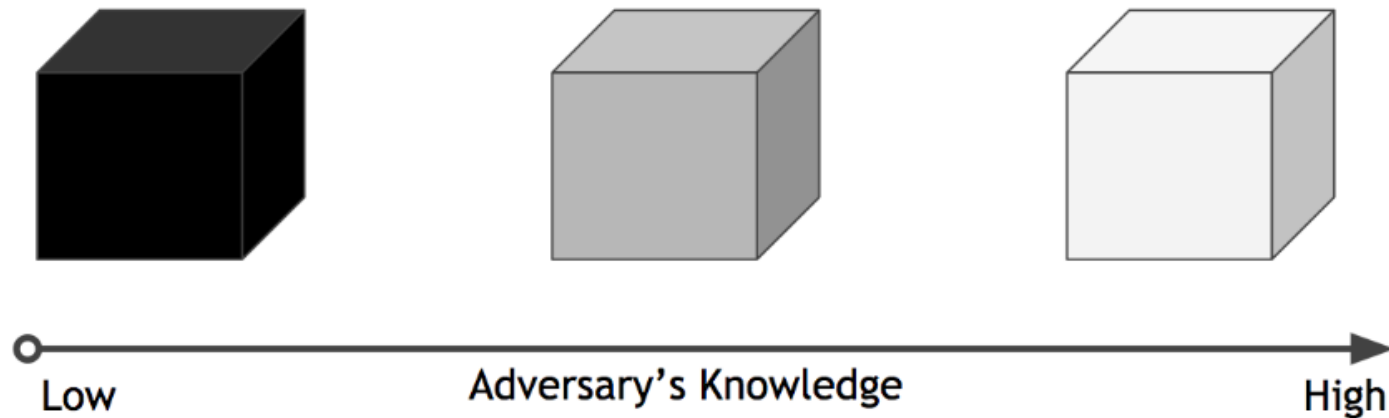
Untargeted/Targeted Attacks

- Untargeted attack

$$f(\boldsymbol{x}_{adv}) \neq y$$

$$L_{adv}(\boldsymbol{x}) = -L(\boldsymbol{x}, y)$$

- Targeted attack

$$f(\boldsymbol{x}_{adv}) = y_{adv}, \quad y_{adv} \neq y$$

$$L_{adv}(\boldsymbol{x}) = L(\boldsymbol{x}, y_{adv})$$

Adversarial Examples in Different Types

Original	Adversarial	Prediction	Original	Adversarial	Prediction
Aaron Tveit	Aaron Tveit	A.R. Rahman	Stop sign	Stop sign	Added lane
Abbie Cornish	Abbie Cornish	Abigail Breslin	Stop sign	Stop sign	Speed limit 35
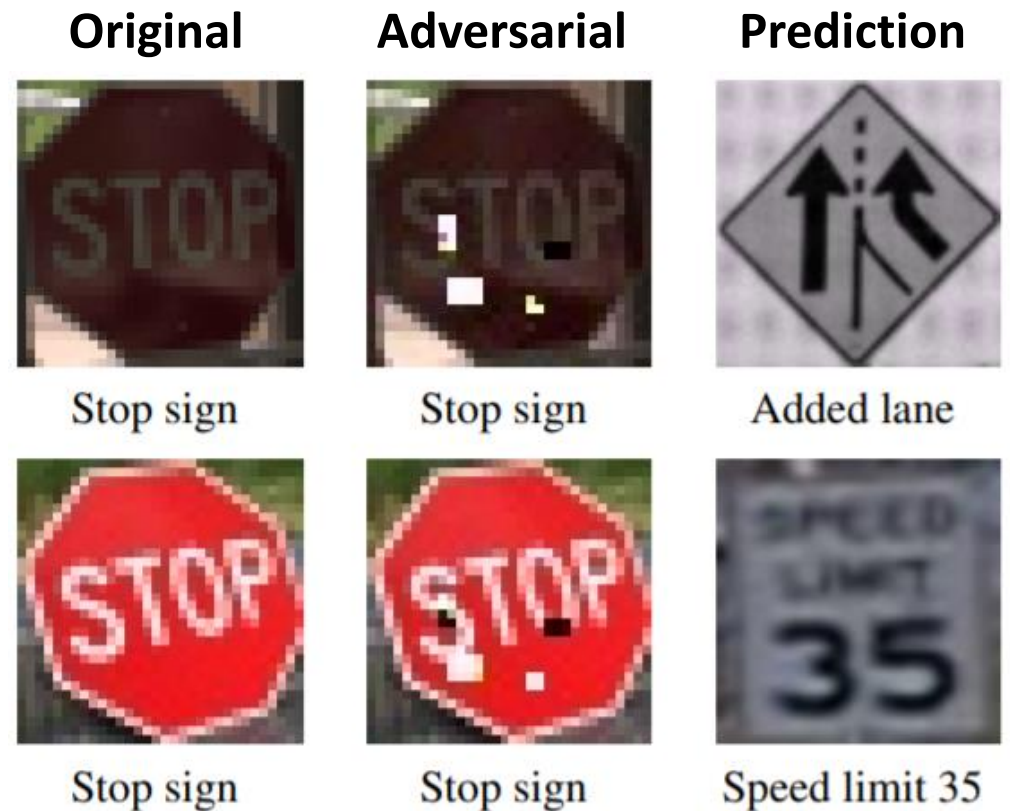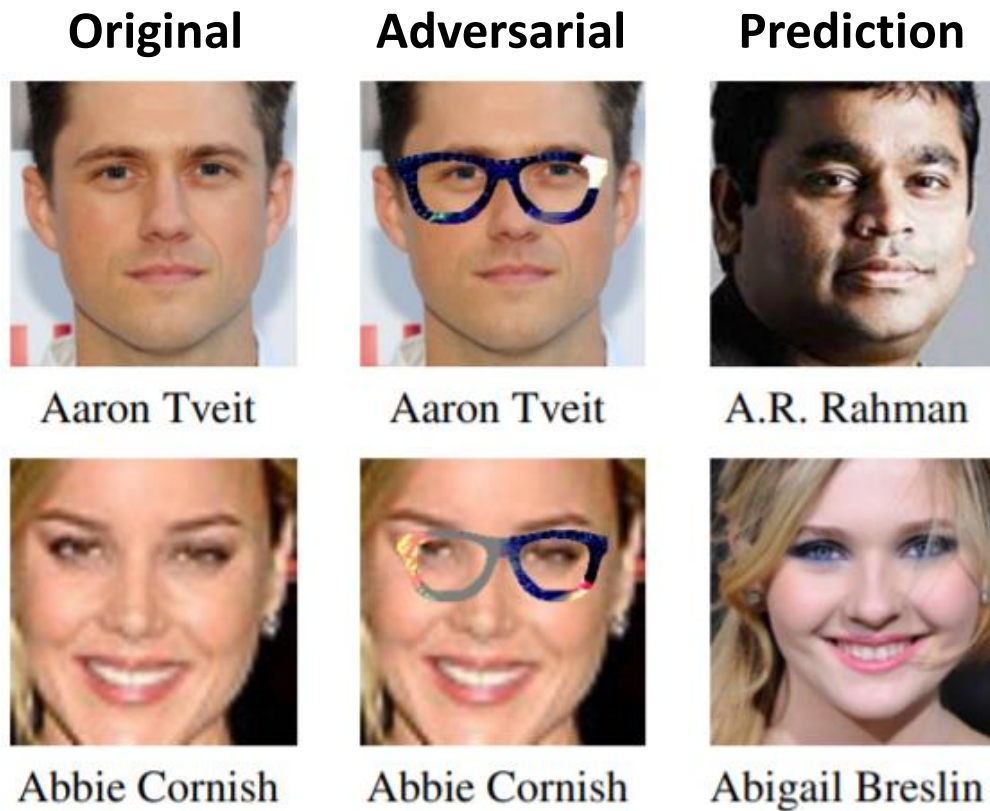
[Wu et al. ICLR'20]

Adversarial Examples in Physical World

[Hu et al. ICCV'21]

[Ranjan et al. ICCV'19]

Adversarial Examples in Different Tasks

Semantic segmentation

Object detection
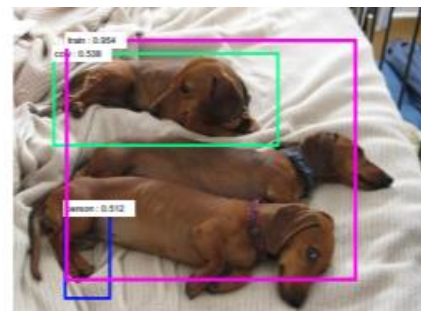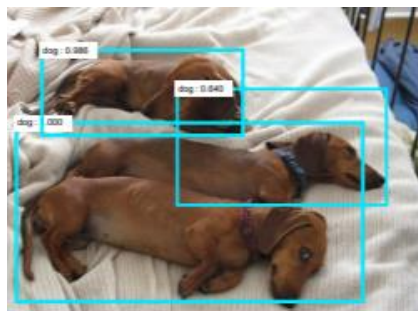
Optical flow

[Xie et al. ICCV'17]

[Ranjan et al. ICCV'19]

Adversarial Defenses

- **Image transformation**: Remove perturbations from input images.

$$f(\boldsymbol{x}_{adv}) \neq y$$

$$f(\textcolor{red}{\boldsymbol{T}}(\boldsymbol{x}_{adv})) = y$$

- **Adversarial training**: Enhance the robustness of networks itself.

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{(x,y) \sim \mathbb{D}} \left[\max_{\delta \in \mathbb{S}} L(x + \delta, y; \theta) \right]$$

Image Transformation-based Defenses

- Image preprocessing methods:
  - **Color precision reduction** (pixel value quantization)
  - **JPEG compression** (frequency domain quantization)
  - **Denoising** (Gaussian blur, median, mean, bilateral, non-local means, etc.)
  - **Color space** (RGB, HSV, YUV, LAB, etc.)
  - **Contrast** (histogram equalization)
  - **Noise injection** (add noise on adversarial examples)
  - **FFT perturbation** (similar to JPEG)
  - **Swirl** (rotation)
  - **Resizing**
  - **Gray scale**

- Generative model methods:
  - **Defense-GAN**
    [Samangouei et al. ICLR'18]
  - **PixelDefend**
    [Song et al. ICLR'18]

[Das et al. KDD'18]
[Xu et al. NDSS'18]
[Guo et al. ICLR'18]
[Raff et al. CVPR'19]

Image Transformation-based Defenses

- [Athalye et al. ICML'19] proposed **adaptive attacks**, which defeat most image transformation-based defenses.

- Strong white-box attacks are generated through **gradients**, e.g., FGSM and PGD attacks.

- Image transformation-based defenses mostly rely on **gradient masking**, which can be defeated by adaptive attacks.

- Three types of masked gradients:
  - Shattered gradients ← BPDA
  - Stochastic gradients ← EOT
  - Exploding & vanishing gradients ← BPDA or EOT or Both

Defense	Dataset	Distance	Accuracy
Buckman et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%*
Ma et al. (2018)	CIFAR	0.031 (ℓ_∞)	5%
Guo et al. (2018)	ImageNet	0.005 (ℓ_2)	0%*
Dhillon et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%
Xie et al. (2018)	ImageNet	0.031 (ℓ_∞)	0%*
Song et al. (2018)	CIFAR	0.031 (ℓ_∞)	9%*
Samangouei et al. (2018)	MNIST	0.005 (ℓ_2)	55%**
Madry et al. (2018)	CIFAR	0.031 (ℓ_∞)	47%
Na et al. (2018)	CIFAR	0.015 (ℓ_∞)	15%

Adversarial Training

- Adversarial training is a strong defense against white-box attacks.
- **Core idea: Train with adversarial examples.**
- Adversarial training does **not** cause masked gradients.
- It has been widely used as a standard baseline defense.

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\max_{\delta \in \mathbb{S}} L(x + \delta, y; \theta) \right]$$
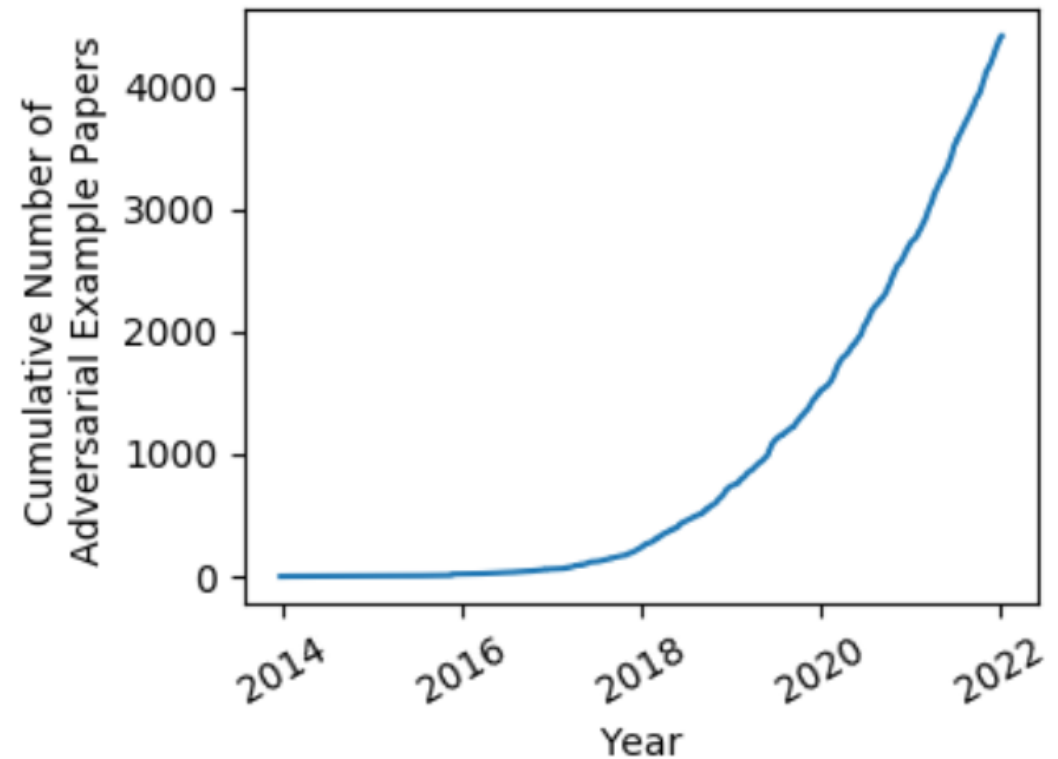
Generate adversarial examples

Train model parameters

[Madry et al. ICLR'18]

Why Study Adversarial Examples?

- Deep learning models are being widely used in real-world applications, such as autonomous driving. Their **safety** is critical.

- We aim to build **robust** DL models that we can **trust**.

https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html

Our Latest Publication in IEEE T-IP

Defending Against Multiple and Unforeseen Adversarial Videos

Shao-Yuan Lo, *Student Member, IEEE* and Vishal M. Patel, *Senior Member, IEEE*

IEEE Transactions on Image Processing (T-IP), 2022

Why Videos?

- Most research in adversarial examples focuses on static **images**.

- Adversarial attacks and defenses for **videos** are less explored.

- To the best of our knowledge, this work is the **first** defense against white-box attacks in the video domain.

- We provide **comprehensive baseline results** for adversarial robustness in the video domain.

Adversarial Videos

- Video is a stack of consecutive images.
- A naïve way to generate adversarial videos:
  Use image-based method directly.

$$x^{adv} = x + \epsilon \cdot sign(\nabla_x L(x, y; \theta))$$

$$Image: x \in R^{C \times H \times W}$$

$$Video: x \in R^{\textcolor{red}{F} \times C \times H \times W}$$

Adversarial Framing (AF)

correct: Boston bull
unattacked: Boston bull
attacked: maypole

correct: ocarina
unattacked: loupe
attacked: maypole

correct: tusker
unattacked: tusker
attacked: maypole

correct: gas pump
unattacked: gas pump
attacked: maypole

correct: Egyptian cat
unattacked: tabby
attacked: maypole

Task: Action recognition
Dataset: UCF-101

Attack	$W = 1$	$W = 2$	$W = 3$	$W = 4$
None	85.95%			
RF	82.57%	80.53%	81.11%	79.74%
BF	84.94%	84.73%	84.75%	84.59%
AF	65.77%	22.12%	9.45%	2.05%

[Zajac et al. AAAI'19]

Salt-and-Pepper Attack (SPA)

- Add unbounded perturbations on a number of randomly selected pixels.

- The perturbation looks like salt-and-pepper noise.

- A kind of L0-norm attack.

- Decrease action recognition accuracy from **89.0%** to **8.4%** on UCF-101.
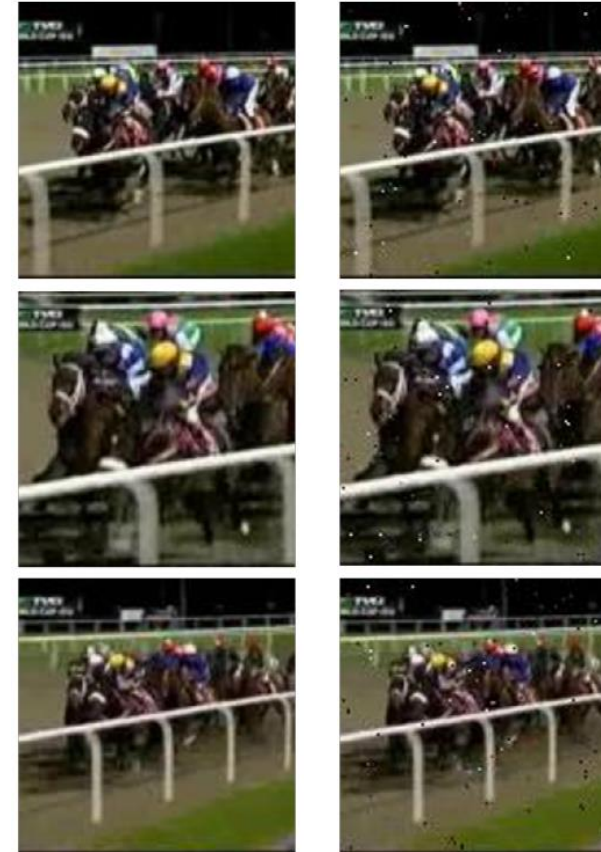
Clean   SPA

Adversarial Video Types

- PGD:
Projective gradient descent
[Madry et al. ICLR'18]

- ROA:
Rectangular occlusion
[Wu et al. ICLR'20]

- AF:
Adversarial Framing
[Zajac et al. AAAI'19]

- SPA:
Salt-and-Pepper noise

Clean   PGD   ROA   AF   SPA

Adversarial Video Types

- PGD:
  Projective gradient descent
  [Madry et al. ICLR'18]

- ROA:
  Rectangular occlusion
  [Wu et al. ICLR'20]

- AF:
  Adversarial Framing
  [Zajac et al. AAAI'19]

- SPA:
  Salt-and-Pepper noise

| Clean | PGD | ROA | AF | SPA |

How to simultaneously defend against multiple types of attacks?

Problem: Multi-perturbation Robustness

- Standard adversarial training has poor **multi-perturbation robustness**.
- Training: δ_{PGD}
- Test: Clean, δ_{PGD}, δ_{ROA}, δ_{AF}, δ_{SPA}

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\max_{\delta\in\mathbb{S}} L(x+\delta, y; \theta) \right]$$

Generate **one type** of adversarial examples

Train model parameters

[Madry et al. ICLR'18]

Problem: Multi-perturbation Robustness

- Dataset: UCF-101 (action recognition)

- Model: 3D ResNeXt-101

- Attack setting:
  PGD Linf: ε=4/255, T=5
  ROA: patch size=30x30
  AF: width=10
  SPA: #pixels=100, T=5

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**89.0**	3.3	0.5	1.6	8.4	20.6	0.0
AT-PGD	78.6	**49.0**	5.0	0.6	67.1	40.1	0.3
AT-ROA	82.6	12.5	**69.0**	54.0	17.6	47.1	7.9
AT-AF	84.6	7.1	3.9	**80.5**	12.2	37.7	2.1
AT-SPA	83.5	36.9	2.6	0.7	**69.5**	38.6	0.2

Problem: Multi-perturbation Robustness

- **Average** adversarial training is better, but not enough.
- Training: Clean, δ_{PGD}, δ_{ROA}, δ_{AF}, δ_{SPA}
- Test: Clean, δ_{PGD}, δ_{ROA}, δ_{AF}, δ_{SPA}

$$\theta^* = \arg\min_\theta \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\sum_{i=1}^{N} \max_{\delta_i \in \mathbb{S}_i} L(x + \delta_i, y; \theta) \right]$$

Generate **multiple types** of adversarial examples

Train model parameters

[Tramèr & Boneh NeurIPS'19]

Problem: Multi-perturbation Robustness

| Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**89.0**	3.3	0.5	1.6	8.4	20.6	0.0
AT-PGD	78.6	**49.0**	5.0	0.6	67.1	40.1	0.3
AT-ROA	82.6	12.5	**69.0**	54.0	17.6	47.1	7.9
AT-AF	84.6	7.1	3.9	**80.5**	12.2	37.7	2.1
AT-SPA	83.5	36.9	2.6	0.7	**69.5**	38.6	0.2
AVG [30] (NeurIPS'19)	74.5	43.1	55.6	3.5	57.2	46.8	3.5

Observation: Distinct Data Distributions

- Why average adversarial training is **not** an ideal strategy?

- Example: **Clean** vs. **PGD**.

- Clean and PGD have **distinct** data distributions.

- The statistics estimation at **BN** may be confused when facing a mixture distribution.
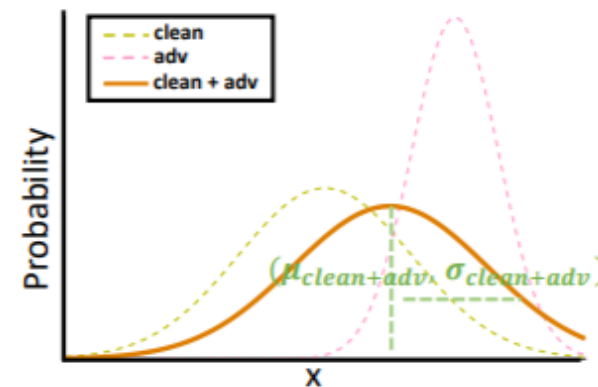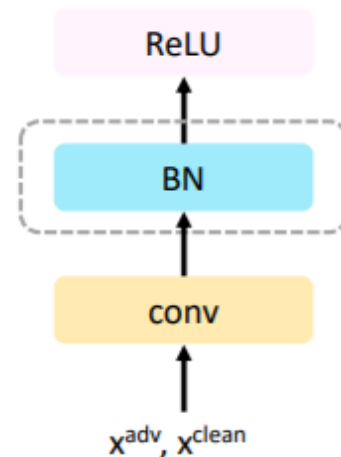
[Xie et al. CVPR'20]

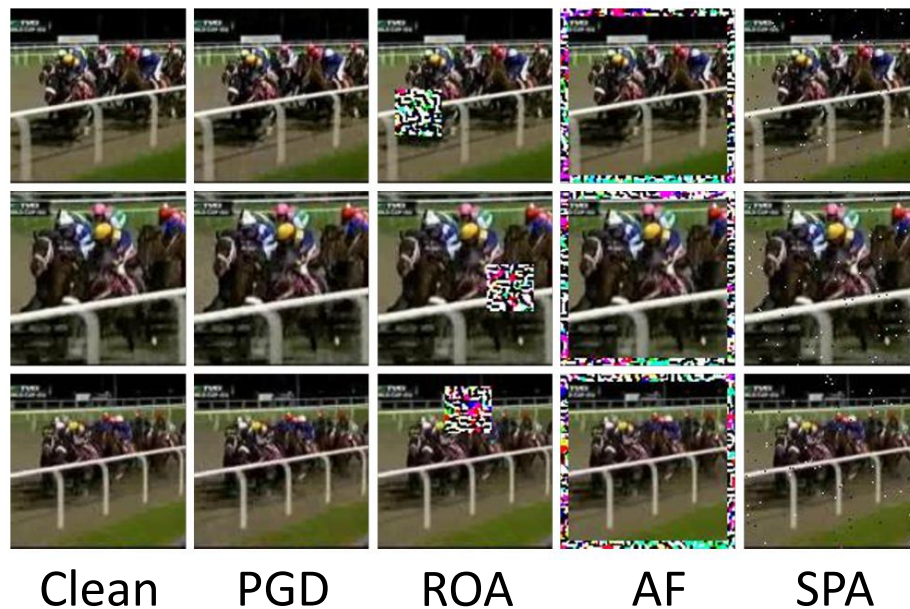# Observation: Distinct Data Distributions

- Example: **Clean** vs. **PGD**.

- An **auxiliary BN** guarantees that data from different distributions are normalized separately.
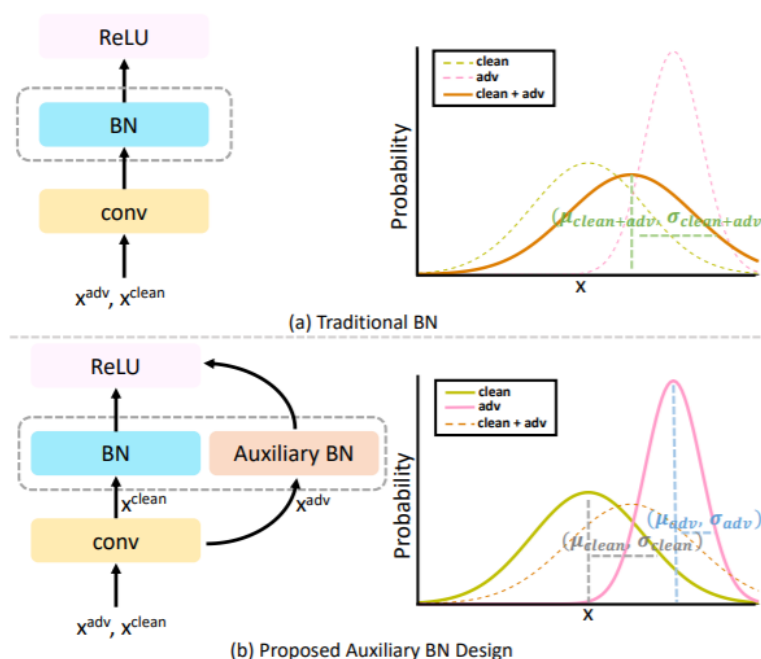

[Xie et al. CVPR'20]

Extension for Multi-perturbation Robustness

- What about **multiple** attack types?

- Example: Clean, PGD, ROA, AF, SPA

- Our assumption: Different attack types have **distinct** data distributions.

Clean   PGD   ROA   AF   SPA

Extension for Multi-perturbation Robustness

- What about **unforeseen** attack types?
- Example:
  - **Known**: Clean, PGD, ROA
  - **Unforeseen**: AF, SPA
- **Lp-norm attacks**: PGD, SPA
- **Physically realizable attacks**: ROA, AF
- Our assumption: Similar attack types have **similar** data distributions.

Clean   PGD   ROA   AF   SPA

33
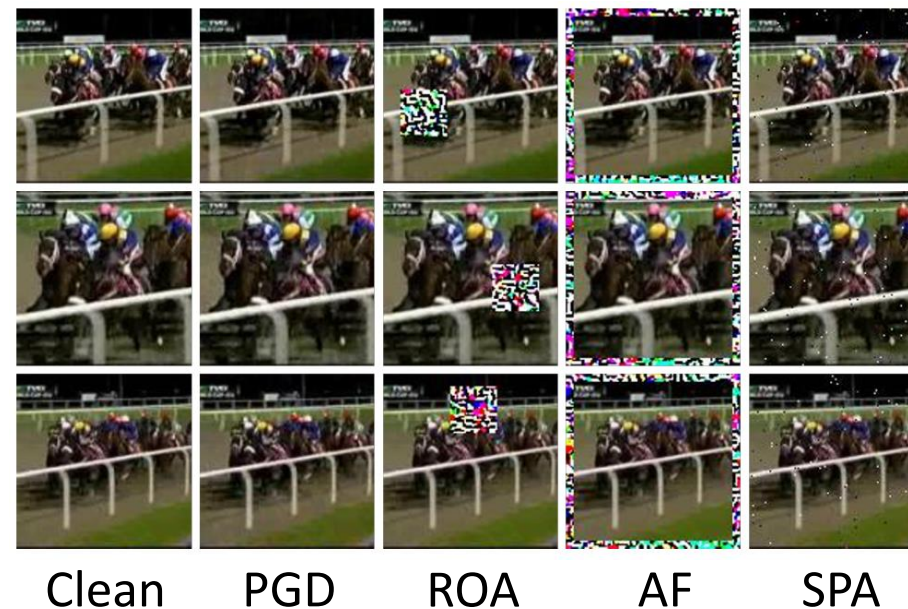
Our Solution: Multi-BN Structure

- Example:
  - **Known**: Clean, PGD, ROA
  - **Unforeseen**: AF, SPA

- **Lp-norm attacks**: PGD, SPA

- **Physically realizable attacks**: ROA, AF
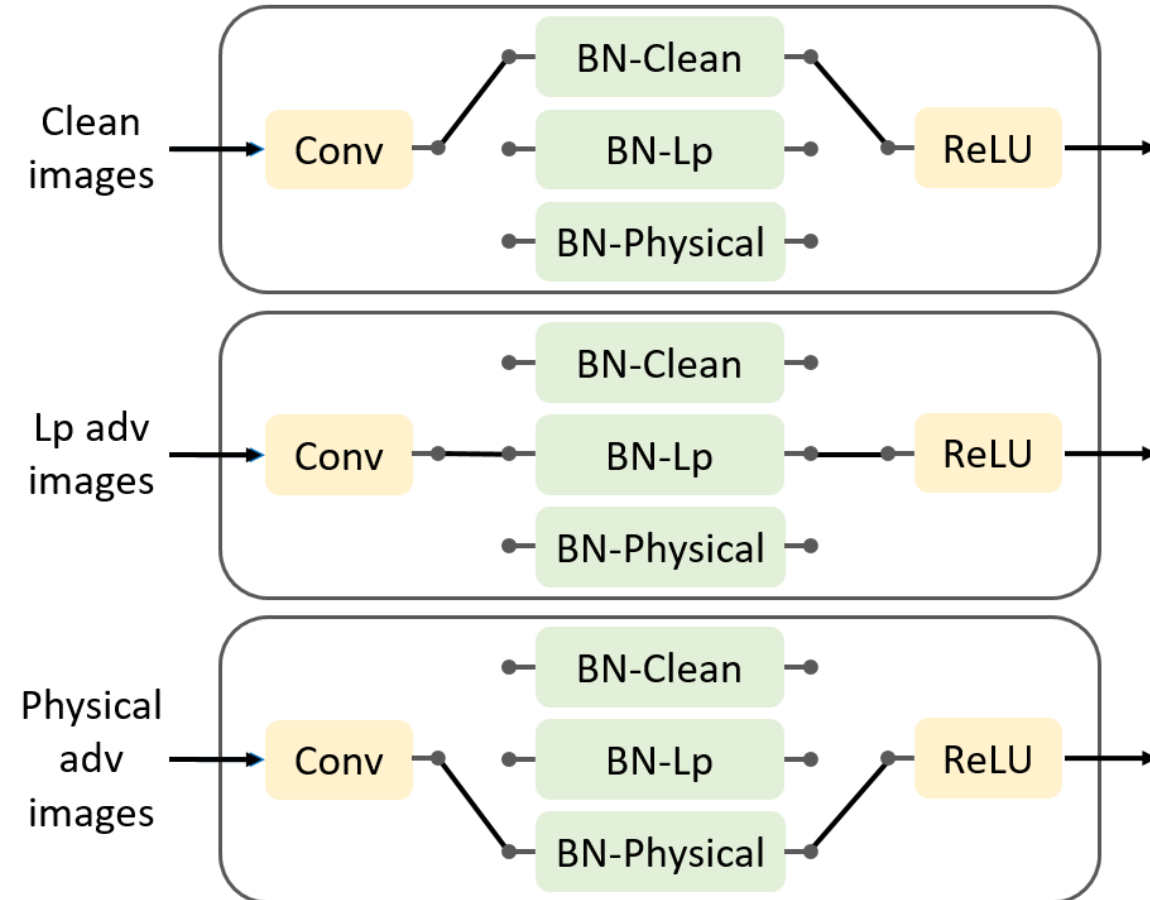
Our Solution: Multi-BN Structure

- Training: Clean, δ_{PGD}, δ_{ROA}

- Test: Clean, δ_{PGD}, δ_{ROA}, δ_{AF}, δ_{SPA}

$$\theta = \theta^c + \sum_{i=0}^{N} \theta_i^b$$

$$\theta^* = \arg \min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\underbrace{L(x, y; \theta^c, \theta_0^b)}_{\text{Clean data}} + \underbrace{\sum_{i=1}^{N} \max_{\delta_i \in \mathbb{S}_i} L(x + \delta_i, y; \theta^c, \theta_i^b)}_{\text{Generate multiple types of adversarial examples}} \right]$$
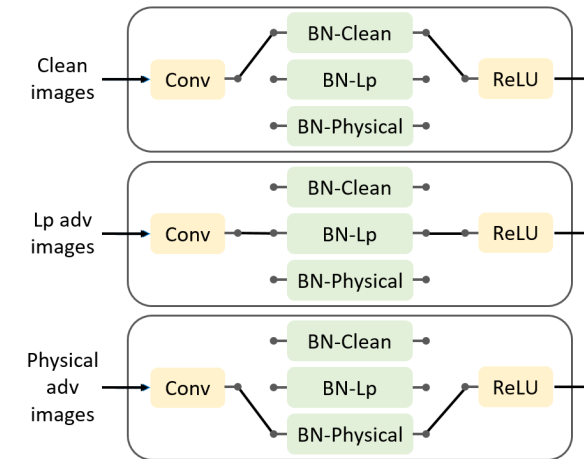
Train model parameters

Our Solution: Multi-BN Structure

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**89.0**	3.3	0.5	1.6	8.4	20.6	0.0
AT-PGD	78.6	**49.0**	5.0	0.6	67.1	40.1	0.3
AT-ROA	82.6	12.5	**69.0**	54.0	17.6	47.1	7.9
AT-AF	84.6	7.1	3.9	**80.5**	12.2	37.7	2.1
AT-SPA	83.5	36.9	2.6	0.7	**69.5**	38.6	0.2
MultiBN-manual	83.7	46.4	65.6	57.0	60.4	**62.6**	**40.7**

Our Solution: Multi-BN Structure

- Performance (%) of each BN branch on the five input types.

BN Branch	Clean	PGD	ROA	AF	SPA
BN-Clean	**83.7**	21.3	13.5	5.9	23.8
BN-Lp	79.0	**46.4**	7.7	1.9	**60.4**
BN-Physical	83.0	23.5	**65.6**	**57.0**	26.6

- Our assumptions are valid:
  - Different attack types have **distinct** data distributions.
  - Similar attack types have **similar** data distributions.

BN Selection Module

- At inference time, the input data have to pass through the corresponding BN branch **automatically**.

- The **adversarial video detector** is achieved by a video classifier.

- **Gumbel-Softmax** function [Jang et al. ICLR'17] is a **differentiable** approximation of the *argmax* operation (vanilla Softmax also works).
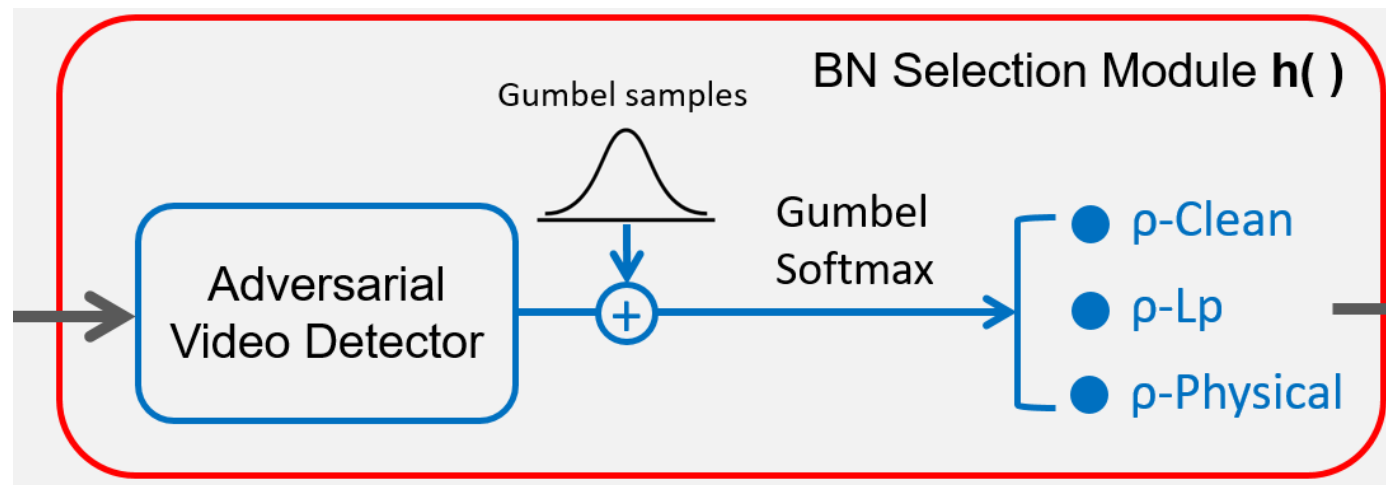
BN Selection Module

- Use Gumbel-Softmax scores as ratio factors to weight each BN branch's output features.

$$\hat{z} = \sum_{k=1}^{K} \rho_k z_k$$

$K: \# \, BN \, branches$

$\rho_1, \dots, \rho_K : ratio \, factors$

$z_1, \dots, z_K : each \, BN \, branch's \, output \, features$

$\hat{z} : weighted \, features$
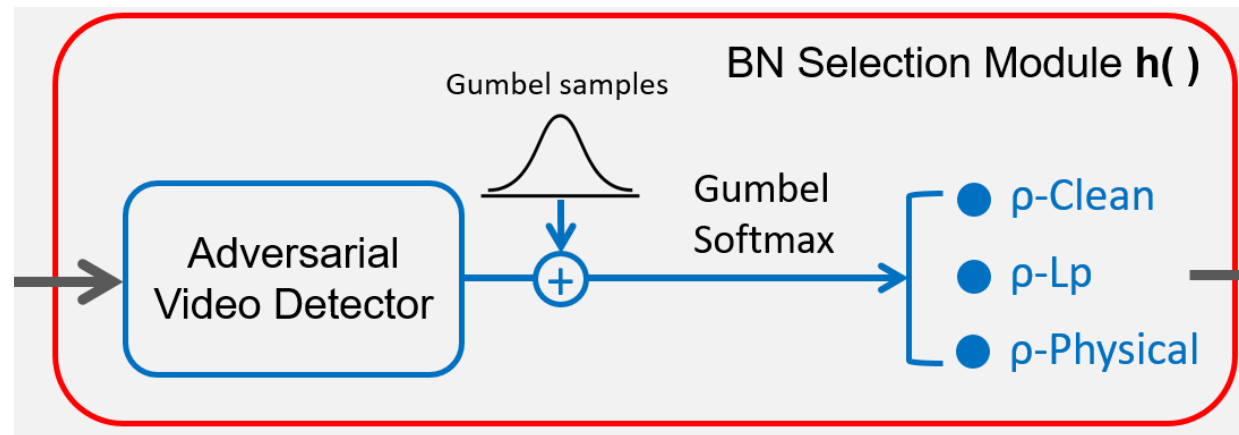
Entire Framework

- End-to-end pipeline:

$$\tilde{y} = f(x + \delta_i; \theta^c, \theta^b, \theta^{det})$$
$$= g(x + \delta_i, h(x + \delta_i; \theta^{det}); \theta^c, \theta^b).$$
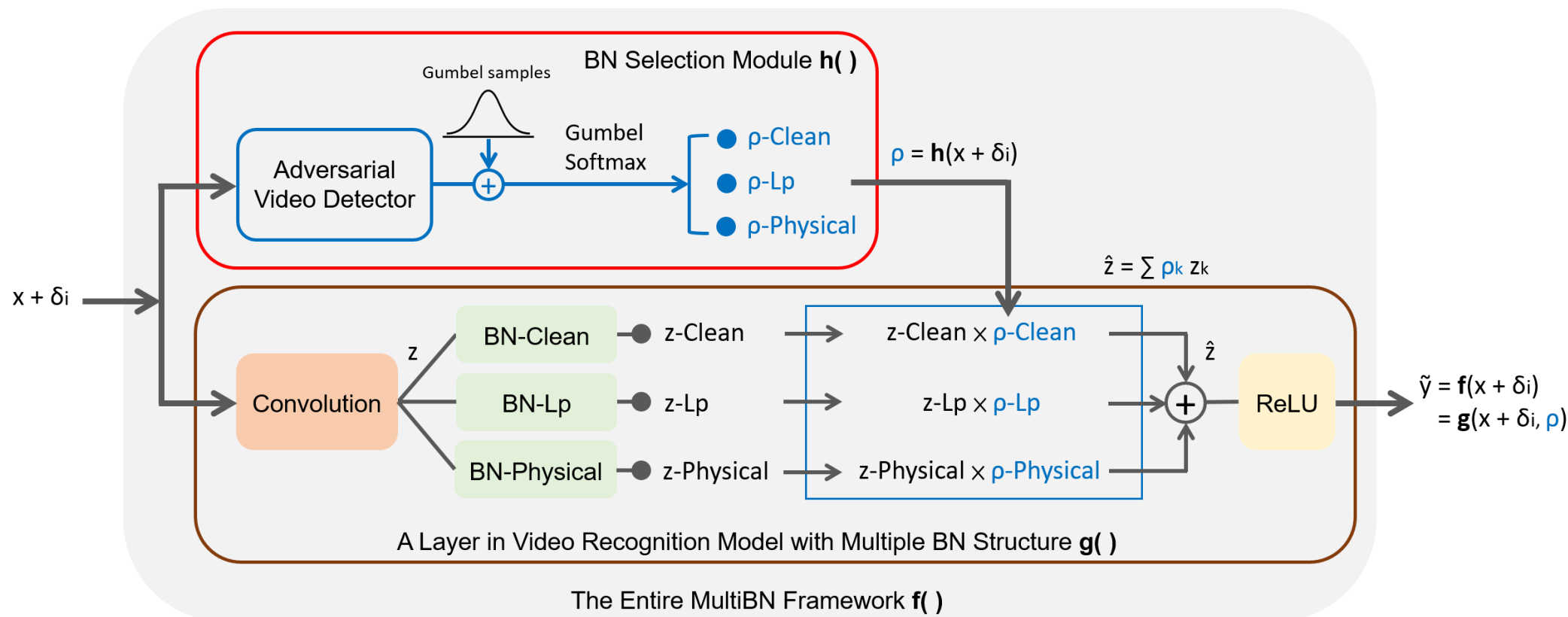
Entire Framework

- End-to-end training:

$$\theta^* = \arg\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathbb{D}} \left[L(x, y; \theta) + \lambda \cdot L(x, y^{det}; \theta^{det}) \right.$$

$$+ \sum_{i=1}^{N} \left(\max_{\delta_i \in \mathbb{S}_i} L(x + \delta_i, y; \theta) + \lambda \cdot L(x + \delta_i, y^{det}; \theta^{det})) \right]$$

The Entire MultiBN Framework **f( )**
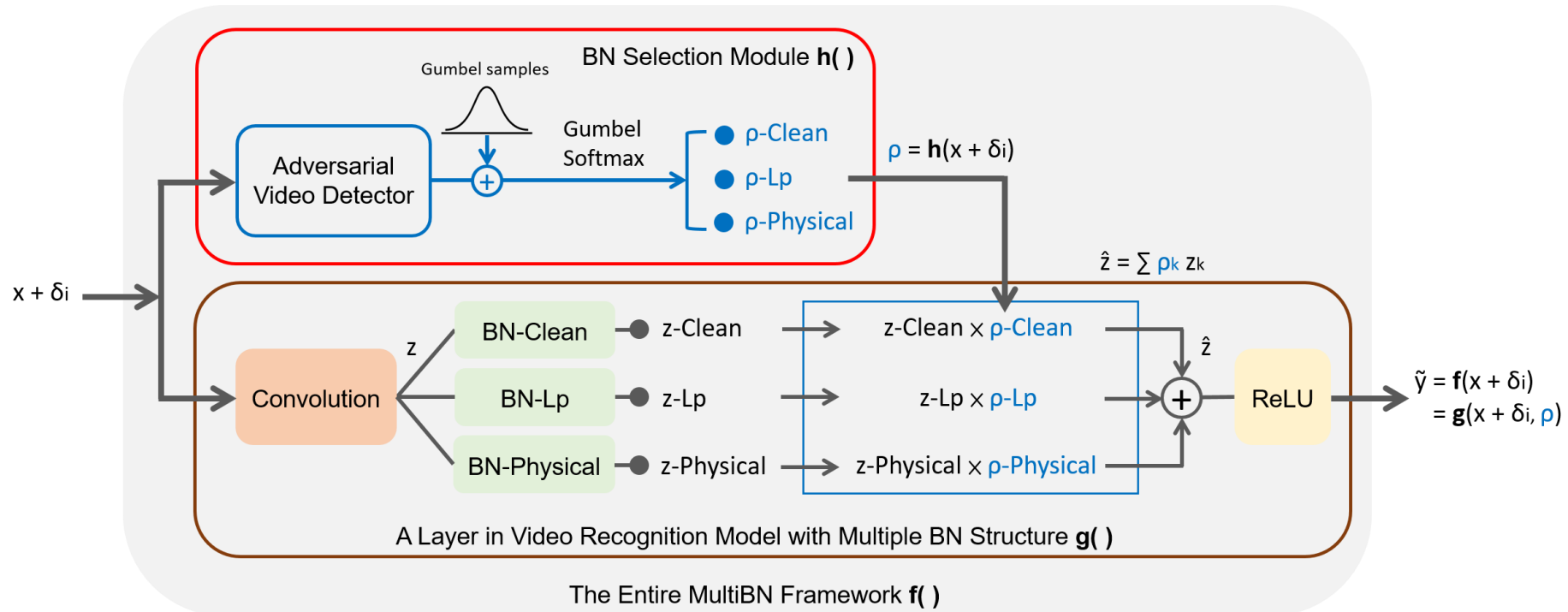
Experimental Setup

- Dataset: UCF-101 (action recognition)

- Model: 3D ResNeXt-101

- Attack setting:
  PGD Linf: $\varepsilon=4/255$, T=5
  ROA: patch size=30x30
  AF: width=10
  SPA: #pixels=100, T=5

- White-box attacks

- Untargeted attacks

Results

Dataset: UCF-101

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**89.0**	3.3	0.5	1.6	8.4	20.6	0.0
TRADE [19] (ICML'19)	82.3	29.0	5.7	3.3	42.2	32.5	1.9
AVG [26] (NeurIPS'19)	68.9	38.1	51.4	18.5	49.6	45.3	17.3
MAX [26] (NeurIPS'19)	72.8	32.5	31.0	5.8	49.4	38.3	5.5
MSD [27] (ICML'20)	70.2	43.2	1.7	1.6	**56.0**	34.6	0.7
MultiBN (ours)	74.2	**44.6**	**58.6**	**44.3**	53.7	**55.1**	**34.8**

Dataset: HMDB-51

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**65.1**	0.0	0.0	0.0	0.3	13.1	0.0
TRADE [19] (ICML'19)	54.8	6.8	0.3	0.0	20.5	16.5	0.0
AVG [26] (NeurIPS'19)	39.0	14.3	17.1	2.8	26.2	19.9	1.4
MAX [26] (NeurIPS'19)	48.6	13.9	16.0	0.1	30.3	21.8	0.0
MSD [27] (ICML'20)	41.4	18.2	0.1	0.0	**31.2**	18.2	0.0
MultiBN (ours)	51.1	**22.0**	**23.7**	7.8	29.9	**26.9**	**5.0**

43

Results: Robustness Against Adaptive Attacks

- Construct an adaptive attack, which **jointly** attacks the **target model part** and the **BN selection module part**.

- The intuition is to generate adversarial examples which can also fool the BN selection module to let it select the incorrect BN branch, and thus become easier to fool the target model.

$$\delta = \arg\max_{\delta \in \mathbb{S}} \left[L(x + \delta, y; \theta) + \lambda \cdot L(x + \delta, y^{det}; \theta^{det}) \right]$$
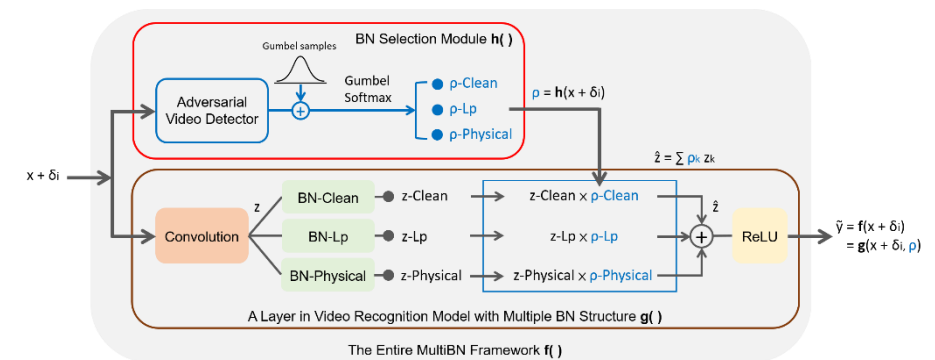
Results: Robustness Against Adaptive Attacks

- The canonical attack has the greatest attacking strength.
- The proposed MultiBN is robust against adaptive attacks.
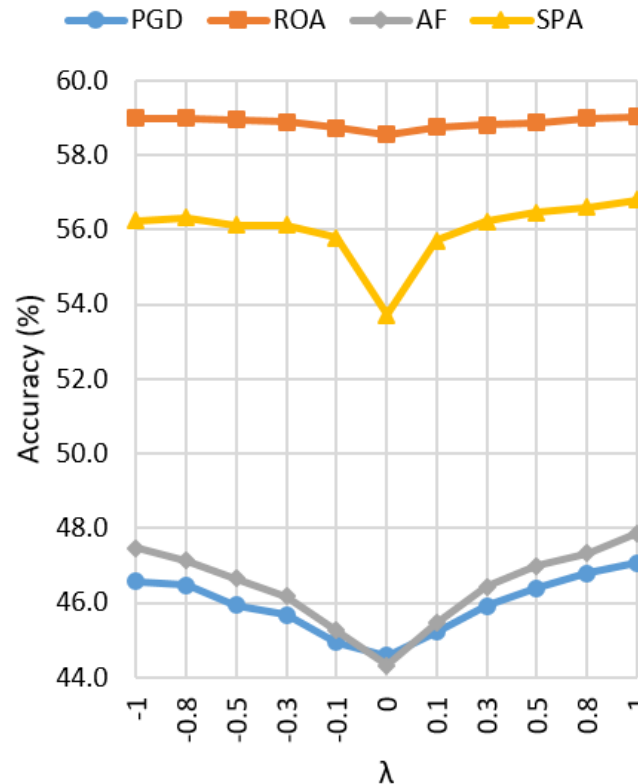
Results: Different Attack Budget
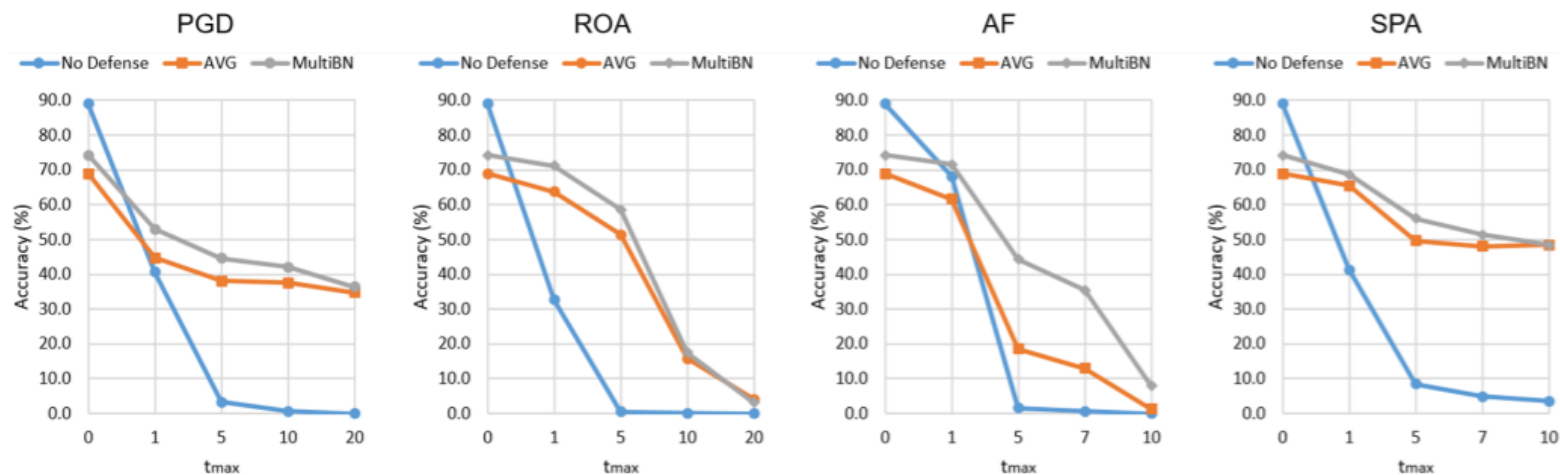
Fig. 3: Results (%) under the four attack types with varied numbers of attack iterations t_{max}.
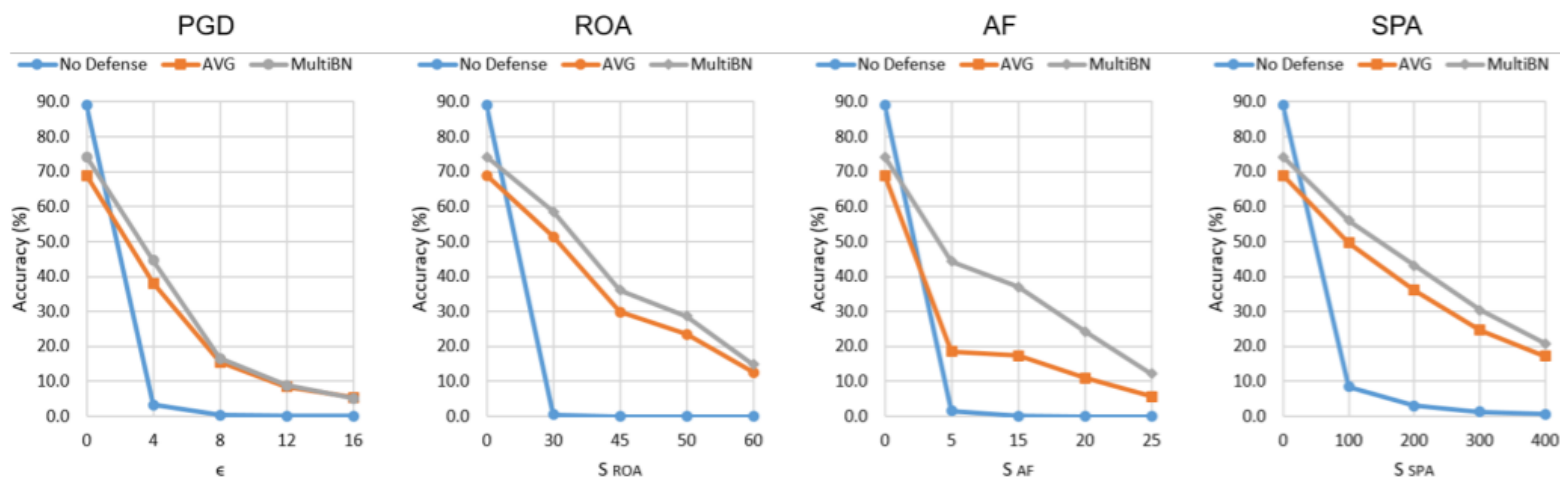
Fig. 4: Results (%) under the four attack types with varied perturbation bounds.

Results: Robustness Against Black-box Attacks

- Generate adversarial videos on a **surrogate** model:
  3D Wide ResNet-50

- Test on the **target** model: 3D ResNeXt-101

Model	Clean	PGD	ROA	AF	SPA	Union
TRADE [23] (ICML'19)	**82.3**	**81.0**	60.8	65.0	**78.0**	49.3
AVG [30] (NeurIPS'19)	68.9	68.4	68.0	62.0	68.4	56.2
MAX [30] (NeurIPS'19)	72.8	72.4	71.4	63.5	71.9	57.9
MSD [31] (ICML'20)	70.2	69.8	40.1	52.2	69.1	31.3
MultiBN (ours)	74.2	73.6	**74.0**	**72.4**	71.5	**63.5**

Results on Images

- Dataset: CIFAR-10
- Model: ResNet-18

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**94.3**	0.0	4.7	0.1	16.3	23.1	0.0
TRADE [23] (ICML'19)	71.4	14.7	34.7	30.4	52.8	40.8	10.1
AVG [30] (NeurIPS'19)	86.4	47.2	53.6	60.5	<u>67.8</u>	63.1	28.1
MAX [30] (NeurIPS'19)	87.7	46.3	60.0	54.6	**73.6**	<u>64.4</u>	<u>33.7</u>
MSD [31] (ICML'20)	93.0	**52.7**	6.7	7.1	59.6	43.8	2.2
MultiBN (ours)	<u>94.2</u>	<u>49.7</u>	**74.9**	**66.7**	60.9	**69.3**	**36.9**

Our other related works

Overcomplete Representations Against Adversarial Videos (OUDefend)

- A typical autoencoder downsamples features and learns **undercomplete** representations.

Input image — Latent Space Representation — Reconstructed image

https://ai.plainenglish.io/convolutional-autoencoders-cae-with-tensorflow-97e8d8859cbe.

- OUDefend learns both **undercomplete** representations and **overcomplete** representations (upsample features)

O-branch

- 3x3x3 Conv 3D + Upsampling
- 3x3x3 Conv 3D + Downsampling
- 1x1x1 Conv 3D

Input features

Output features

U-branch

50

Shao-Yuan Lo, Jeya Maria Jose Valanarasu, and Vishal M. Patel. Overcomplete Representations Against Adversarial Videos. ICIP 2021.
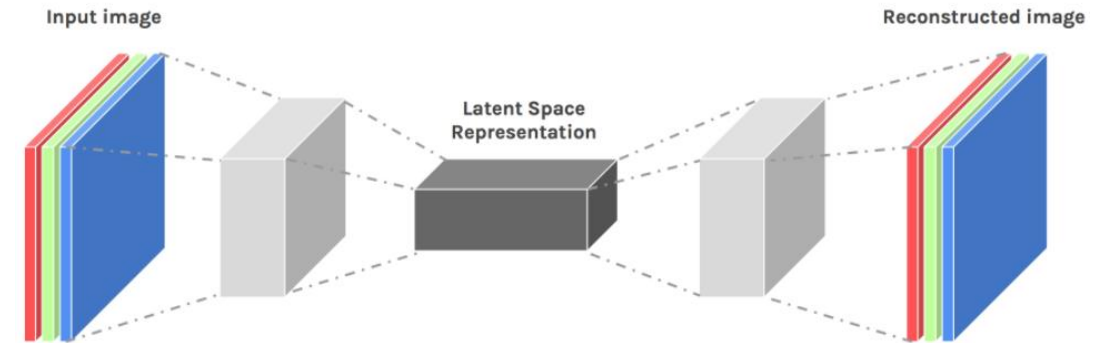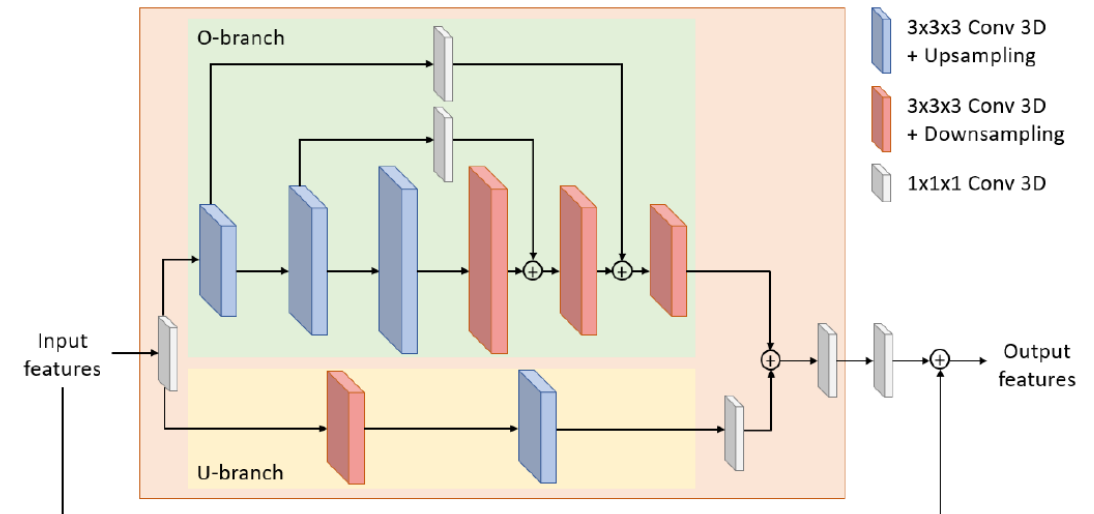
Overcomplete Representations Against Adversarial Videos (OUDefend)

- **Undercomplete** representations have large receptive fields to collect global information, but they overlook local details.

- **Overcomplete** representations have opposite properties.

- OUDefend balances **global** and **local** features by learning those two representations.

Shao-Yuan Lo, Jeya Maria Jose Valanarasu, and Vishal M. Patel. Overcomplete Representations Against Adversarial Videos. ICIP 2021.
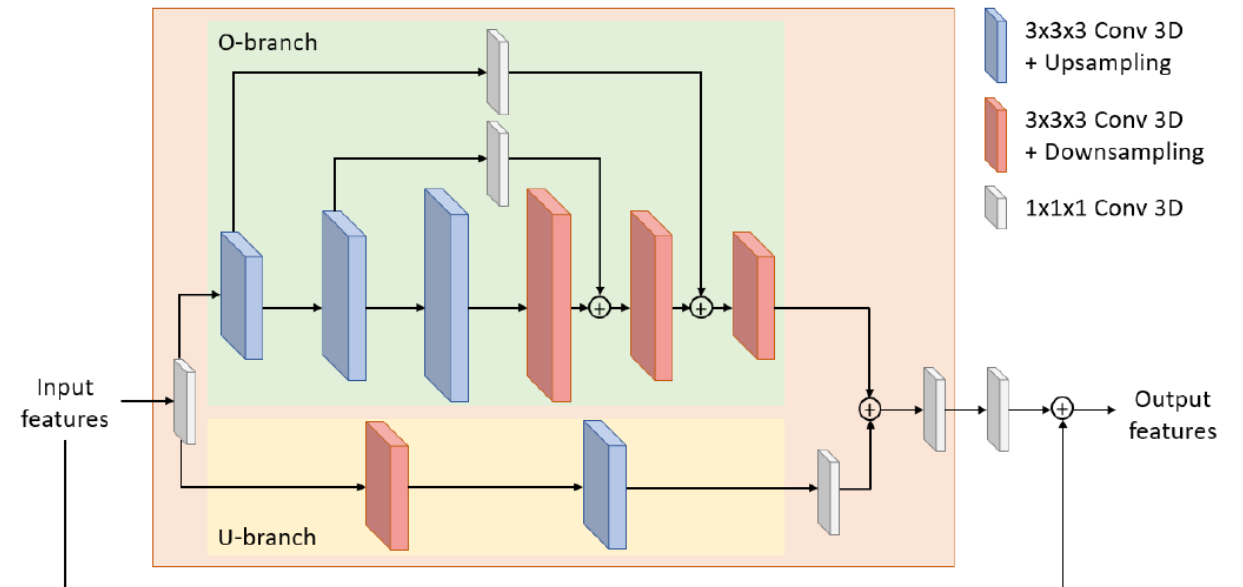
Overcomplete Representations Against Adversarial Videos (OUDefend)

- Append OUDefend blocks to the target network (after each res block).
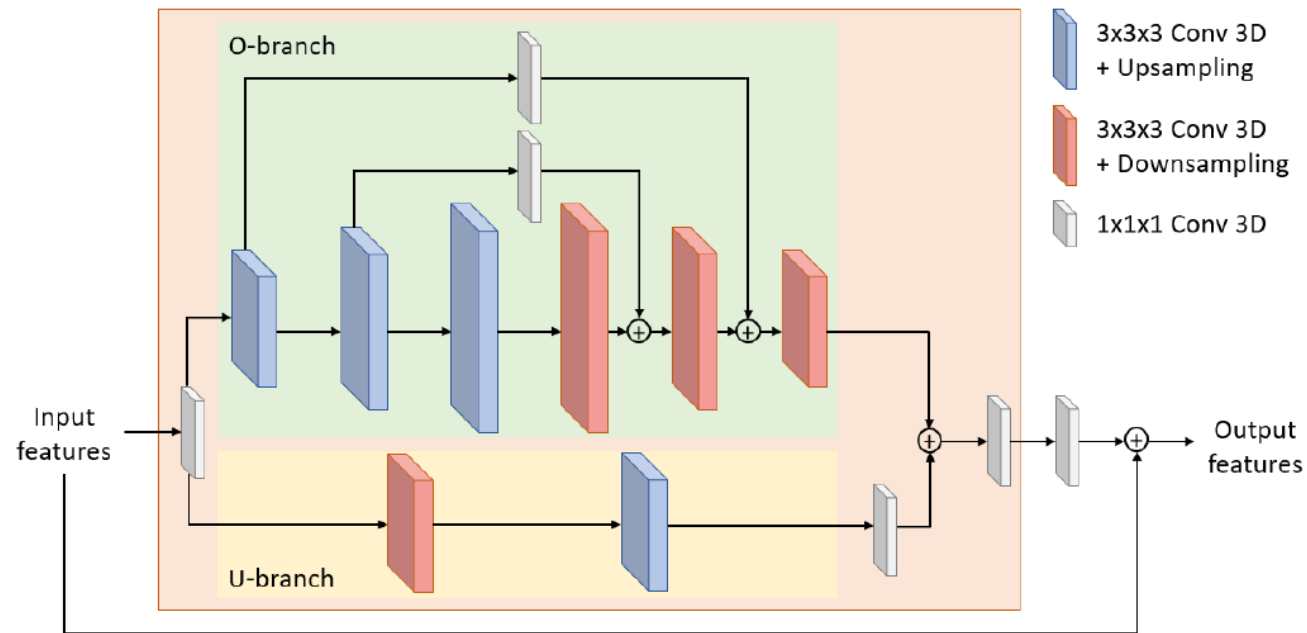
layer name	output size	18-layer
conv1	112×112	
conv2_x	56×56	$\begin{bmatrix} 3{\times}3,\ 64 \\ 3{\times}3,\ 64 \end{bmatrix} \times 2$
conv3_x	28×28	$\begin{bmatrix} 3{\times}3,\ 128 \\ 3{\times}3,\ 128 \end{bmatrix} \times 2$
conv4_x	14×14	$\begin{bmatrix} 3{\times}3,\ 256 \\ 3{\times}3,\ 256 \end{bmatrix} \times 2$
conv5_x	7×7	$\begin{bmatrix} 3{\times}3,\ 512 \\ 3{\times}3,\ 512 \end{bmatrix} \times 2$

Shao-Yuan Lo, Jeya Maria Jose Valanarasu, and Vishal M. Patel. Overcomplete Representations Against Adversarial Videos. ICIP 2021.

Overcomplete Representations Against Adversarial Videos (OUDefend)

Dataset: UCF-101

- No Defense: Original network trained on clean data

- Madry [Madry et al. ICLR'18] : Original network trained by adversarial training (AT)

- Xie-A [Xie et al. CVPR'19]: Feature denoising (3D conv) network with AT

- Xie-B [Xie et al. CVPR'19]: Feature denoising (2D conv frame-by-frame) network with AT

- OUDefend: Proposed OUDefend network with AT

Method	#Params	Clean	PGD Linf	PGD L2	MultAV	ROA	AF	SPA	Avg_adv
No Defense	33.0M	76.90	2.56	3.25	7.19	0.16	0.24	4.39	2.97
Madry	33.0M	76.90	33.94	35.05	47.00	41.29	55.99	55.99	48.01
Xie-A	33.7M	70.82	31.48	33.25	42.69	37.59	58.87	49.14	42.17
Xie-B	34.8M	69.47	30.19	32.65	41.87	38.22	58.74	49.14	41.80
OUDefend	33.6M	**77.90**	**34.18**	**35.32**	**47.63**	**42.00**	**56.25**	**56.29**	**49.52**

Shao-Yuan Lo, Jeya Maria Jose Valanarasu, and Vishal M. Patel. Overcomplete Representations Against Adversarial Videos. ICIP 2021.

Error Diffusion Halftoning Against Adversarial Examples

- Quantize each pixel in the raster order one-by-one, and spread the quantization error to the neighboring pixels.

Input image (I)   Halftone (Q)
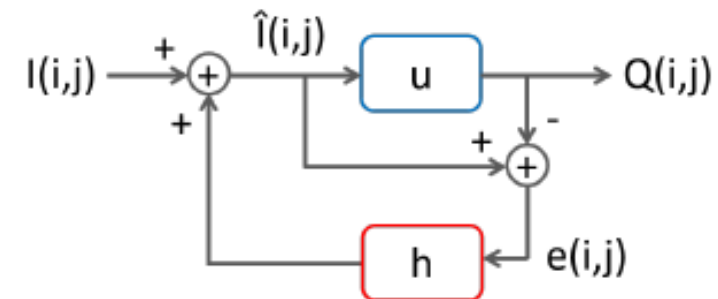

Floyd and Steinberg. Proceedings of the Society of Information Display, 1976.

$$\hat{I}(i,j) = I(i,j) + \sum_{m,n \in S} h(m,n)e(i-m,j-n)$$

$$Q(i,j) = u(\hat{I}(i,j) - \theta) \quad e(i,j) = \hat{I}(i,j) - Q(i,j)$$

u: unit step function   h: error filter

Shao-Yuan Lo and Vishal M. Patel. Error Diffusion Halftoning Against Adversarial Examples. ICIP 2021.   54

Error Diffusion Halftoning Against Adversarial Examples

- The **quantization operation** invalid the adversarial variations.

- **Updating the values of the neighboring pixels repeatedly** makes the adaptive attacks hard to identify the mapping between the original image and the corresponding halftone.

- **Spreading quantization errors produces** better halftoning quality and tends to enhance edges and object boundary in an image.

- Take **both** adversarial robustness and clean data performance.

- Complementary to adversarial training.

Error Diffusion Halftoning Against Adversarial Examples

- Dataset: CIFAR-10

- Attacks (white-box): PGD [Madry et al.] and Mult [Lo and Patel]

Method	Training	Clean	PGD-ℓ_∞	PGD-ℓ_2	Mult-ℓ_∞	Mult-ℓ_2	Avg$_{adv}$	Avg$_{all}$
Vanilla	Standard training	**94.03**	0.01	0.20	0.05	0.01	0.07	18.86
Gaussian blur		90.17	0.20	1.34	0.17	0.05	0.44	18.39
Non-local means		88.66	0.02	0.49	0.03	0.00	0.14	17.84
JPEG compression		90.06	2.97	4.82	1.81	0.22	2.46	19.98
Bit-depth reduction		78.87	**15.26**	10.84	**10.79**	**4.52**	**10.35**	**24.06**
Halftoning (ours)		88.57	9.53	**11.98**	5.54	1.07	7.03	23.34
Vanilla	Adversarial training	83.31	51.15	50.68	54.10	40.29	49.06	55.91
Gaussian blur		75.96	44.59	47.12	45.07	32.48	42.32	49.04
Non-local means		75.47	44.67	45.29	16.59	14.53	30.27	39.31
JPEG compression		24.97	38.99	43.72	59.15	44.72	46.65	42.31
Bit-depth reduction		71.66	47.34	42.40	48.50	41.63	44.97	50.31
Halftoning (ours)		**84.37**	**60.01**	**56.56**	**67.37**	**88.44**	**68.10**	**71.35**

Shao-Yuan Lo and Vishal M. Patel. Error Diffusion Halftoning Against Adversarial Examples. ICIP 2021.

Error Diffusion Halftoning Against Adversarial Examples

- Mean square differences between the features of clean images and the features of adversarial examples.

Shao-Yuan Lo and Vishal M. Patel. Error Diffusion Halftoning Against Adversarial Examples. ICIP 2021.
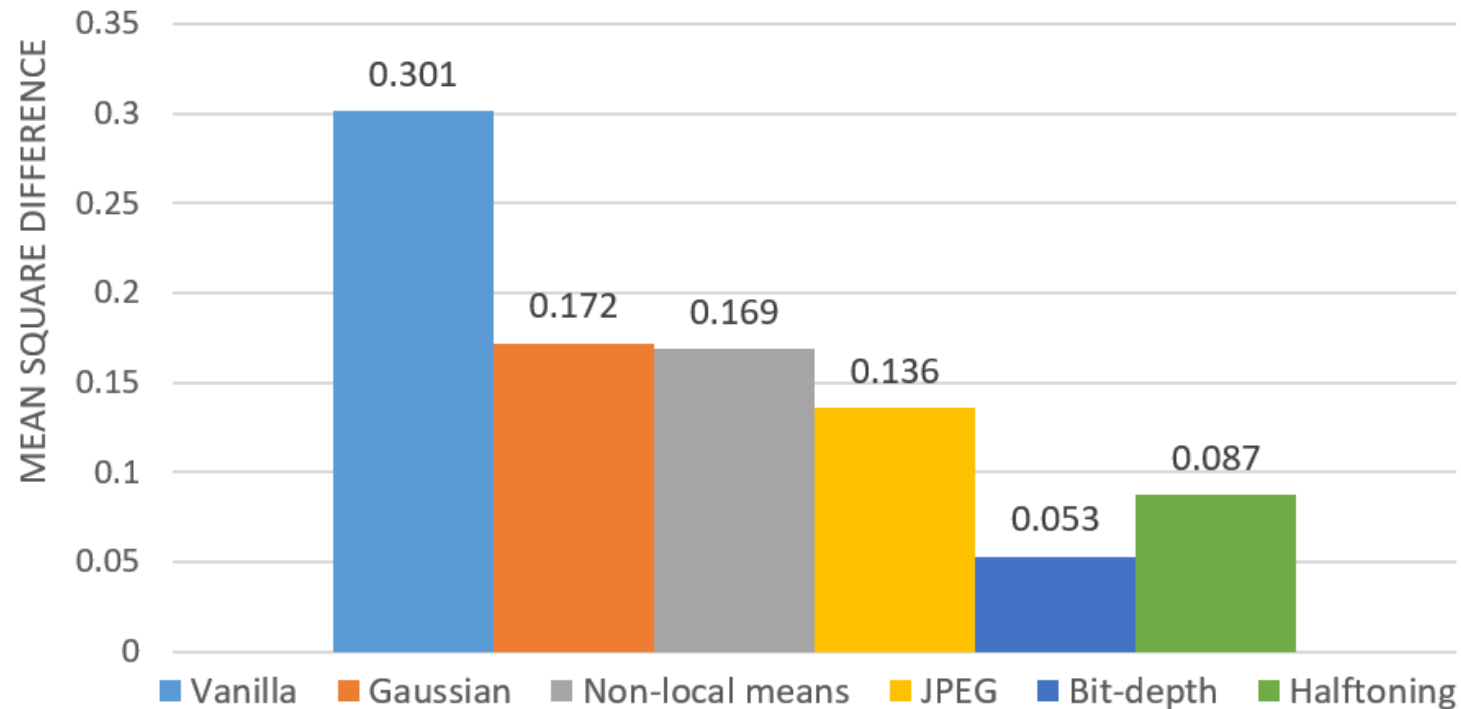
Multiplicative Adversarial Videos (MultAV)

- Additive:

$$\mathbf{x}^{t+1} = Clip_{\mathbf{x},\epsilon}^{\ell_\infty}\{\mathbf{x}^t + \alpha \cdot sign(\nabla_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t, \mathbf{y}; \boldsymbol{\theta}))\}$$

$$\mathbf{x}^{t+1} = Clip_{\mathbf{x},\epsilon}^{\ell_2}\{\mathbf{x}^t + \alpha \cdot \frac{\nabla_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t, \mathbf{y}; \boldsymbol{\theta})}{\|\nabla_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t, \mathbf{y}; \boldsymbol{\theta})\|_2}\}$$

- Multiplicative:

$$\mathbf{x}^{t+1} = Clip_{\mathbf{x},\epsilon_m}^{RB-\ell_\infty}\{\mathbf{x}^t \odot \alpha_m^{sign(\nabla_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t, \mathbf{y}; \boldsymbol{\theta}))}\}$$
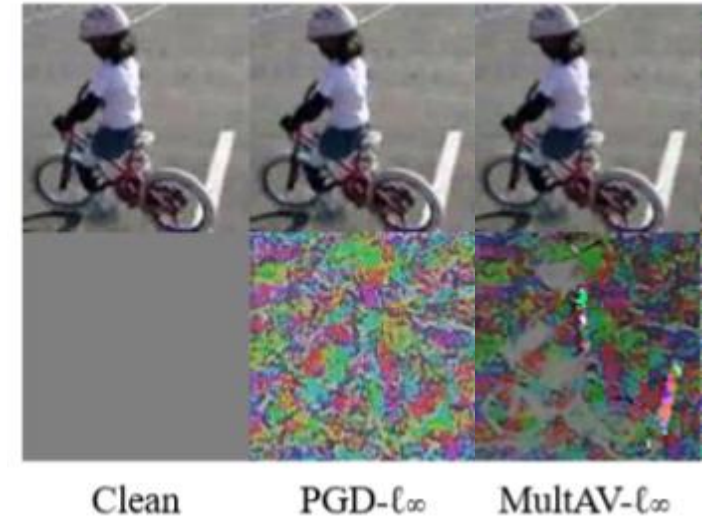
$$\mathbf{x}^{t+1} = Clip_{\mathbf{x},\epsilon_m}^{RB-\ell_2}\{\mathbf{x}^t \odot \alpha_m^{\frac{\nabla_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t, \mathbf{y}; \boldsymbol{\theta})}{\|\nabla_{\mathbf{x}^t}\mathcal{L}(\mathbf{x}^t, \mathbf{y}; \boldsymbol{\theta})\|_2}}\}$$

Shao-Yuan Lo and Vishal M. Patel. MultAV: Multiplicative Adversarial Videos. AVSS 2021.

Multiplicative Adversarial Videos (MultAV)

Task: Action recognition
Dataset: UCF-101

Network	Clean
3D ResNet-18	76.90

MultAV-ℓ_∞	MultAV-ℓ_2	MultAV-ROA	MultAV-AF	MultAV-SPA
7.19	2.67	2.30	0.26	4.02

Clean   PGD-ℓ_∞ MultAV-ℓ_∞

Shao-Yuan Lo and Vishal M. Patel. MultAV: Multiplicative Adversarial Videos. AVSS 2021.

Summary

- Adversarial examples cause serious safety concerns.

- Adversarial robustness is a rising research topic.

- Our works
  - MultiBN (T-IP 2022)
  - OUDefend (ICIP 2021)
  - Halftone (ICIP 2021)
  - MultAV (AVSS 2021)

Acknowledgement

Vishal M. Patel

Jeya Maria Jose Valanarasu

Poojan Oza

Thanks for your attention