Adversarial attacks on deep learning for computer vision

Ajmal Saeed Mian
Professor
http://ajmalsaeed.net/

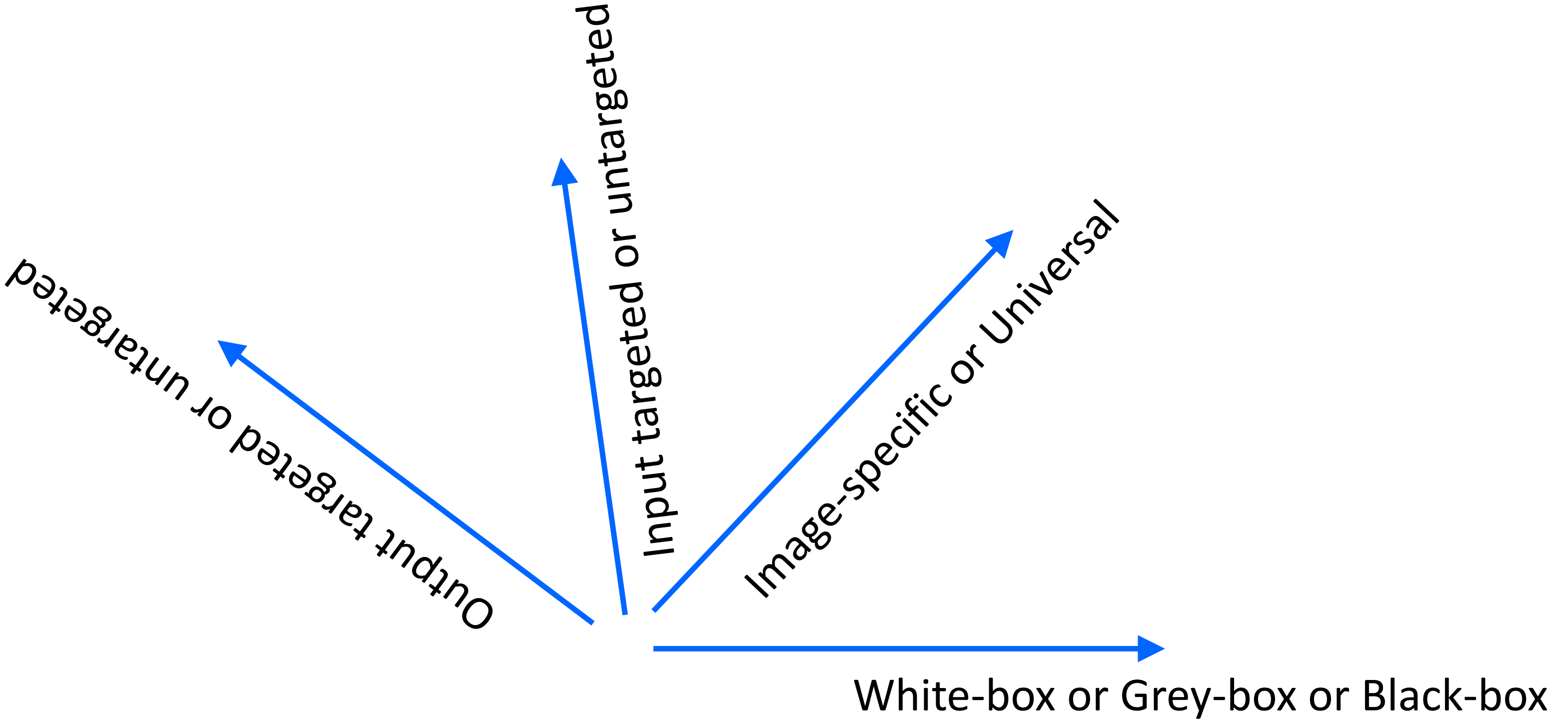
THE UNIVERSITY OF WESTERN AUSTRALIA

Overview

- Introduction to adversarial attacks and defenses
- Defense against Universal Adversarial Perturbations
- Label Universal Targeted Attack (LUTA)
- Attack to explaining deep networks
- Spatio temporal attack on joints based human action recognition

Types of Attacks

An attack on a ML algorithm is defined as modification in the input data that changes its decision.

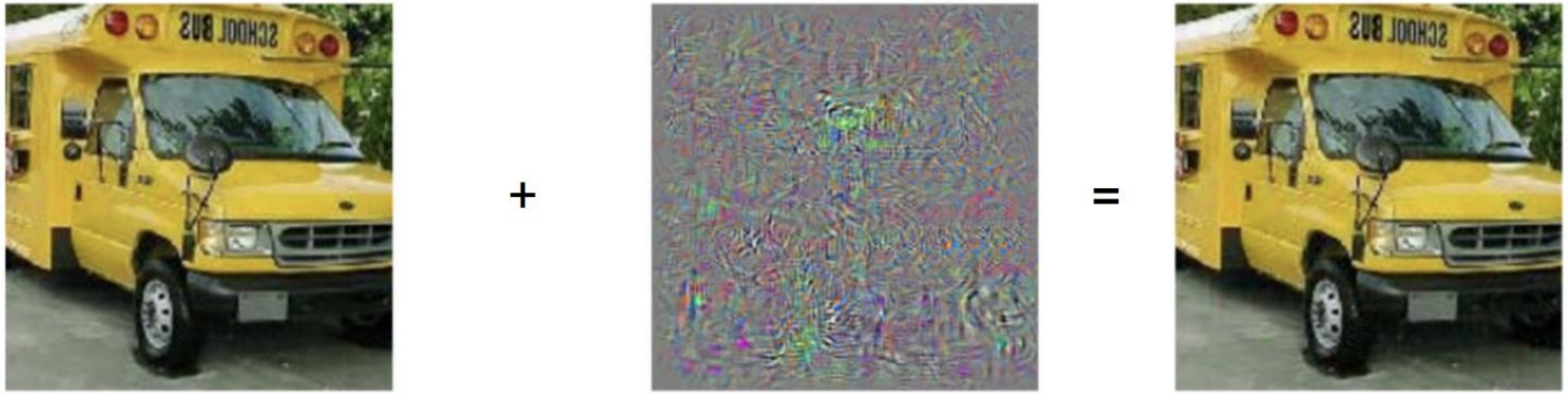


Naveed Akhtar and Ajmal Mian, *"Threat of Adversarial Attacks on Deep Learning for Computer Vision: A Survey"*, IEEE Access, 8 Feb 2018.

Adversarial Attacks

- Small perturbations that appear harmless to the human eye
- Cause state-of-the-art DNNs to misclassify an image

Bus Ostrich



$\mathbf{I}_c$ $\boldsymbol{\rho}$

$$\min_{\boldsymbol{\rho}} ||\boldsymbol{\rho}||_2 \ \text{ s.t. } \mathcal{C}(\mathbf{I}_c + \boldsymbol{\rho}) = \ell; \ \mathbf{I}_c + \boldsymbol{\rho} \in [0,1]^m$$

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, “Intriguing properties of neural networks”, arXiv:1312.6199v4, 2013.

Attacks in the Real World

T. Gu, B. Dolan-Gavitt, S. Garg, “Badnets: Identifying vulnerabilities in the machine learning model supply chain”, arXiv:1708.06733v2 2017.

Attacks in the Real World

S. Thys, W. Van Ranst, T. Goedemé, “Fooling automated surveillance cameras: adversarial patches to attack person detection”, CVPR Workshop 2019

Fooling Face Recognition

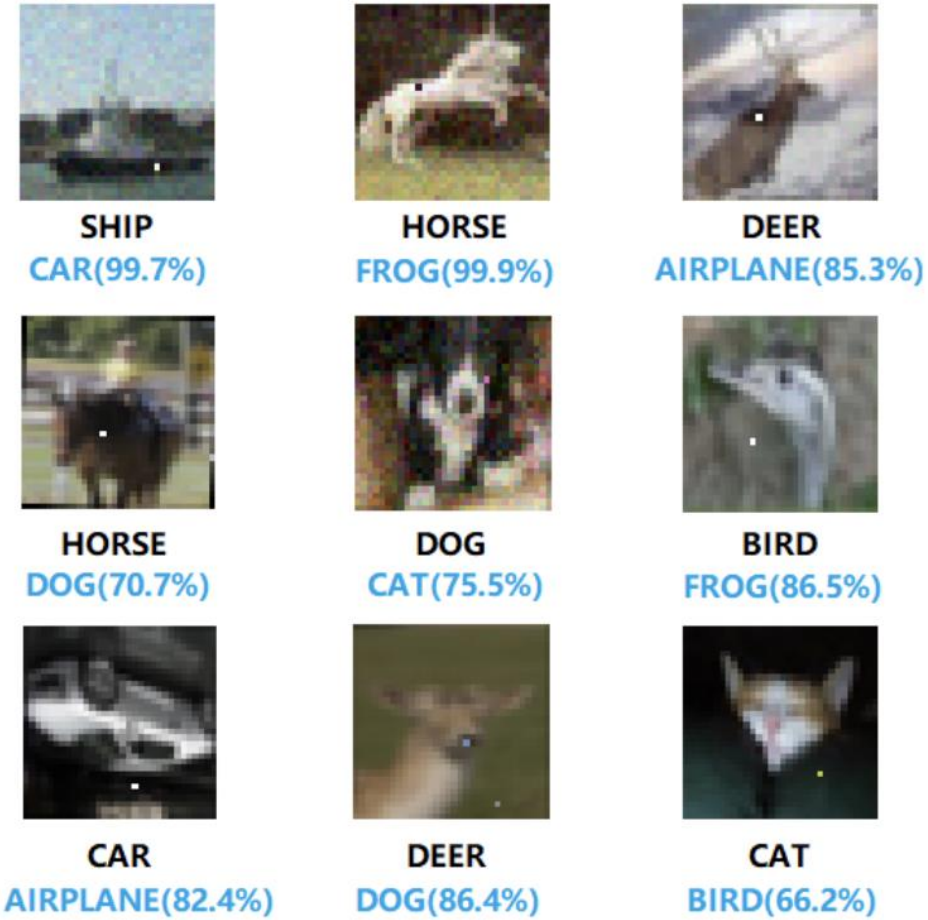
(a) (b) (c) (d)

M. Sharif, S. Bhagavatula, L. Bauer, M. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition", ACM SIGSAC Conference on Computer and Communications, 2016.

One Pixel Attacks

CIFAR10

67% of test images fooled



IMAGENET

Only 16% test images fooled

Jiawei Su, Danilo Vasconcellos Vargas, Sakurai Kouichi, “One pixel attack for fooling deep neural networks” arXiv:1710.08864v7, 2017

One Character Attack in NLP

Against **BERT** for sentiment, 1-char attack send error from 90.3% → 45.8%.

Alteration	**Movie Review**	**Label**
Original	A triumph, relentless and beautiful in its downbeat darkness	+
Swap	A triumph, relentless and beuatiful in its downbeat darkness	–
Drop	A triumph, relentless and beautiful in its dwnbeat darkness	–

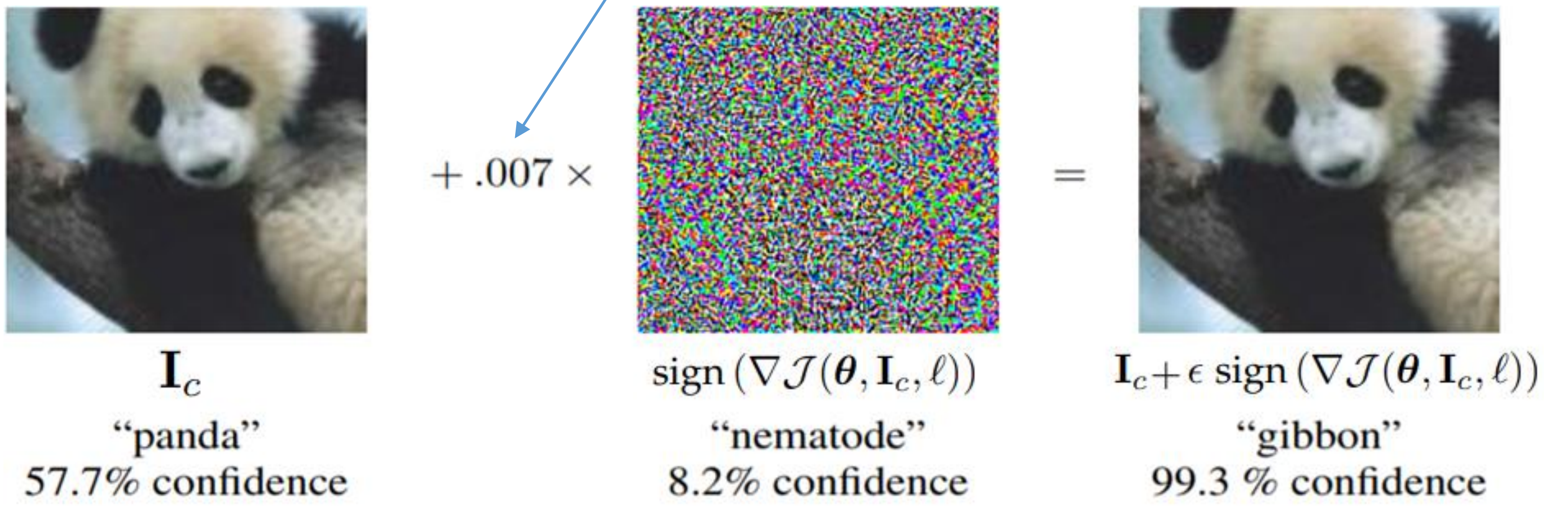
D. Pruthi, B. Dhingra, Z. Lipton. "Combating Adversarial Misspellings with Robust Word Recognition", ACL 2019.

Imperceptibility Constraint

- Perturbations to the input generally have imperceptibility constraint
- ℓ_0 constraint i.e. only perturb a few pixels (glasses, patch)
- ℓ_2 constraint (projection on ℓ_2 ball)
- ℓ_∞ constraint (projection on ℓ_∞ ball)

Fast Gradient Sign Method (FGSM)

- A more efficient method $\boldsymbol{\rho} = \epsilon \ \text{sign}\left(\nabla \mathcal{J}(\boldsymbol{\theta}, \mathbf{I}_c, \ell)\right)$



Iterative FGSM (I-FGSM)
Momentum I-FGSM (MI-FGSM)
Diverse Input I-FGSM (DI^2-FGSM)
M-DI^2-FGSM)

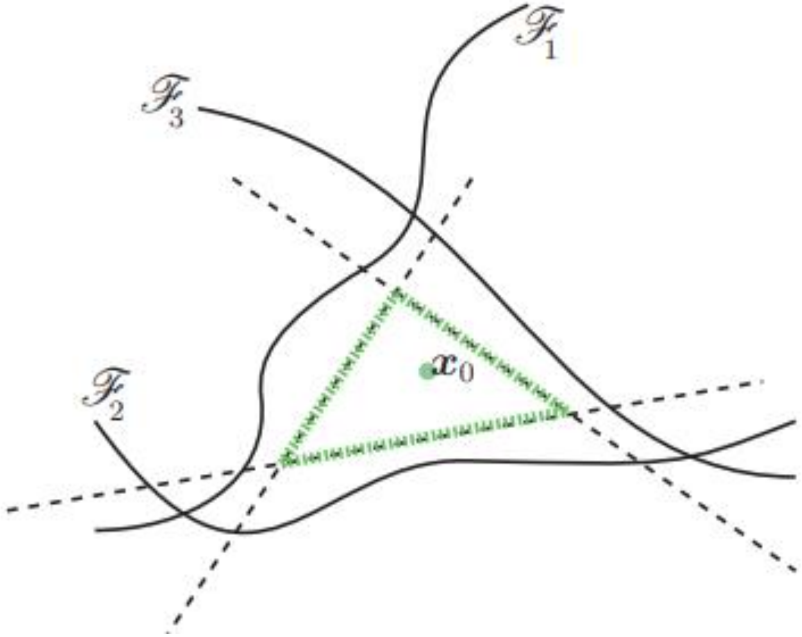
$$\mathbf{I}_{\boldsymbol{\rho}}^{i+1} = \text{Clip}_{\epsilon}\left\{\mathbf{I}_{\boldsymbol{\rho}}^{i} + \alpha \ \text{sign}(\nabla \mathcal{J}(\boldsymbol{\theta}, \mathbf{I}_{\boldsymbol{\rho}}^{i}, \ell)\right\}$$

I. Goodfellow, J. Shlens, C. Szegedy, "Explaining and harnessing adversarial examples", arXiv:1412.6572v3, 2014

BIM, ILCM, PGD, C&W

- Basic Iterative Method (BIM) is basically similar to I-FGSM
- Iterative Least-Likely Class Method (ILCM) sets the target class as the least likely one
- Projected Gradient Descend (PGD) is very famous powerful attack. It is similar to an ℓ_∞ bounded I-FGSM but the authors show more advantages such as its use for robust training without 'label-leaking'
- Carlini & Wagner (C&W) define a set of optimization functions that completely break the defensive distillation. This attack is powerful but computationally expensive

DeepFool



Algorithm 2 DeepFool: multi-class case

input: Image $\boldsymbol{x}$, classifier f.
output: Perturbation $\hat{\boldsymbol{r}}$.

Initialize $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$, $i \leftarrow 0$.
while $\hat{k}(\boldsymbol{x}_i) = \hat{k}(\boldsymbol{x}_0)$ **do**
  for $k \neq \hat{k}(\boldsymbol{x}_0)$ **do**
    $\boldsymbol{w}'_k \leftarrow \nabla f_k(\boldsymbol{x}_i) - \nabla f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)$
    $f'_k \leftarrow f_k(\boldsymbol{x}_i) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)$
  end for
  $\hat{l} \leftarrow \arg\min_{k \neq \hat{k}(\boldsymbol{x}_0)} \frac{|f'_k|}{\|\boldsymbol{w}'_k\|_2}$
  $\boldsymbol{r}_i \leftarrow \frac{|f'_{\hat{l}}|}{\|\boldsymbol{w}'_{\hat{l}}\|_2^2} \boldsymbol{w}'_{\hat{l}}$
  $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \boldsymbol{r}_i$
  $i \leftarrow i + 1$
end while
return $\hat{\boldsymbol{r}} = \sum_i \boldsymbol{r}_i$

- Decision boundaries are approximated by a polyhedron
- At each iteration, the vector that reaches the polyhedron boundary is computed and added to the current estimate

SM Moosavi-Dezfooli, A Fawzi, P Frossard, “DeepFool: a simple and accurate method to fool deep neural networks”, CVPR 2016.

Black Box Attacks

1. Query based attacks (aka Decision based attacks)
   - Query the target model
   - Inspect the decision (or output probabilities)
   - Change perturbation in the image accordingly

2. Transfer based attacks
   - Use a surrogate (substitute) model to learn perturbations in white-box setting
   - Perturbations transfer well especially if the training data is known

Universal Adversarial Perturbations

A single perturbation to fool a network on *any image* with a high probability (e.g. 0.8+).

Perturbations *generalize well* across different models, posing a threat to Deep Learning in practice.

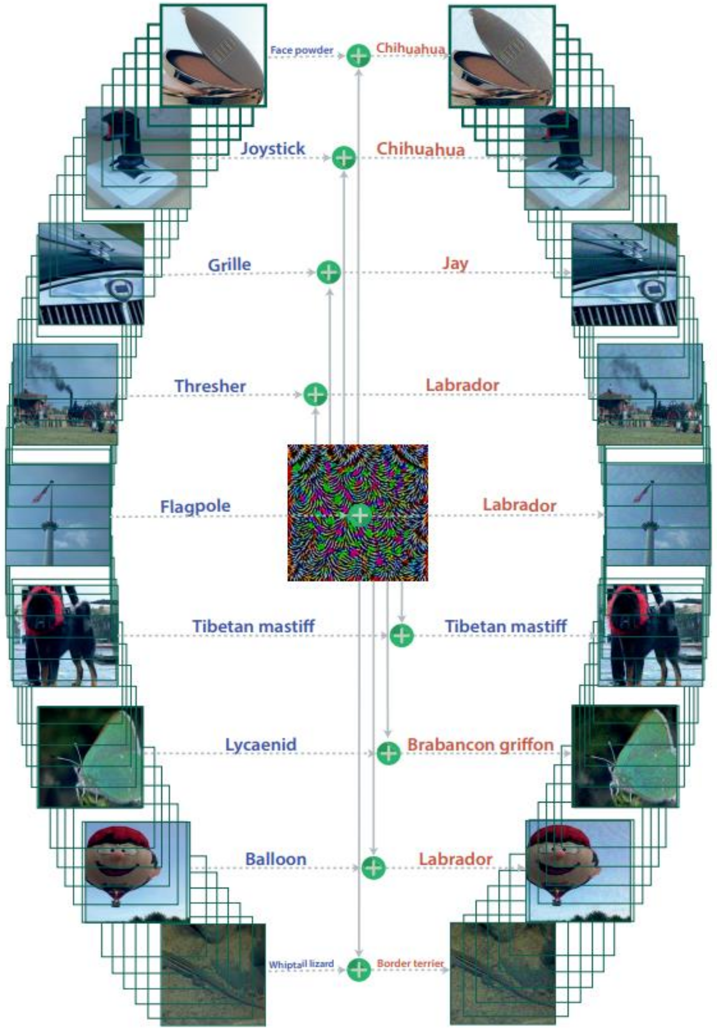


Image from [7]

S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal Adversarial Perturbations”, CVPR 2017.

Universal Adversarial Perturbations

Objective

$$\|v\|_p \leq \xi$$

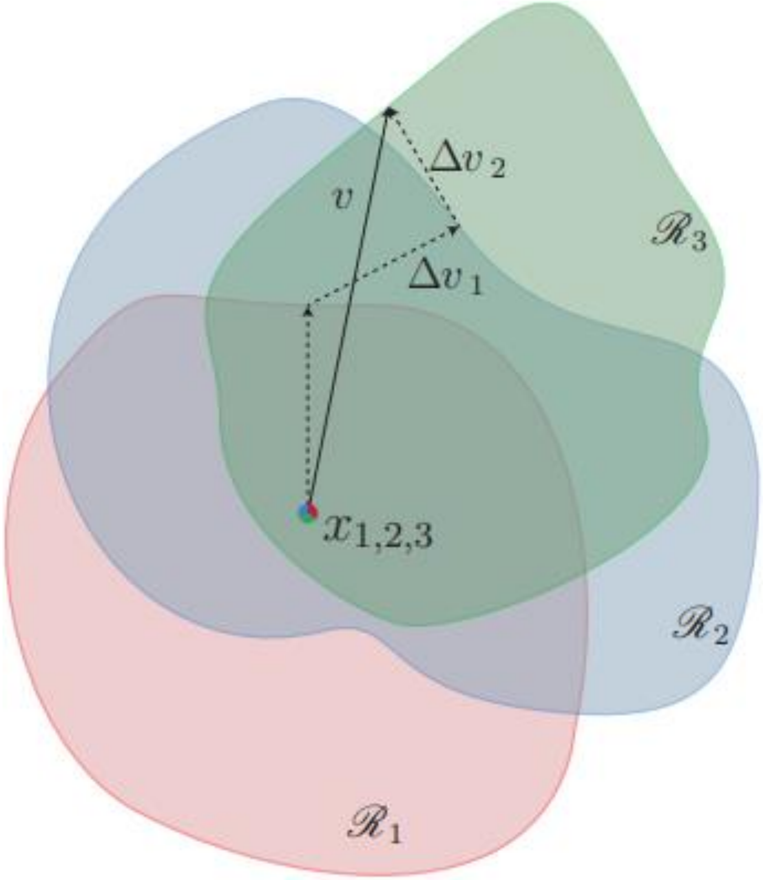
$$\mathbb{P}_{x \sim \mu}\left(\hat{k}(x+v) \neq \hat{k}(x)\right) \geq 1-\delta$$

v : desired perturbation

ξ : perturbation norm threshold

δ : fooling ratio

Universal Adversarial Perturbations



Graphical Illustration

Deepfool or FGSM or I-FGSM

Algorithm 1 Computation of universal perturbations.

input: Data points X, classifier $\hat{k}$, desired ℓ_p norm of the perturbation ξ, desired accuracy on perturbed samples δ.
output: Universal perturbation vector v.
Initialize $v \leftarrow 0$.
while $\mathrm{Err}(X_v) \leq 1 - \delta$ **do**
  for each datapoint $x_i \in X$ **do**
    if $\hat{k}(x_i + v) = \hat{k}(x_i)$ **then**
      Compute the *minimal* perturbation that sends $x_i + v$ to the decision boundary:

$$\Delta v_i \leftarrow \arg\min_r \|r\|_2 \text{ s.t. } \hat{k}(x_i + v + r) \neq \hat{k}(x_i).$$

      Update the perturbation:

$$v \leftarrow \mathcal{P}_{p,\xi}(v + \Delta v_i).$$

    end if
  end for
end while

“Defense against universal adversarial perturbations”,

Naveed Akhtar, Jian Liu, and Ajmal Mian, CVPR 2018.

Defense Against Universal Perturbations

$\mathfrak{S}_c \in \mathbb{R}^{\mathrm{d}}$: distribution of clean images

$\mathbf{I}_c \sim \mathfrak{S}_c$: a clean image is a sample

$\mathcal{C}(\mathbf{I}_c) : \mathbf{I}_c \rightarrow \ell \in \mathbb{R}$: deep model maps image to a class label

$\boldsymbol{\rho} \in \mathbb{R}^{\mathrm{d}}$: is a universal perturbation, if

$$\underset{\mathbf{I}_c \sim \mathfrak{S}_c}{\mathrm{P}}\Big(\mathcal{C}(\mathbf{I}_c) \neq \mathcal{C}(\mathbf{I}_c + \boldsymbol{\rho})\Big) \geq \delta \quad \text{s.t.} \quad ||\boldsymbol{\rho}||_p \leq \xi \longleftarrow \quad 2{,}000 \text{ for } l_2$$

Fooling ratio Lp-norm

0.8 l_∞, l_2

Naveed Akhtar, Jian Liu, and Ajmal Mian, “Defense against universal adversarial perturbations”, CVPR 2018.

Defense Against Universal Perturbations

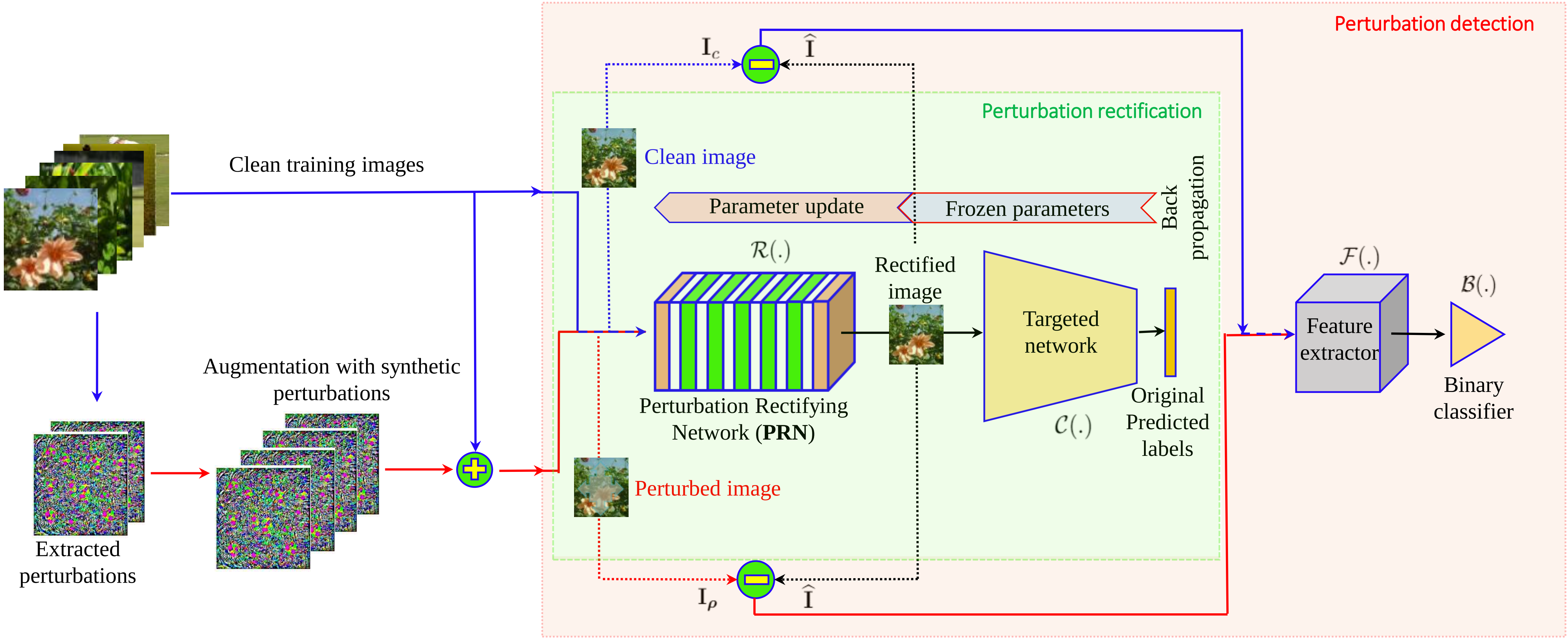
We seek

1) A detector: $\mathcal{D}(\mathbf{I}_{\boldsymbol{\rho}/c}) \; : \; \mathbf{I}_{\boldsymbol{\rho}/c} \rightarrow [0, 1]$

2) A rectifier: $\mathcal{R}(\mathbf{I}_{\boldsymbol{\rho}}) \; : \; \mathbf{I}_{\boldsymbol{\rho}} \rightarrow \widehat{\mathbf{I}}, \quad \underset{\mathbf{I}_c \sim \mathfrak{S}_c}{\mathrm{P}} \Big(\mathcal{C}(\widehat{\mathbf{I}}) = \mathcal{C}(\mathbf{I}_c) \Big) \approx 1$

Naveed Akhtar, Jian Liu, and Ajmal Mian, “Defense against universal adversarial perturbations”, CVPR 2018.

Defense Against Universal Perturbations



rectifier: $\mathcal{R}(.)$ detector: $\mathcal{B}(\mathcal{F}(\mathbf{I}_{\rho/c} - \mathcal{R}(\mathbf{I}_{\rho/c})))$

Naveed Akhtar, Jian Liu, and Ajmal Mian, "Defense against universal adversarial perturbations", CVPR 2018.

Synthetic Perturbation Generation

- Allows better training of PRN
- Search the positive orthant of the subspace spanned by original perturbations while satisfying norm constraints

Algorithm 1 ℓ_∞-norm synthetic perturbation generation

Input: Pre-generated perturbation samples $\mathcal{P} \subseteq \mathbb{R}^{\mathrm{d}}$, number of new samples to be generated η, threshold ξ.

Output: Synthetic perturbations $\mathcal{P}_s \subseteq \mathbb{R}^d$

```
set P_s = {}; ℓ2-threshold = E[{||ρ_{i∈P}||_2}_{i=1}^{|P|}];
P_n = P with ℓ2-normalized elements.
while |P_s| < η do
    set ρ_s = 0
    while ||ρ_s||_∞ < ξ do
        z ~ unif(0, 1) ⊙ ξ
        ρ_s = ρ_s + (z ⊙ ~rand P_n)
    end while
    if ||ρ_s||_2 ≥ ℓ2-threshold then
        P_s = P_s ∪ ρ_s
    end if
end while
return
```

set $\mathcal{P}_s = \{\}$; ℓ_2-threshold $= \mathbb{E}\left[\{||\boldsymbol{\rho}_{i\in\mathcal{P}}||_2\}_{i=1}^{|\mathcal{P}|}\right]$; $\mathcal{P}_n = \mathcal{P}$ with ℓ_2-normalized elements.

while $|\mathcal{P}_s| < \eta$ **do**

  set $\boldsymbol{\rho}_s = \mathbf{0}$

  while $||\boldsymbol{\rho}_s||_\infty < \xi$ **do**

    $z \sim \text{unif}(0, 1) \odot \xi$

    $\boldsymbol{\rho}_s = \boldsymbol{\rho}_s + (z \odot \overset{\text{rand}}{\sim} \mathcal{P}_n)$

  end while

  if $||\boldsymbol{\rho}_s||_2 \geq \ell_2$-threshold **then**

    $\mathcal{P}_s = \mathcal{P}_s \bigcup \boldsymbol{\rho}_s$

  end if

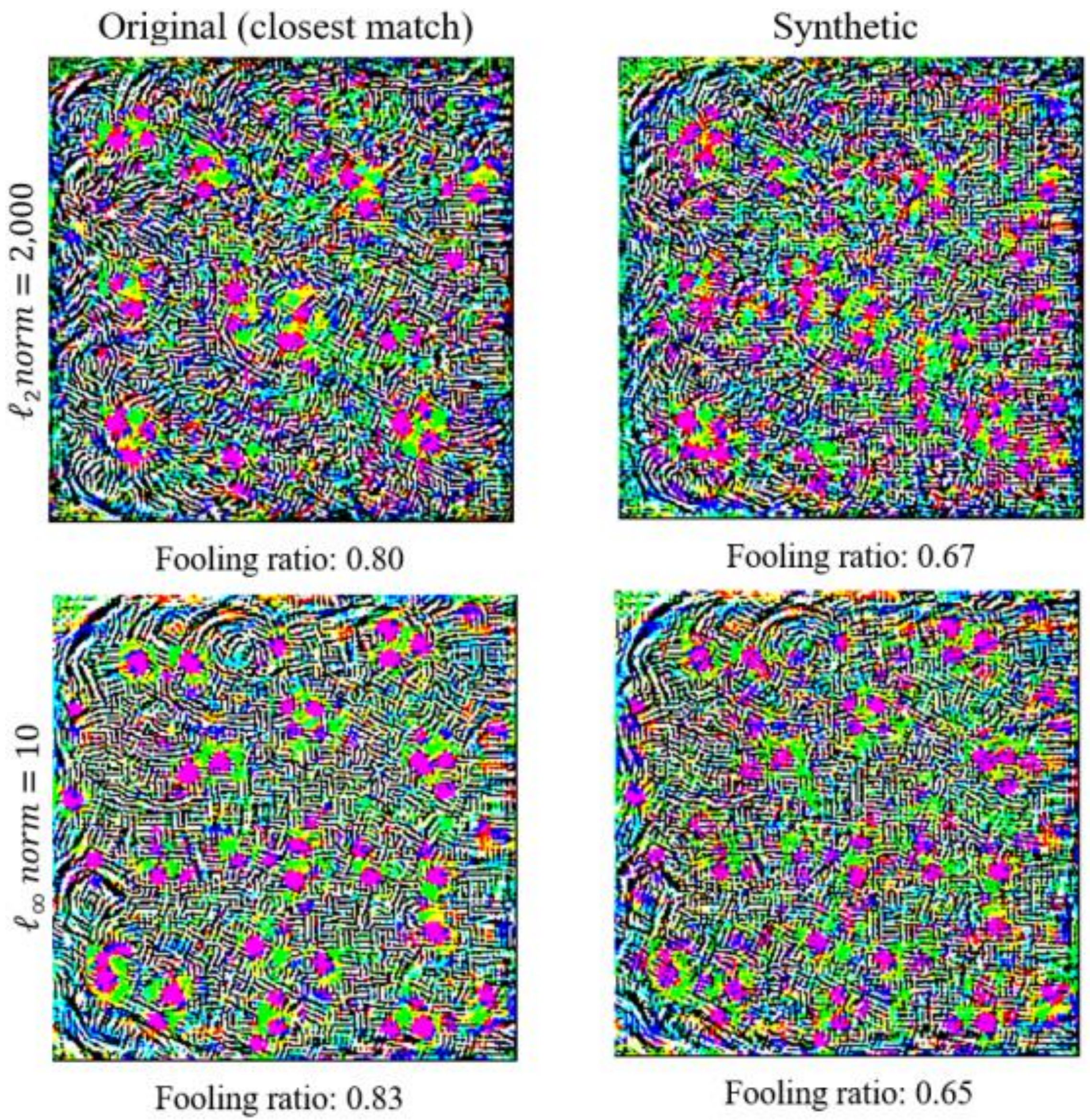
end while

return

Synthetic Perturbation Generation

- Allows better training of PRN

- Search the positive orthant of the subspace spanned by original perturbations while satisfying norm constraints



Evaluation

Cross-validation set of ImageNet [8], 40,000 Training, 10,000 testing

Defending CaffeNet [3], VGG-F network [9] and GoogLeNet [10]

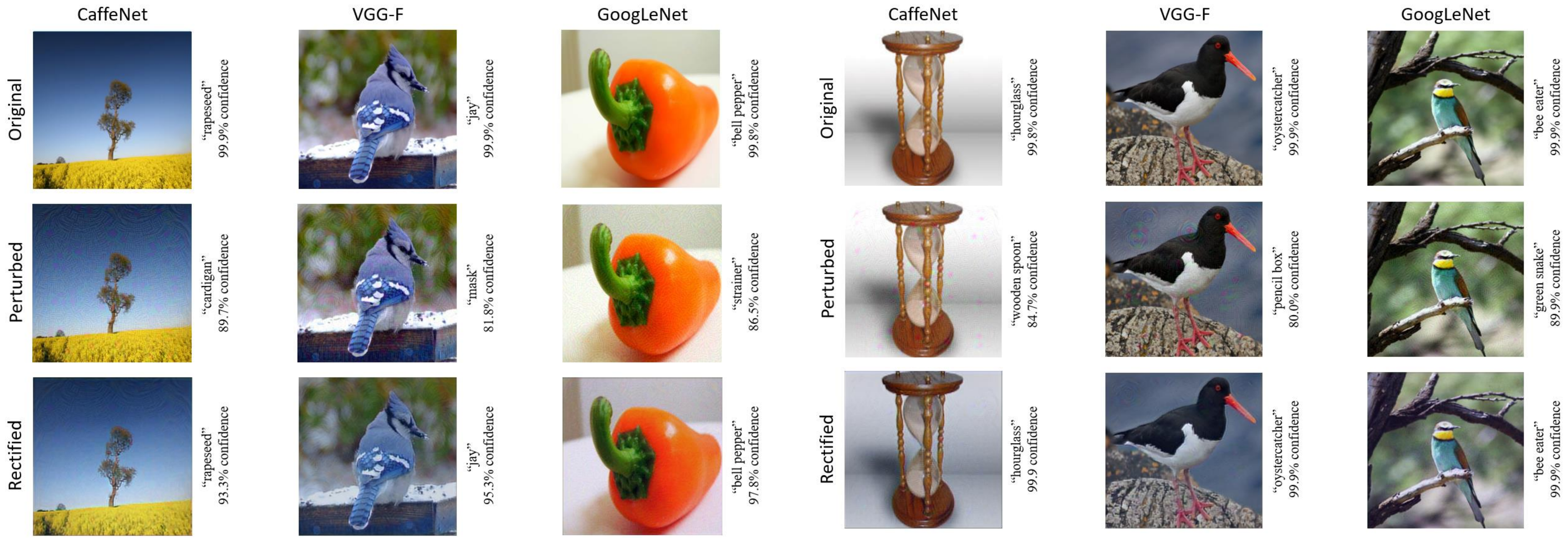
Protocol A- Use all 10,000 test samples

Protocol B- Use test samples correctly classified in clean form

Input images are perturbed with 0.5 probability

[3] Krishevsky et al, NIPS 2012. [8] Russakovsky et al, IJCV 2015. [9] Chatfield et al, arXiv.CS 2014.

Results



Naveed Akhtar, Jian Liu, and Ajmal Mian, "Defense against universal adversarial perturbations", CVPR 2018.

Same Network Defense

PRN-gain : Percentage improvement in accuracy on perturbed images

PRN-restoration : Percentage of restored accuracy on all images

Detection rate : Accuracy of detector

Defense rate : Percentage of restored accuracy on all images, incorporating detection

GoogLeNet

Metric	Same test/train perturbation type				Different test/train perturbation type			
	ℓ_2-type		ℓ_∞-type		ℓ_2-type		ℓ_∞-type	
	Prot-A	Prot-B	Prot-A	Prot-B	Prot-A	Prot-B	Prot-A	Prot-B
PRN-gain (%)	77.0	77.1	73.9	74.2	76.4	77.0	72.6	73.4
PRN-restoration (%)	97.0	92.4	95.6	91.3	97.1	92.7	93.8	89.3
Detection rate (%)	94.6	94.6	98.5	98.4	92.4	92.3	81.3	81.2
Defense rate (%)	97.4	94.8	96.4	93.7	97.5	94.9	94.3	91.6

Naveed Akhtar, Jian Liu, and Ajmal Mian, "Defense against universal adversarial perturbations", CVPR 2018.

Same Network Defense

CaffeNet

Metric	Same test/train perturbation type				Different test/train perturbation type			
	ℓ_2-type		ℓ_∞-type		ℓ_2-type		ℓ_∞-type	
	Prot-A	Prot-B	Prot-A	Prot-B	Prot-A	Prot-B	Prot-A	Prot-B
PRN-gain (%)	67.2	69.0	78.4	79.1	65.3	66.8	77.3	77.7
PRN-restoration (%)	95.1	89.9	93.6	88.7	92.2	87.1	91.7	85.8
Detection rate (%)	98.1	98.0	97.8	97.9	84.2	84.0	97.9	98.0
Defense rate (%)	96.4	93.6	95.2	92.5	93.6	90.1	93.2	90.0

VGG-F

Metric	Same test/train perturbation type				Different test/train perturbation type			
	ℓ_2-type		ℓ_∞-type		ℓ_2-type		ℓ_∞-type	
	Prot-A	Prot-B	Prot-A	Prot-B	Prot-A	Prot-B	Prot-A	Prot-B
PRN-gain (%)	72.1	73.3	84.1	84.3	68.3	69.2	84.7	84.8
PRN-restoration (%)	93.2	86.2	90.3	83.2	88.8	81.2	91.1	83.3
Detection rate (%)	92.5	92.5	98.6	98.6	92.5	92.5	98.1	98.1
Defense rate (%)	95.5	91.4	92.2	87.9	90.0	85.9	93.7	89.1

Naveed Akhtar, Jian Liu, and Ajmal Mian, “Defense against universal adversarial perturbations”, CVPR 2018.

Cross Network Defense

Protocol A

Defense rate (%) ℓ_2	VGG-F	CaffeNet	GoogLeNet
VGG-F [4]	95.5	91.5	82.4
CaffeNet [16]	94.8	96.2	77.3
GoogLeNet [37]	88.3	87.3	97.4

Defense rate (%) ℓ_∞	VGG-F	CaffeNet	GoogLeNet
VGG-F [4]	92.2	88.9	74.8
CaffeNet [16]	93.5	95.2	73.8
GoogLeNet [37]	88.4	85.4	96.4

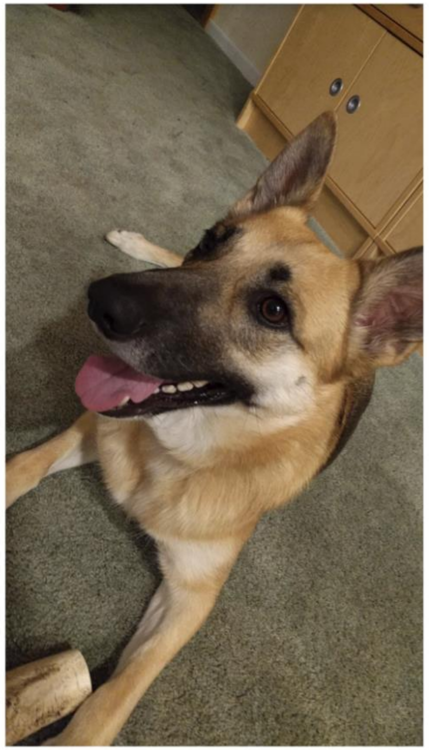
Non-diagonal entries are cross-network defense rates

Other Defense Techniques

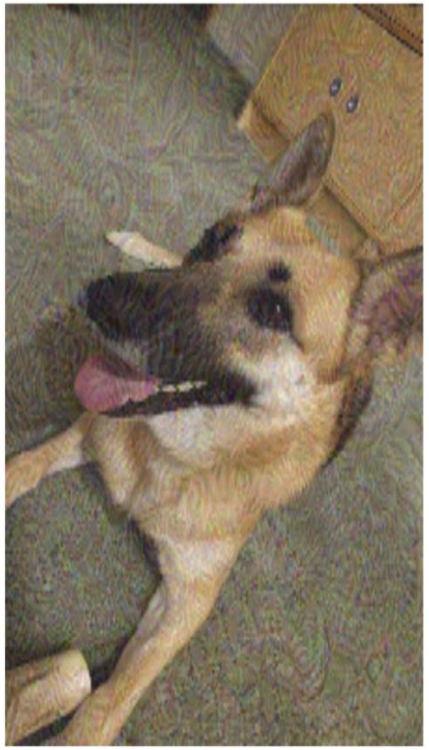
Adversarial Training

1. Generate Adversarial Examples

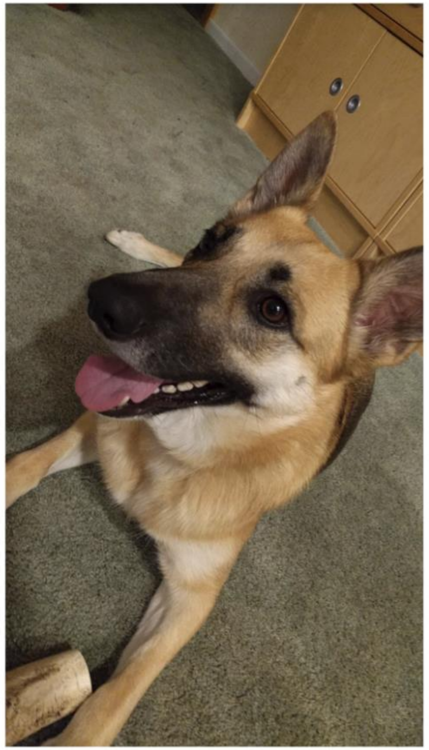
2. Add to Train Dataset

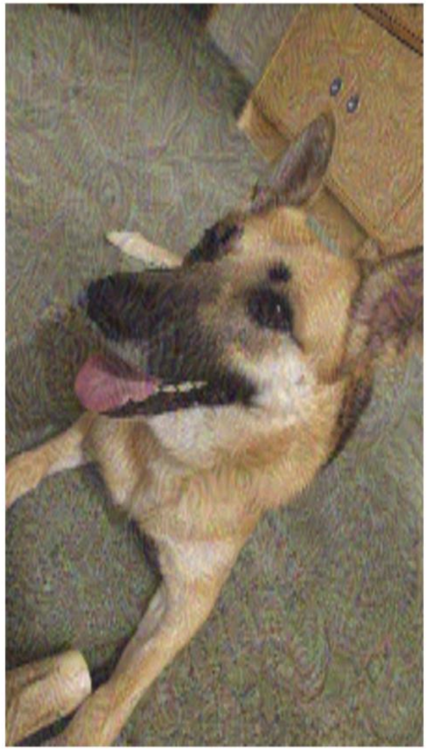


Dog



Frog



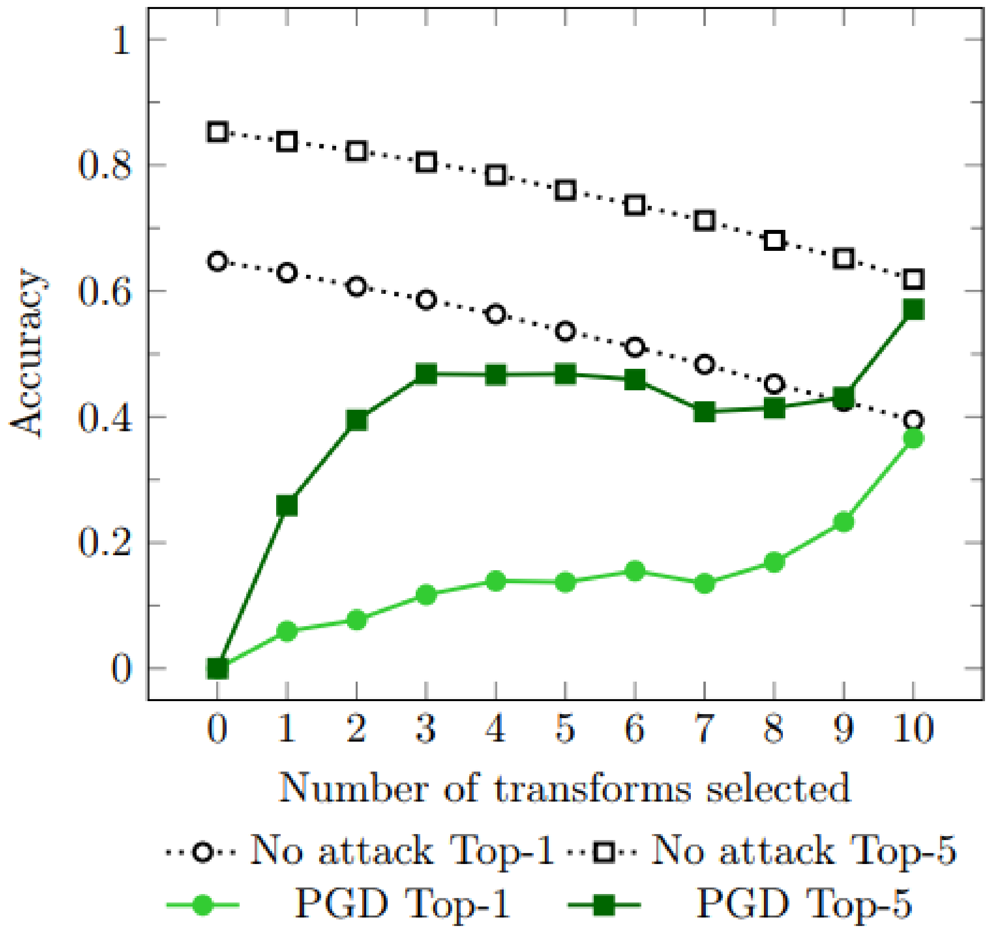


Dog

Goodfellow et al arXiv 2014

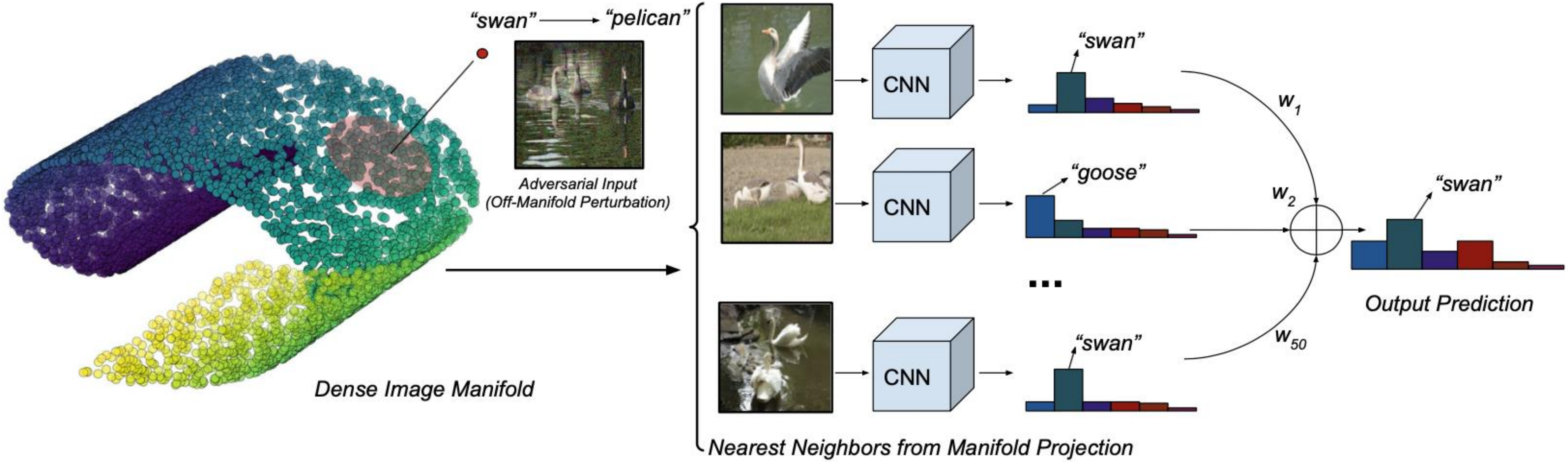
Barrage of Random Transforms

- Stochastic combination of weak defences
- Into a single barrage of randomized transformations
- To build a strong defence against adversarial attacks



Raff et al., "Barrage of Random Transforms for Adversarially Robust Defense", CVPR 2019.

Web-Scale Nearest Neighbor Search

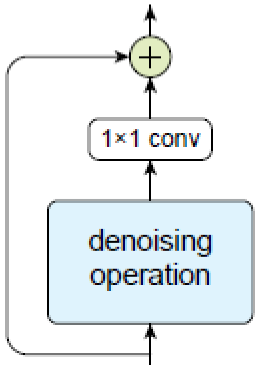


Dubey et al. "Defense against adversarial images using web-scale nearest-neighbor search", CVPR 2019.

Feature Denoising

- Adversarial perturbations lead to noise in the features constructed by networks

- Uses ResNet like denoising block that has denoising operation. The networks are trained end-to-end on adversarially generated samples, allowing them to learn to reduce feature-map perturbations.



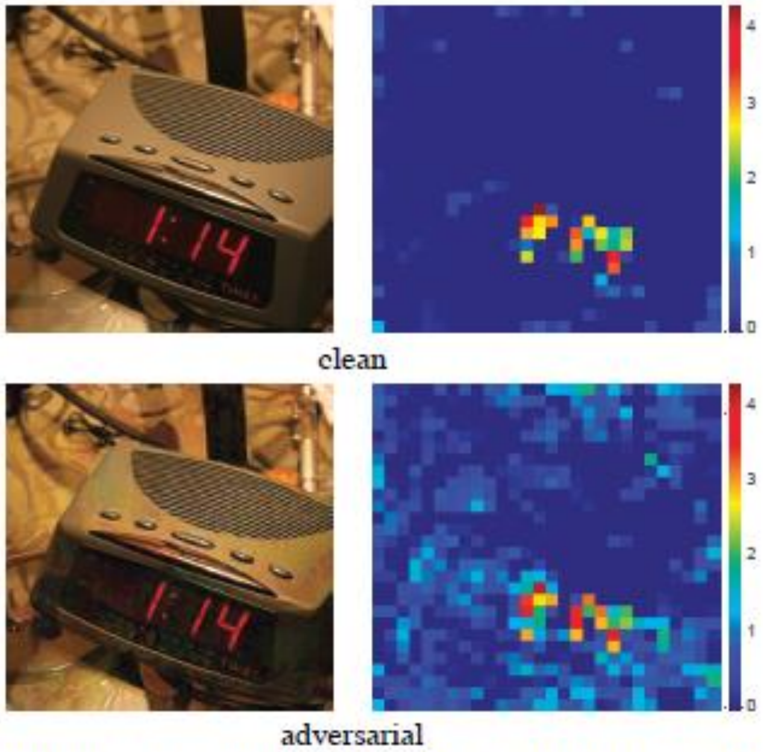


Figure 1. Feature map in the res_3 block of an ImageNet-trained ResNet-50 [9] applied on a clean image (top) and on its adversarially perturbed counterpart (bottom). The adversarial perturbation was produced using PGD [16] with maximum perturbation $\epsilon = 16$ (out of 256). In this example, the adversarial image is incorrectly recognized as “space heater”; the true label is “digital clock”.

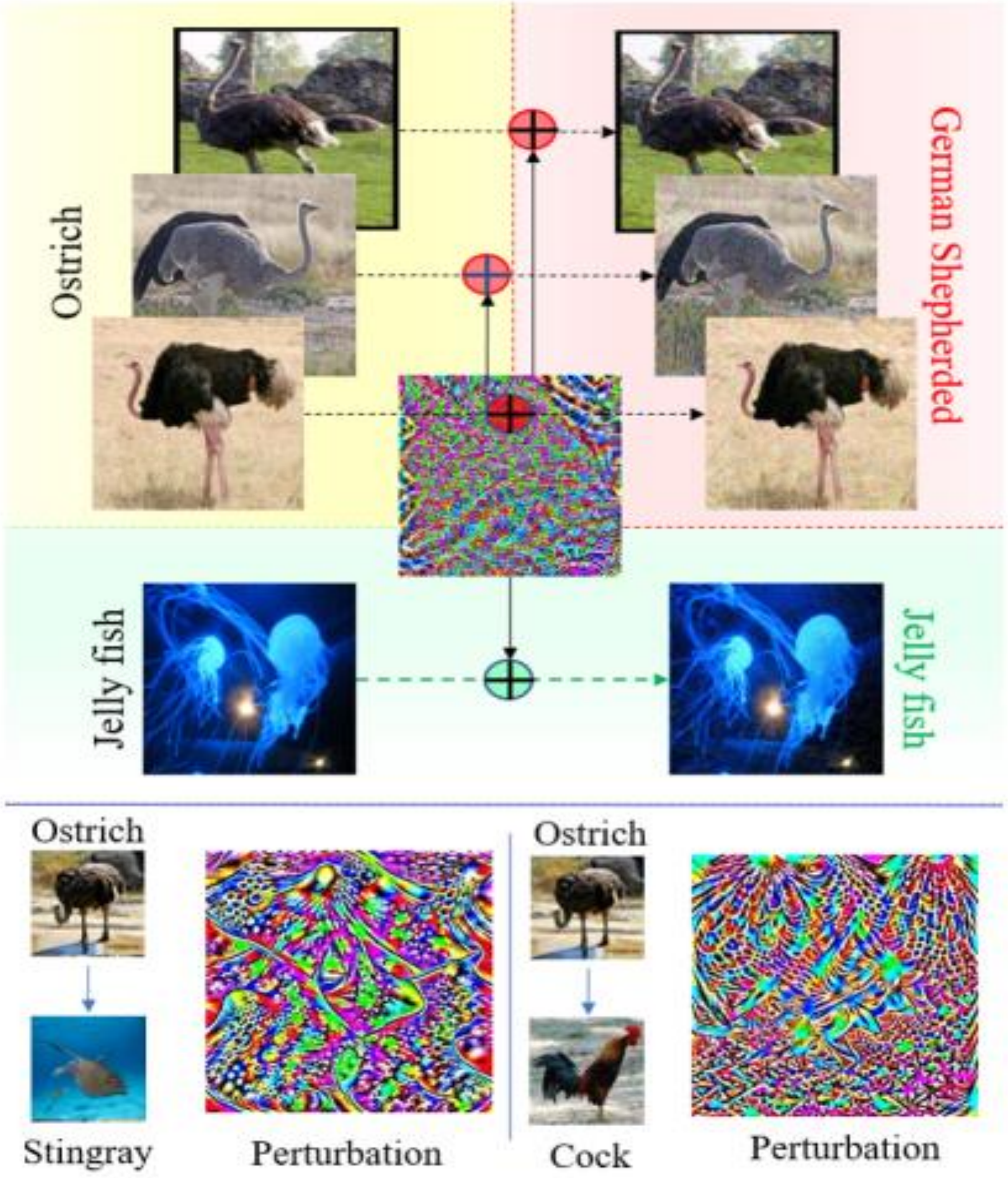
CVPR 2019, Xie et al. “Feature denoising for improving adversarial robustness”

Label Universal Targeted Attack (LUTA)

N. Akhtar, A. Jalwana, M. Bennamoun and Ajmal Mian, “Label Universal Targeted Attack”, arXiv:1905.11544, 2019.
under minor revision in IEEE Trans on Pattern Analysis & Machine Intelligence.

Label Universal Targeted Attack

- The attack is triggered only on a user selected source class
- LUTA fools the network to classify the source class to a specific target class, also user selected
- LUTA is useful beyond fooling (Interesting patterns and region properties)
- Demonstration over a variety of network for ImageNet dataset



Problem Formulation

Let $\Im \in \mathbb{R}^{\mathrm{d}}$ denote the distribution of natural images, and 'ℓ' be the label of its random sample $\mathbf{I}_\ell \overset{\text{rand}}{\sim} \Im$. Let $\mathcal{C}(.)$ be the classifier that maps $\mathcal{C}(\mathbf{I}_\ell) \rightarrow \ell$ with high probability. We restrict the classifier to be a deep neural network with cross-entropy loss. To fool $\mathcal{C}(.)$, we seek a perturbation $\boldsymbol{\rho} \in \mathbb{R}^{\mathrm{d}}$ that satisfies the following constraint:

$$\underset{\mathbf{I}_\ell \sim \Im}{\mathbf{P}}\left(\mathcal{C}(\mathbf{I}_\ell + \boldsymbol{\rho}) \rightarrow \ell_{\text{target}} : \ell_{\text{target}} \neq \ell\right) \geq \zeta \quad \text{s.t.} \quad ||\boldsymbol{\rho}||_p \leq \eta, \tag{1}$$

Algorithm LUTA

Algorithm 1 Label Universal Targeted Attack

Input: Classifier $\mathcal{C}$, source class samples $\mathcal{S}$, non-source class samples $\overline{\mathcal{S}}$, target label ℓ_{target}, perturbation norm η, mini-batch size b, fooling ratio ζ.

Output: Targeted label universal perturbation $\boldsymbol{\rho} \in \mathbb{R}^{\text{d}}$.

Initialize $\boldsymbol{\rho}_0, \boldsymbol{v}_0, \boldsymbol{\omega}_0$ to zero vectors in $\mathbb{R}^{\text{d}}$ and $t = 0$. Set $\beta_1 = 0.9$, and $\beta_2 = 0.999$.

while fooling ratio $< \zeta$ **do**

- $\mathcal{S}_s \overset{\text{rand}}{\sim} \mathcal{S}, \mathcal{S}_o \overset{\text{rand}}{\sim} \overline{\mathcal{S}} : |\mathcal{S}_s| = |\mathcal{S}_o| = \frac{b}{2}$ ◁ *get random samples from the source and other classes*
- $\mathcal{S}_s \leftarrow \text{Clip}\,(\mathcal{S}_s \ominus \boldsymbol{\rho}_t), \mathcal{S}_o \leftarrow \text{Clip}\,(\mathcal{S}_o \ominus \boldsymbol{\rho}_t)$ ◁ *perturb and clip samples with the current estimate*
- $t \leftarrow t + 1$ ◁ *increment*
- $\delta \leftarrow \frac{\underset{\mathbf{s}_i \in \mathcal{S}_s}{\mathbb{E}}[||\nabla_{\mathbf{s}_i} \mathcal{J}(\mathbf{s}_i, \ell_{\text{target}})||_2]}{\underset{\mathbf{s}_i \in \mathcal{S}_o}{\mathbb{E}}[||\nabla_{\mathbf{s}_i} \mathcal{J}(\mathbf{s}_i, \ell)||_2]}$ ◁ *compute scaling factor for gradient normalization*
- $\boldsymbol{\xi}_t \leftarrow \frac{1}{2}\left(\underset{\mathbf{s}_i \in \mathcal{S}_s}{\mathbb{E}}\left[\nabla_{\mathbf{s}_i} \mathcal{J}(\mathbf{s}_i, \ell_{\text{target}})\right] + \delta \underset{\mathbf{s}_i \in \mathcal{S}_o}{\mathbb{E}}\left[\nabla_{\mathbf{s}_i} \mathcal{J}(\mathbf{s}_i, \ell)\right]\right)$ ◁ *compute Expected gradient*
- $\boldsymbol{v}_t \leftarrow \beta_1 \boldsymbol{v}_{t-1} + (1 - \beta_1)\boldsymbol{\xi}_t$ ◁ *first moment estimate*
- $\boldsymbol{\omega}_t \leftarrow \beta_2 \boldsymbol{\omega}_{t-1} + (1 - \beta_2)(\boldsymbol{\xi}_t \odot \boldsymbol{\xi}_t)$ ◁ *raw second moment estimate*
- $\boldsymbol{\rho} \leftarrow \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \,\text{diag}\left(\text{diag}(\sqrt{\boldsymbol{\omega}_t})^{-1} \boldsymbol{v}_t\right)$ ◁ *bias corrected moment ratio*
- $\boldsymbol{\rho}_t \leftarrow \boldsymbol{\rho}_{t-1} + \frac{\boldsymbol{\rho}}{||\boldsymbol{\rho}||_\infty}$ ◁ *update perturbation*
- $\boldsymbol{\rho}_t \leftarrow \Psi(\boldsymbol{\rho}_t)$ ◁ *project on ℓ_p ball*

end while

return

Results on ImageNet Models

Bound	Model	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	Avg.	Leak.
ℓ_∞-norm	VGG-16 [38]	92	76	80	74	82	78	82	80	74	88	80.6±5.8	29.9
	ResNet-50 [16]	92	78	80	72	76	84	78	76	82	78	79.6±5.4	31.1
	Inception-V3 [40]	84	60	70	60	68	90	68	62	72	76	71.0±9.9	24.1
	MobileNet-V2 [36]	92	94	88	78	88	86	74	86	84	94	86.4±6.5	37.1
ℓ_2-norm	VGG-16 [38]	90	84	80	84	94	86	82	92	86	96	87.4±5.3	30.4
	ResNet-50 [16]	96	94	88	84	90	86	86	94	90	90	89.8±3.9	38.0
	Inception-V3 [40]	86	68	62	62	74	72	74	68	66	76	70.8±7.2	45.6
	MobileNet-V2 [36]	94	98	92	76	94	92	76	92	92	96	90.2±7.7	56.0

Table 1. Fooling ratios (%) with $\eta = 15$ for ℓ_∞ and $4,500$ for ℓ_2-norm bounded label-universal perturbations for ImageNet models. The label transformations are T_1: Airship $\rightarrow$ School Bus, T_2: Ostrich $\rightarrow$ Zebra, T_3: Lion $\rightarrow$ Orangutang, T_4: Bustard $\rightarrow$ Camel, T_5: Jelly Fish $\rightarrow$ Killer Wahle, T_6: Life Boat $\rightarrow$ White Shark, T_7: Scoreboard $\rightarrow$ Freight Car, T_8: Pickelhaube $\rightarrow$ Stupa, T_9: Space Shuttle $\rightarrow$ Steam Locomotive, T_{10}: Rapeseed $\rightarrow$ Butterfly. Leakage (last column) is the average fooling of non-source classes into the target label.

Visualization of Perturbations

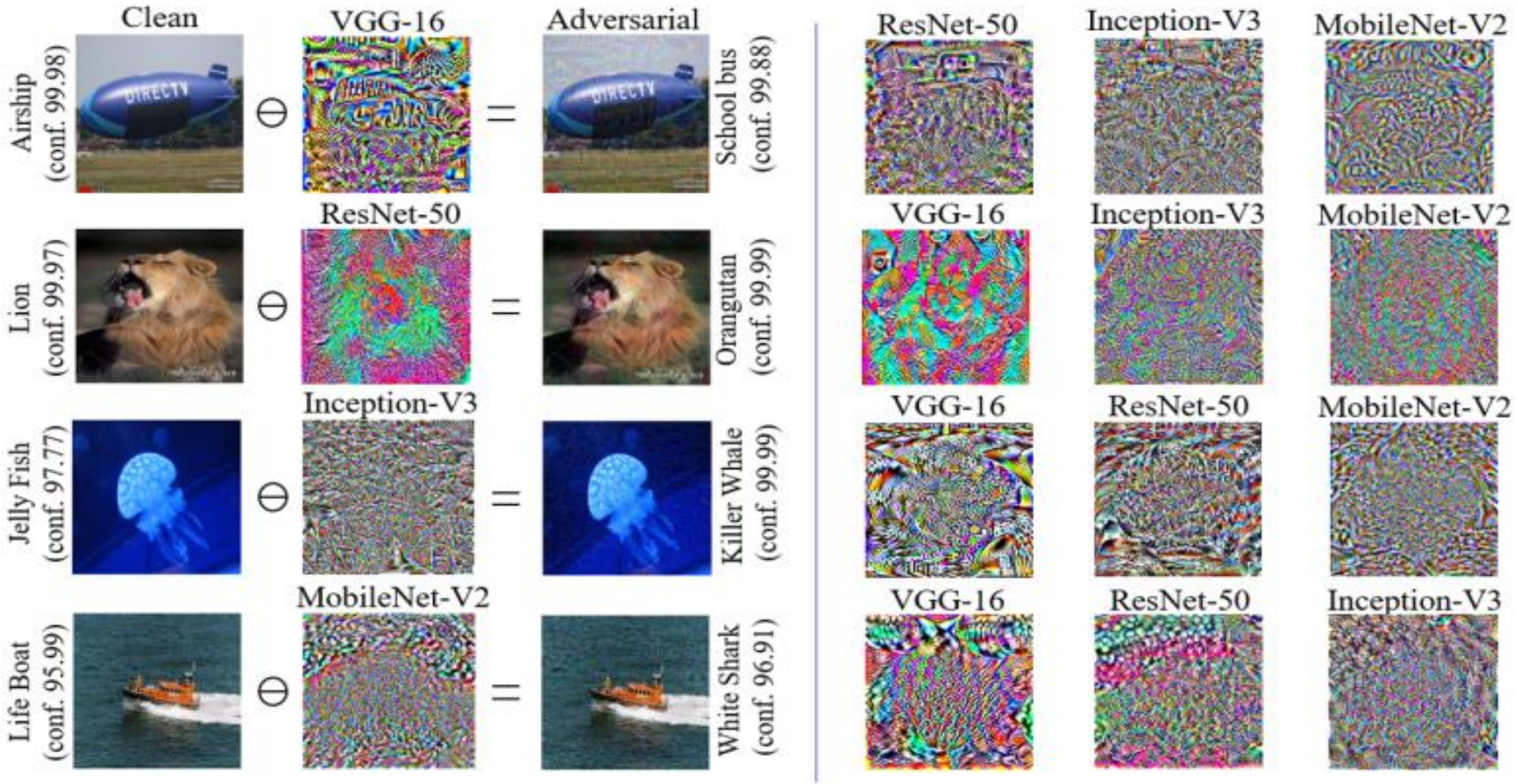
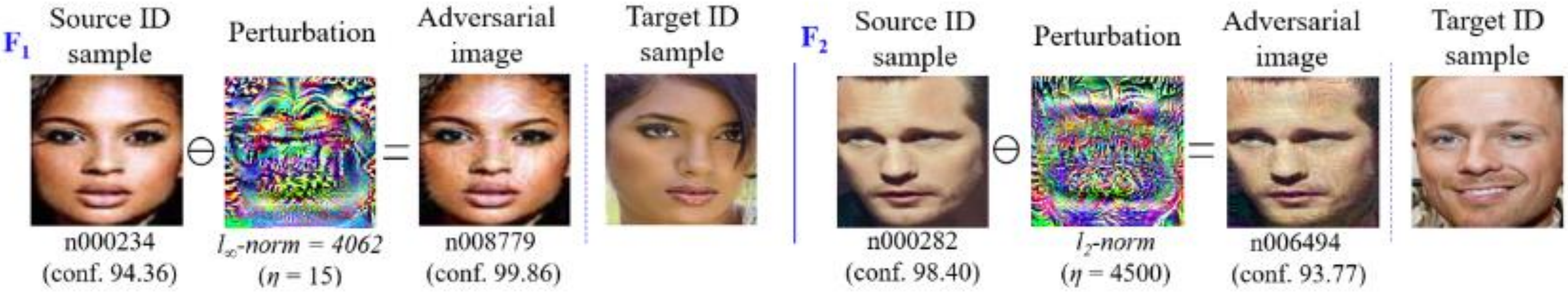


Figure 2. Representative perturbations and adversarial images for ℓ_∞-bounded case ($\eta = 15$). A row shows perturbations for the same source $\rightarrow$ target fooling for the mentioned models. An adversarial example for each model is also shown for reference (left). Following [41], the perturbations are visualized by 10x magnification, shifted by 128 and clamped to 0-255.

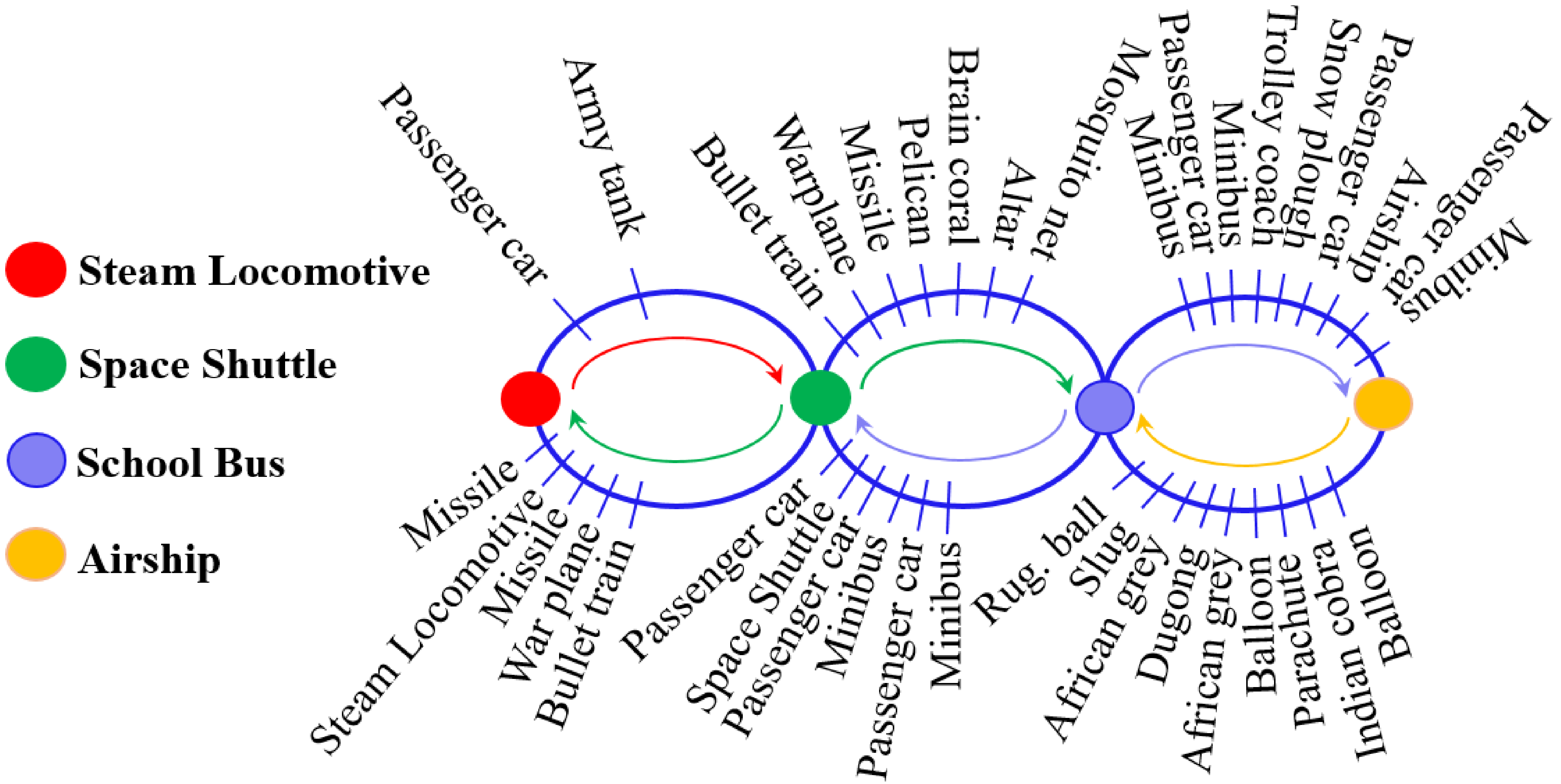
Attack on Face Recognition



ℓ_∞-norm bounded							ℓ_2-norm bounded						
F_1	F_2	F_3	F_4	F_5	Avg.	Leak.	F_1	F_2	F_3	F_4	F_5	Avg.	Leak.
88	76	74	86	84	81.6±6.2	1.9	76	80	78	76	84	78.8±3.3	1.8

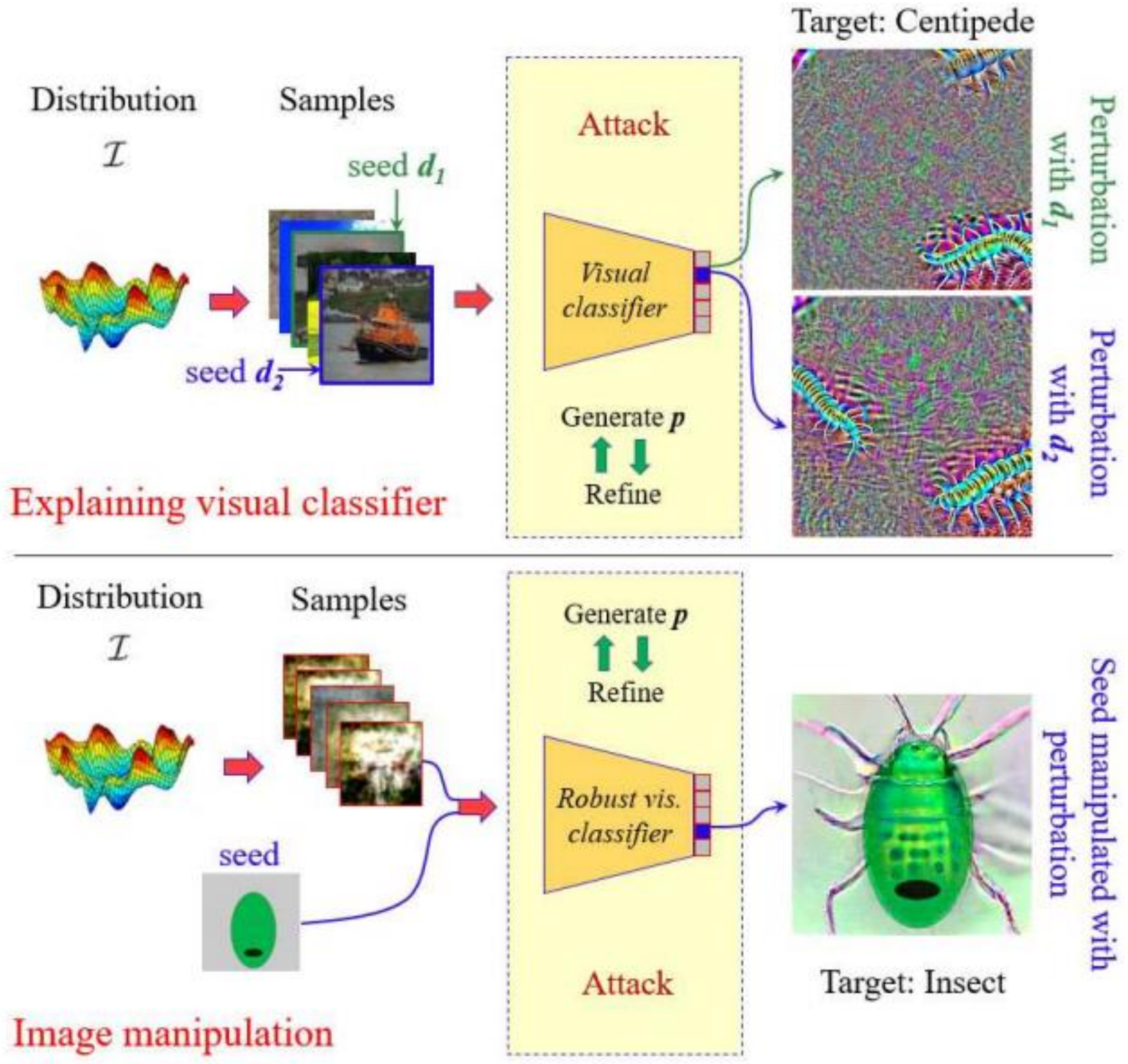
VGGFace data and model were used. F_1, F_2, F_3, F_4, F_5 define face ID switches between certain IDs. Notice the high fooling rate and negligible leakage.

Decision Boundary Analysis



Attack to Explain

Features learned by deep models are in fact aligned with human perception as opposed to the common belief



M. Jalwana, N. Akhtar, M. Bennamoun, Ajmal Mian, "Attack to explain deep representation", CVPR 2020.

Looking at the Perturbations

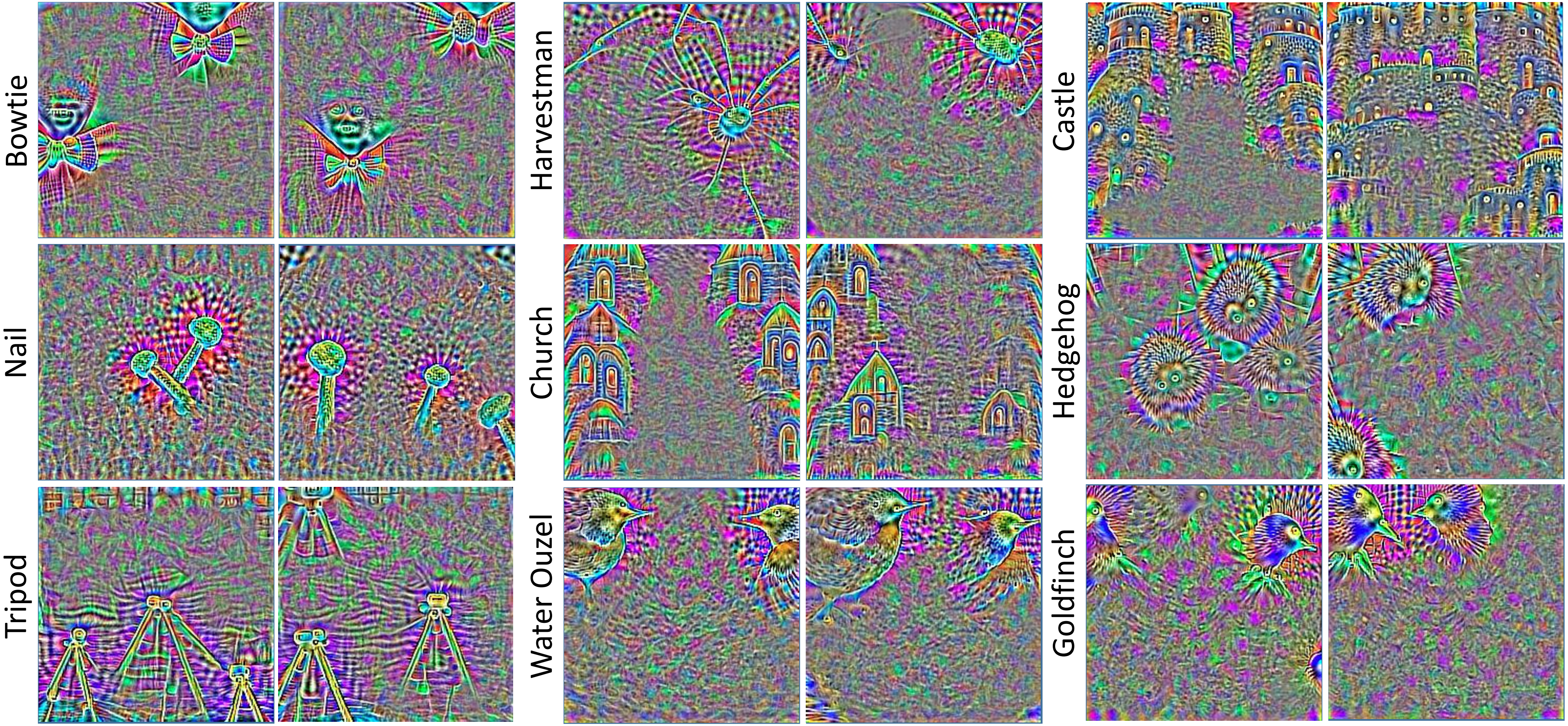


Image Generation

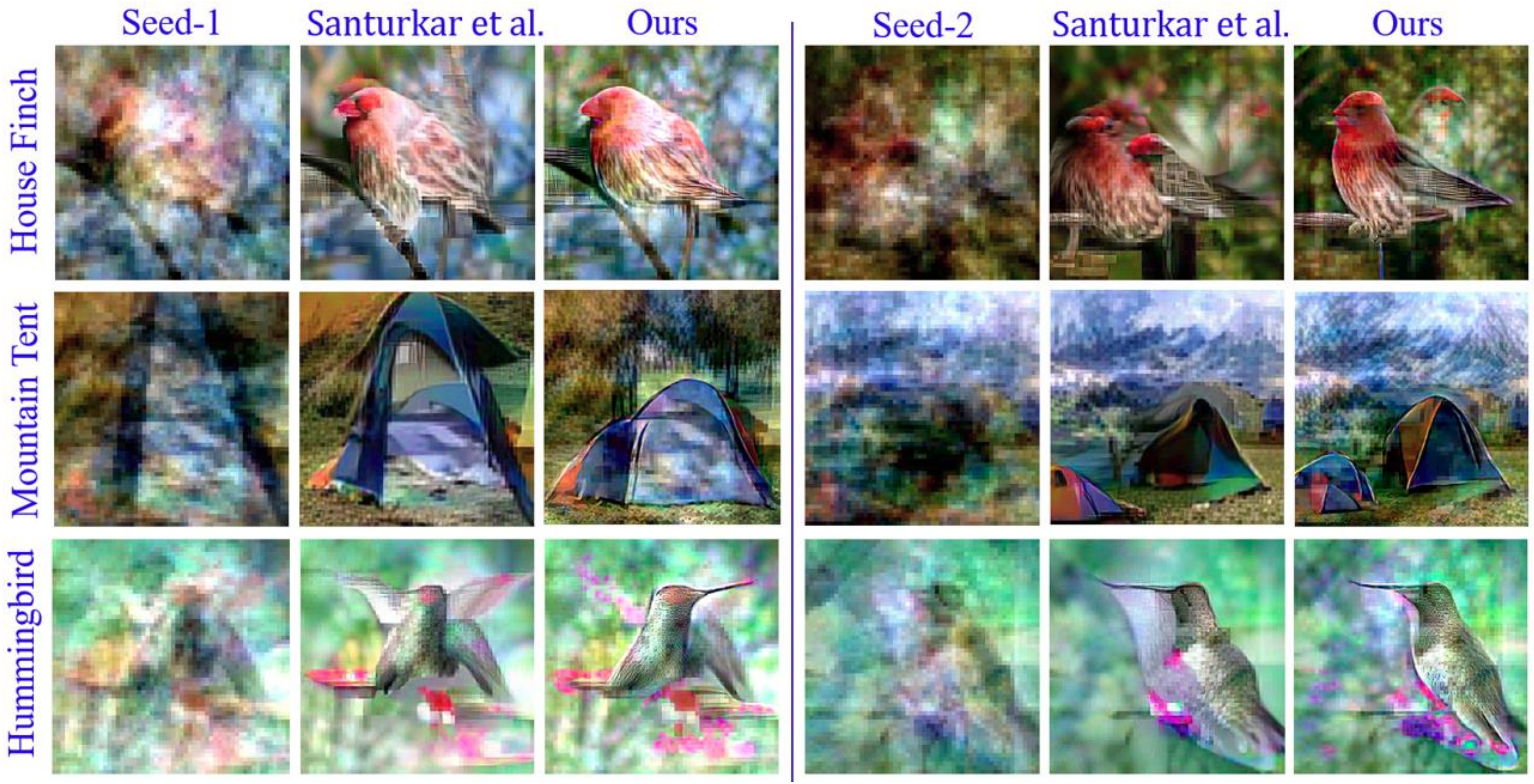
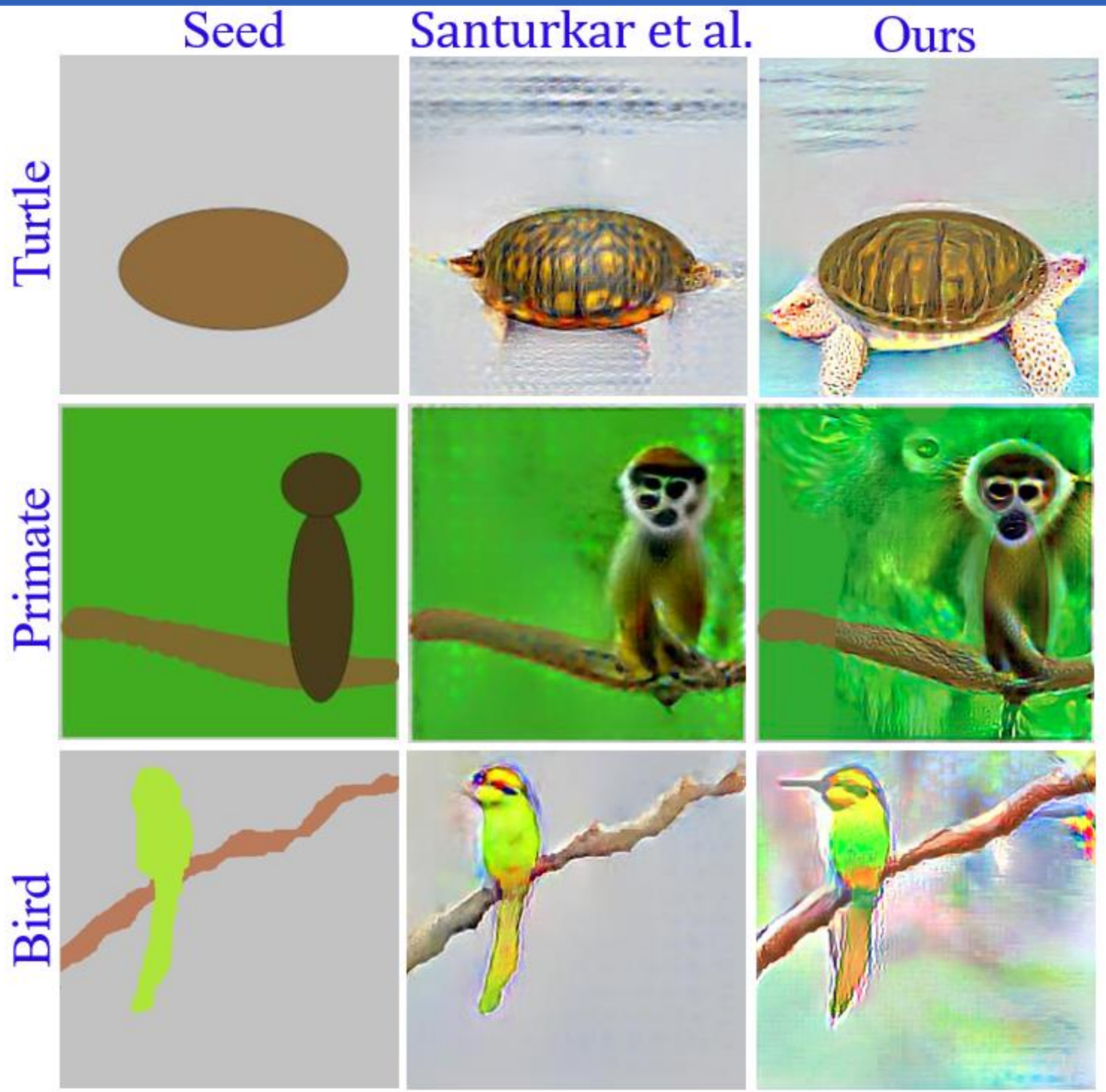


Image Manipulation



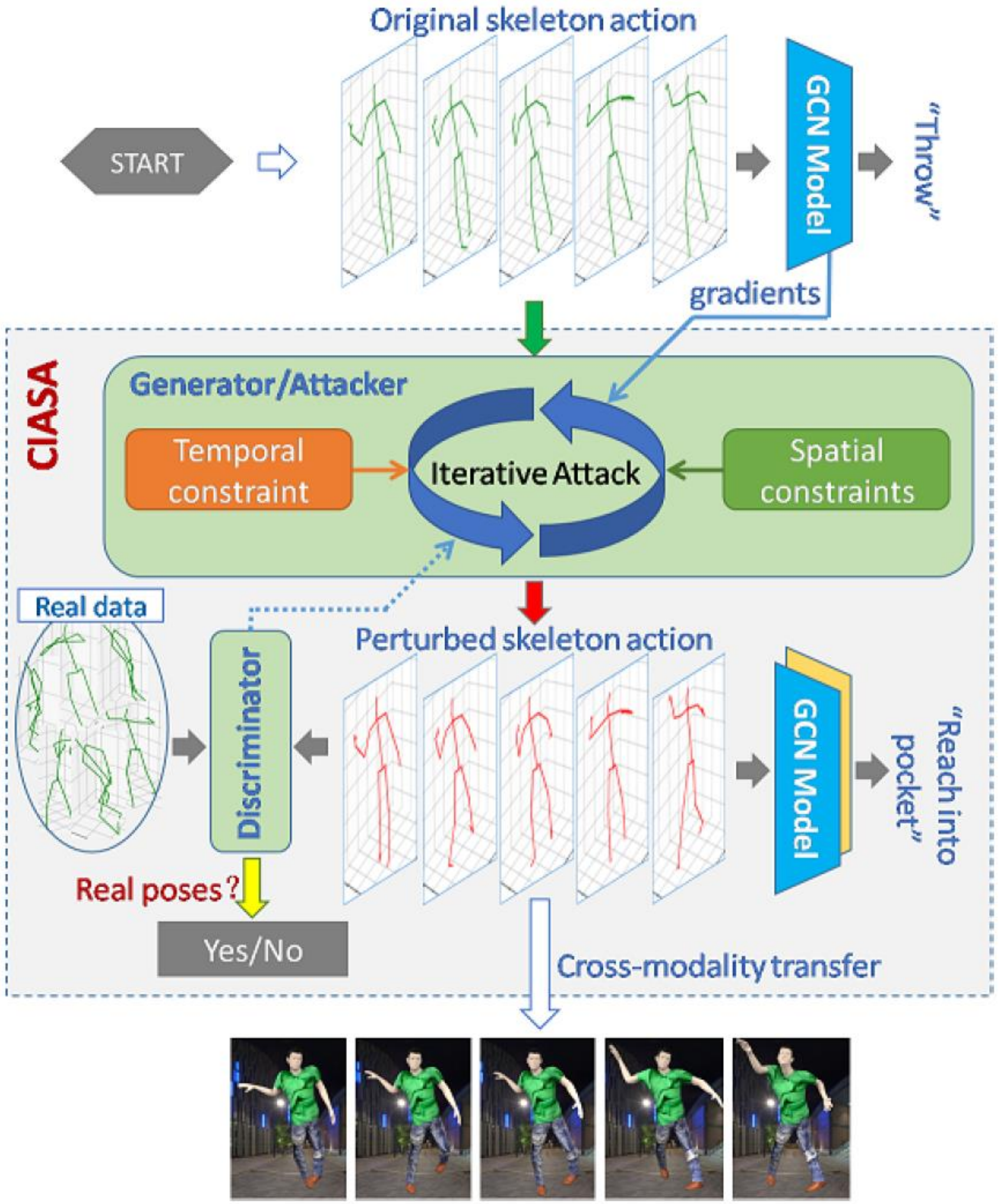
Adversarial Attack on Skeleton-based Human Action Recognition

Jian Liu, Naveed Akhtar, Ajmal Mian

IEEE Transactions on Neural Networks and Learning Systems (TNNLS) 2020

Constrained Iterative Attack for Skeleton Actions

- Like all attacks, our CIASA works on end-to-end deep models
- Attack-Generator ←→ Pose-Discriminator
- The attack generator perturbs the joints iteratively given spatial and temporal constraints
- The discriminator ensures that the perturbed pose is real



ST-GCN Overview

- An action is represented as a sequence of T skeleton frames, each consisting N body joints. An undirected graph $G = (V; E)$ is constructed from the $N \times T$ joints.
- Edges are intra-body $E^S$and inter-frame E^F
- E^S is represented as $N \times N$ binary adjacency matrix specifying connected and unconnected joints of graph nodes
- Graph Convolution at a vertex v_i over vertices v_j is defined as

$$f_{out}(v_i) = \sum_{v_j \in B(v_i)} \frac{1}{Z_i(v_j)} f_{in}(v_j) \cdot w(l_i(v_j))$$

B is the sampling function to define a neighboring node set, l is a labelling function and w are the convolution weights. B and l operate in the spatio-temporal region.

S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," AAAI 2018.

Constraints on Attack-Generator

A. Joints Variation Constraint – joints should not move too far
   1. Global Clipping
   2. Hierarchical Clipping

B. Bone Length Constraint – NO stretching or shrinking of bones

C. Temporal Dynamics Constraint – perturbations should be temporally smooth

D. Anthropometric Plausibility – perturbed skeleton should correspond to a possible human pose

Constrained Iterative Attacker

Our attack is targeted but can degenerate to untargeted

Temporal Smoothness

Anthropometric plausibility (GAN)

Joint variation (magnitude)

Bone length constraint

Algorithm 1 Constrained iterative attacker $\mathcal{A}$ to fool skeleton-base action recognition.

Input: Original graph nodes $V^0 \in \mathbb{R}^{3\times N\times T}$, trained ST-GCN model $\mathcal{F}_\theta()$, desired target class c_{target}, perturbation clipping factor ϵ, max_iter=M, learning rate α
Output: Perturbed graph nodes $V' \in \mathbb{R}^{3\times N\times T}$.

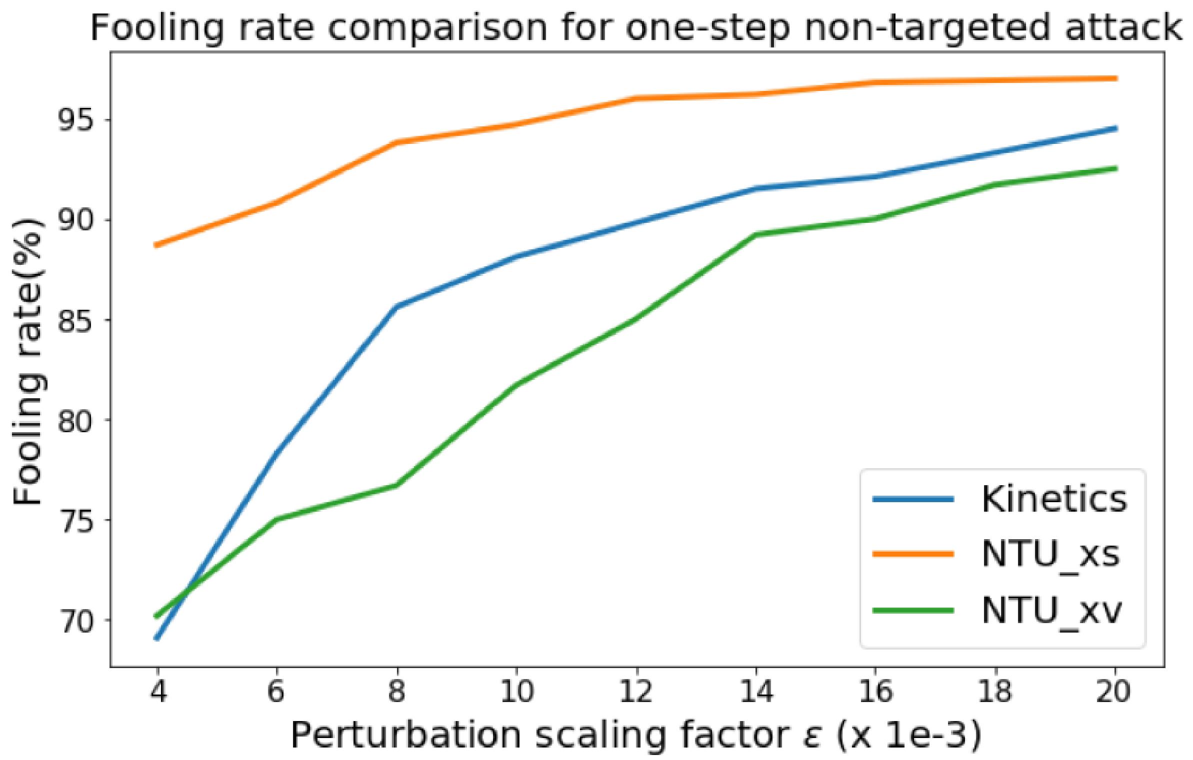
```
set initial V' = V^0
while i < M do
    feed forward Z = F_θ(V')
    L_pred = CrossEntroyLoss(Z, c_target)
    L_smooth = 1/(T-1) Σ_{t=2}^{T} f̈'_t
    L_adv(A) = (D_ω(A(V')) - 1)^2
    L_adv(D) = (D_ω(Ṽ) - 1)^2 + D_ω(V')^2
    L_CIASA = L_pred + λ(L_smooth + L_adv)
    (L_CIASA).Backward() ⇒ gradients
    V', ω = AdamOptimizer([V', ω], gradients)
    if |V' - V^0| > ε then
        V' = Clip(V') ~ [V^0 - ε, V^0 + ε])
    end if
    Skeleton realignment V' = SSR(V')
    i = i + 1
end while
return V'
```

Simple Case (one step non-targeted)

- NTU dataset: 3D human skeletons captured with Kinect-v2. There are 56,880 samples of 60 actions.

- Kinetics: RGB videos of 400 actions with 400+ samples per action

- OpenPose to get the skeleton joints from the Kinetics dataset



ϵ is defined as a fraction of the average skeletal height

Results for Targeted Attack

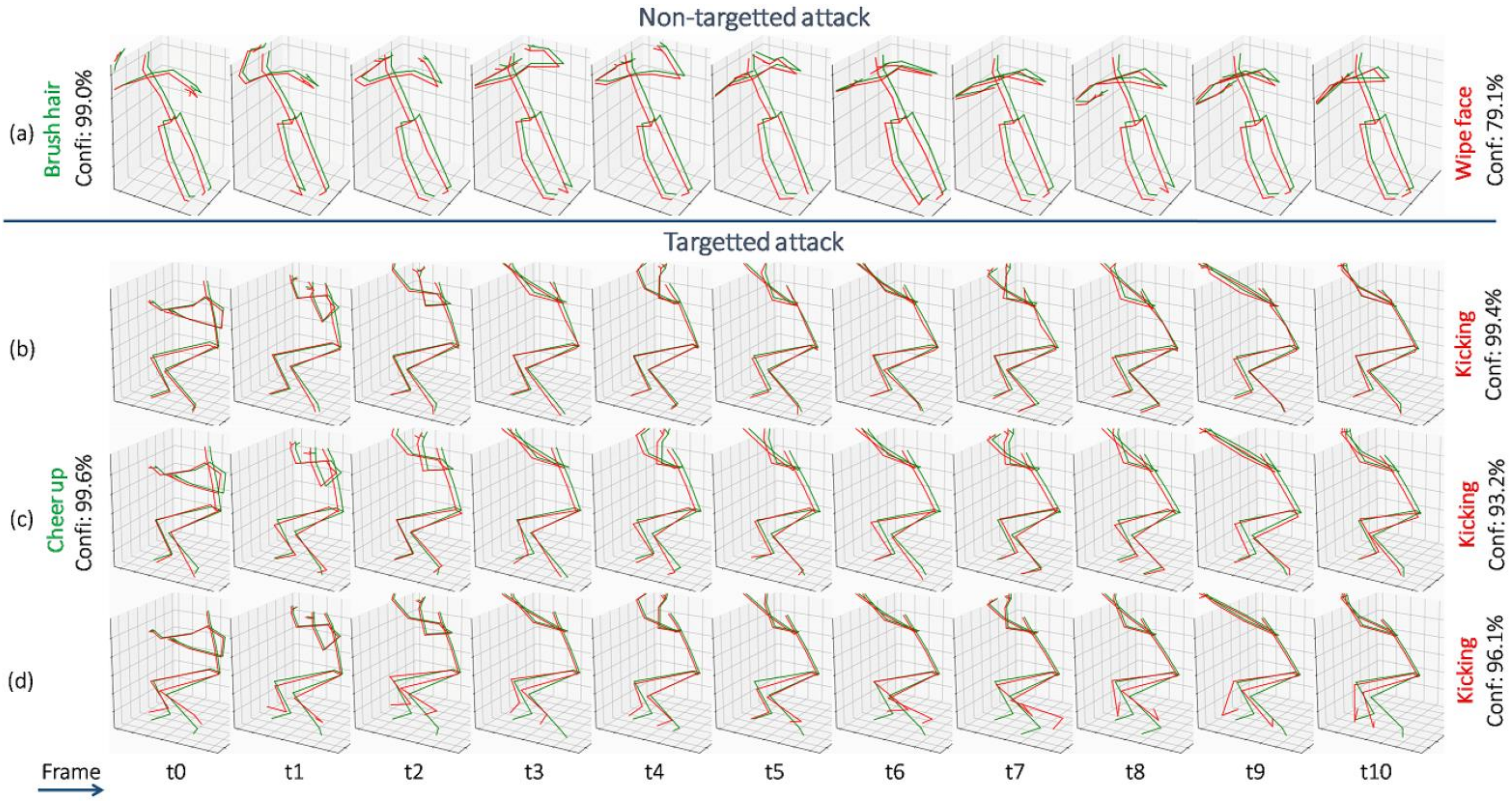
<u>Target is the least likely class.</u>

FOOLING RATES (%) ACHIEVED BY CIASA TARGETED ATTACK (BASIC MODE) WITH DIFFERENT GLOBAL CLIPPING STRENGTH ϵ FOR NTU AND KINETICS DATASETS. BOTH CROSS-SUBJECT NTU_{XS} AND CROSS-VIEW NTU_{XV} PROTOCOLS ARE CONSIDERED FOR THE NTU DATASET.

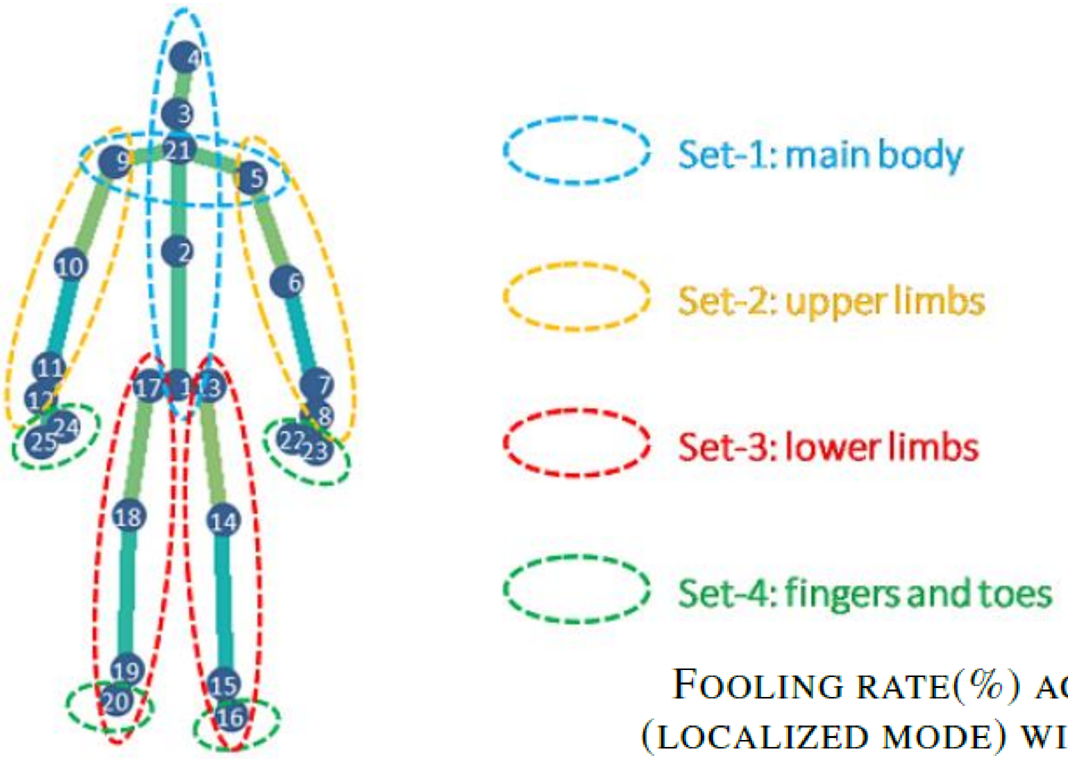
ϵ (× 1e-3)	4	6	8	10	12
Kinetics	82.5	92.5	96.5	97.5	99.3
NTU_{XS}	89.4	96.6	98.7	99.2	99.8
NTU_{XV}	78.2	85.5	93.3	98.9	99.6

Visualization of Attacks

There is an intentional offset so you can see the original (green) and perturbed (red) skeleton.



Localized + Targeted Attack



FOOLING RATE(%) ACHIEVED BY CIASA TARGETED ATTACK (LOCALIZED MODE) WITH DIFFERENT ATTACK REGIONS ON NTU DATASET. BOTH CROSS-SUBJECT AND CROSS-VIEW PROTOCOLS ARE EVALUATED. GLOBAL CLIPPING STRENGTH IS SET TO $\epsilon = 0.04$.

Attack region	set-1	set-2	set-3	set-4
NTU_{XS}	90.8	93.3	61.3	83.3
NTU_{XV}	85.2	91.7	60.0	81.7

Cross Network Transferability

- The 2s-AGCN is two-stream (joint locations + bone directions) adaptive GCN which models a learnable topology of the skeleton

COMPARISON OF CROSS-MODEL RECOGNITION ACCURACY (%) AND FOOLING RATE (%) ON THREE CONFIGURATIONS OF 2S-AGCN FOR CROSS-VIEW NTU PROTOCOL. 'ORIGINAL ACCURACY' IS ON CLEAN DATA. 'ATTACKED ACCURACY' IS ON PERTURBED DATA.

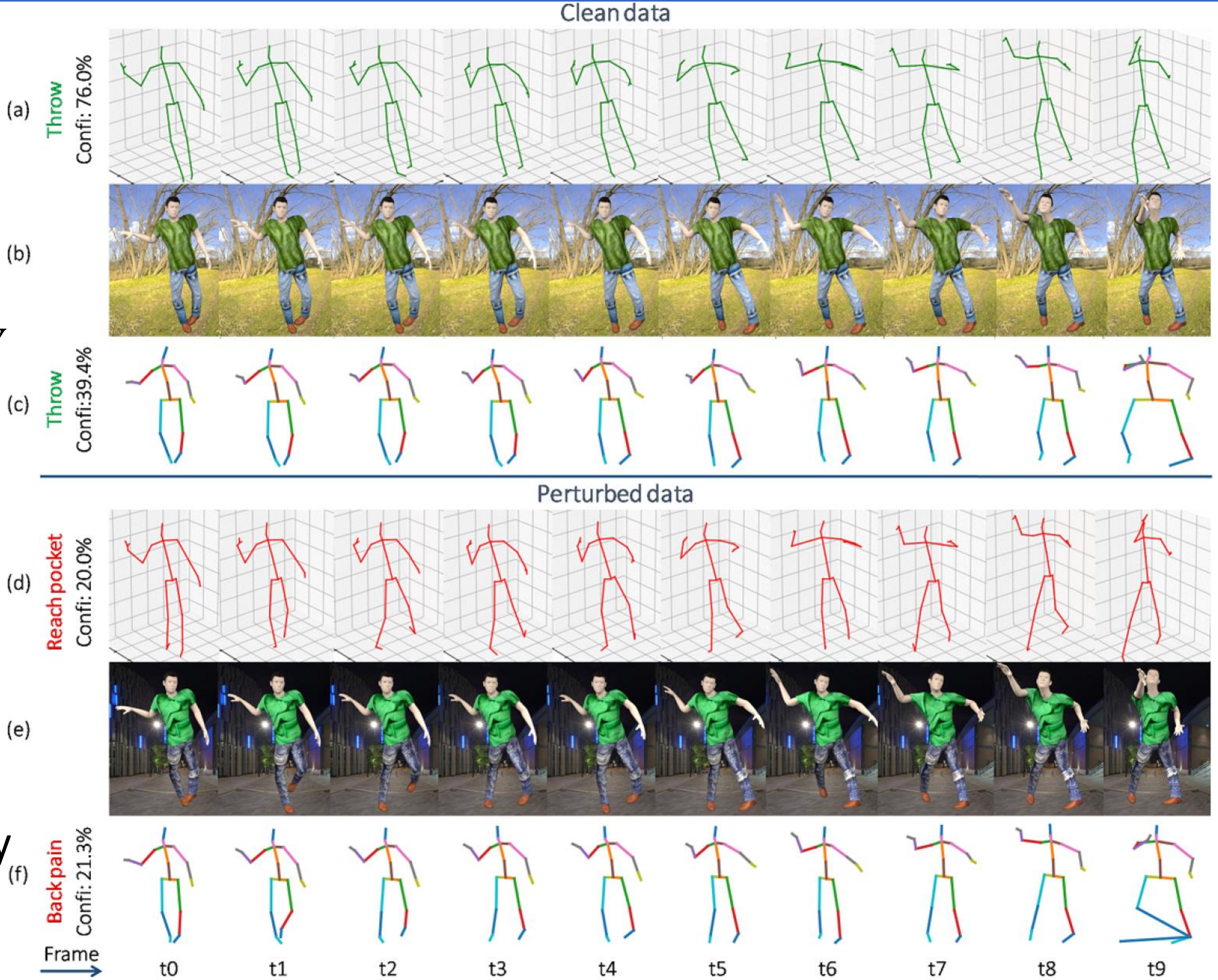
Model	Js-AGCN	Bs-AGCN	2s-AGCN
Original Accuracy	93.7	93.2	95.1
Attacked Accuracy	13.5	6.8	11.8
Fooling rate (%)	86.1	93.1	88.4

CIASA basic mode
$\epsilon = 0.012$

L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," CVPR 2019.

Cross Modality Transferability

- Render the skeletons in Blender using MakeHuman models
- Recover skeleton back with VNect
- Use 240 skeleton actions from NTU_{XV}
- Classify actions with ST-GCN on the VNect-recovered skeleton sequences
- 53.3% accuracy for clean data
  38.9% accuracy for perturbed data
- However, the attack does transfer to RGB video which is intriguing
- These are the first ever cross-modality results on adversarial attacks



Conclusions

- Deep learning is vulnerable to adversarial attacks in white-box and black box setting
- Attacks learned for one network transfer to other networks
- This is a serious threat to real world deployment of deep models
- A silver lining is that attacks can be used to understand deep networks
- Understanding the inner working of deep networks is a first step to achieving robust and explainable AI

Contributors and Code

Australian Government
Australian Research Council

Perturbation Rectifier https://github.com/liujianee/Pertrubation_Rectifying_Network

LUTA : https://github.com/AsimJalwana/LUTA

Synthetic video generation https://github.com/liujianee/MVIPER