Evasion Attacks Against Machine Learning Models (Other Methods)

Recall: Adversarial Examples

- FGSM

$$\boldsymbol{\eta} = \epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right)$$

- Optimization based attack

$$\min d(x, x') + g(x')$$
$$s.t.\ x' is \text{“}valid\text{”}$$

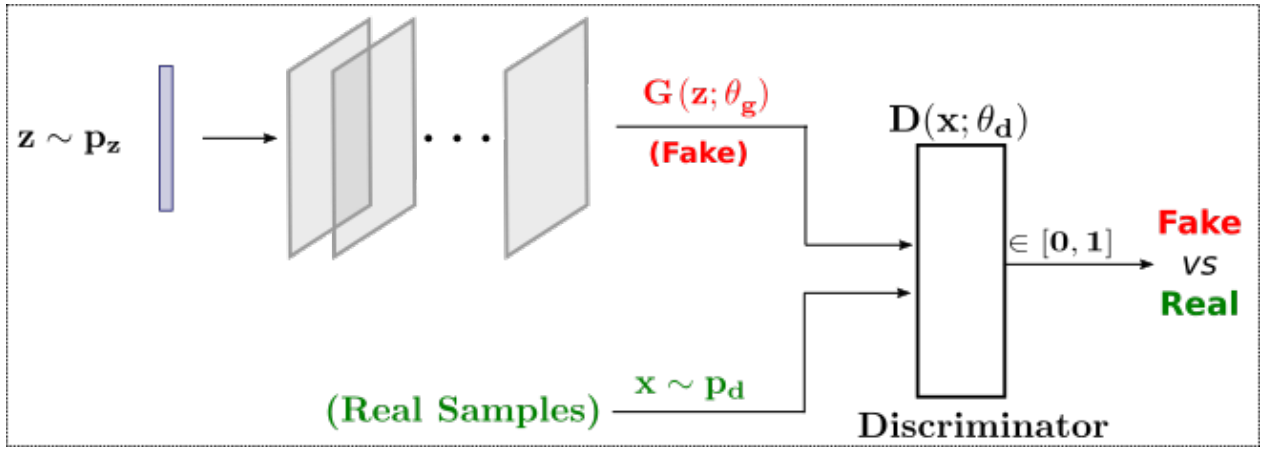
- DeepFool
  - Greedy algorithm to move the instance towards the nearest boundary
- JSMA (Jacobian-based Saliency Map Approach)
  - Compute the saliency map for an X regarding to target y*; modify the max pixel each time
- BIM (Basic Iterative Method)
  - Apply FGSM multiple times with small step size

Generating Adversarial Examples with Adversarial Networks

- How can we generate more realistic adversarial examples?
- How can we generate diverse adversarial examples?
- How to perform blackbox attack efficiently?

Generating Adversarial Examples with Adversarial Networks

- Generative adversarial networks (GANs)

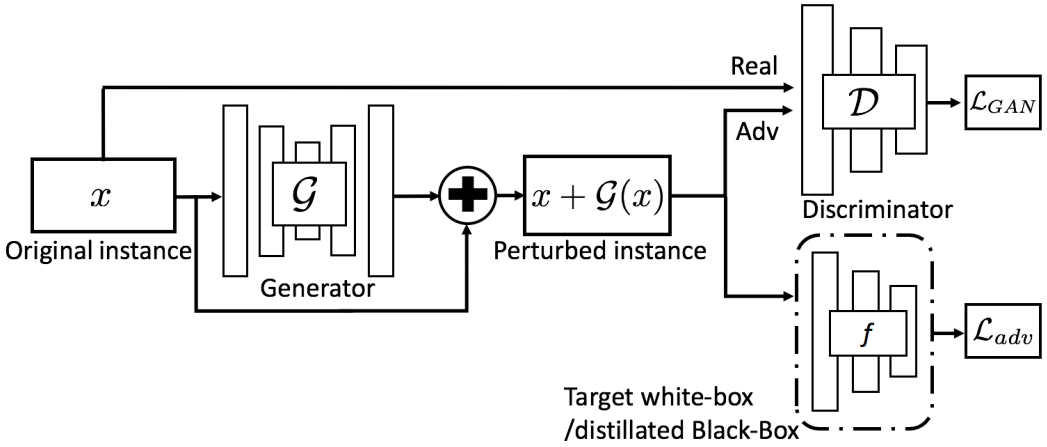


- Generate more realistic instances
- Approximate certain distribution
- Efficient once the generator is trained

Questions:

1. Can we generate more realistic adversarial examples?
2. Can we generate adversarial examples more efficiently?

Generating Adversarial Examples with Adversarial Networks



Black-box can be performed here via distillation

$$\mathcal{L} = \mathcal{L}_{adv}^{f} + \alpha \mathcal{L}_{GAN} + \beta \mathcal{L}_{hinge}$$

$$\mathcal{L}_{GAN} = \mathbb{E}_{x \sim \mathcal{P}_{\text{data}}(x)} \log \mathcal{D}(x) + \mathbb{E}_{x \sim \mathcal{P}_{\text{data}}(x)} \log(1 - \mathcal{D}(x + \mathcal{G}(x)))$$

$$\mathcal{L}_{\text{adv}}^{f} = \mathbb{E}_{x} \ell_f(x + \mathcal{G}(x), t)$$

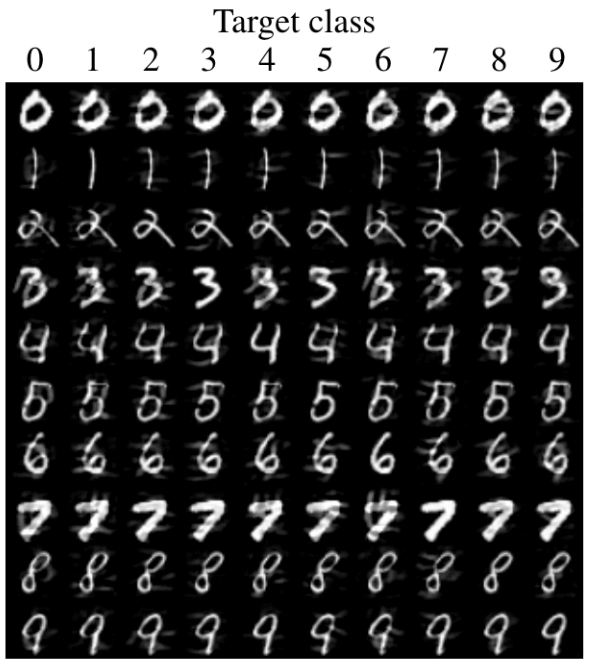
$$\mathcal{L}_{\text{hinge}} = \mathbb{E}_{x} \max(0, \|\mathcal{G}(x)\|_2 - c)$$

Generating Adversarial Examples with Adversarial Networks

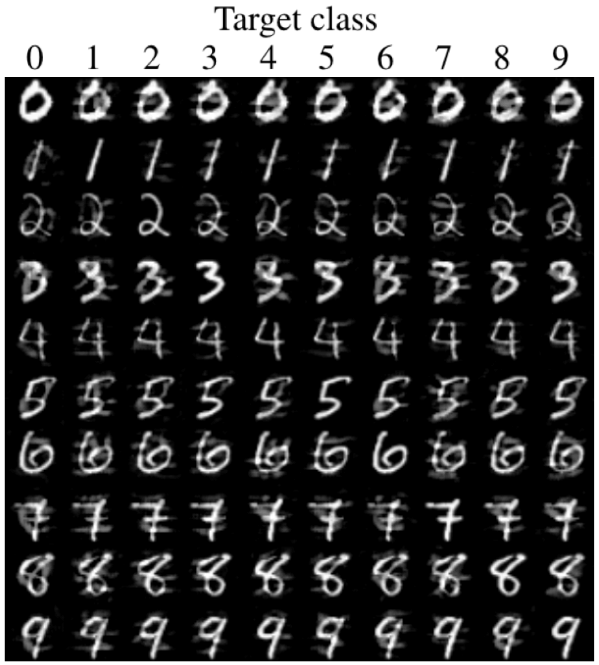
- Advantages

	FGSM	Opt.	Trans.	AdvGAN
Run time	0.06s	>3h	-	<0.01s
Targeted Attack	✓	✓	Ens.	✓
Black-box Attack			✓	✓

Generating Adversarial Examples with Adversarial Networks



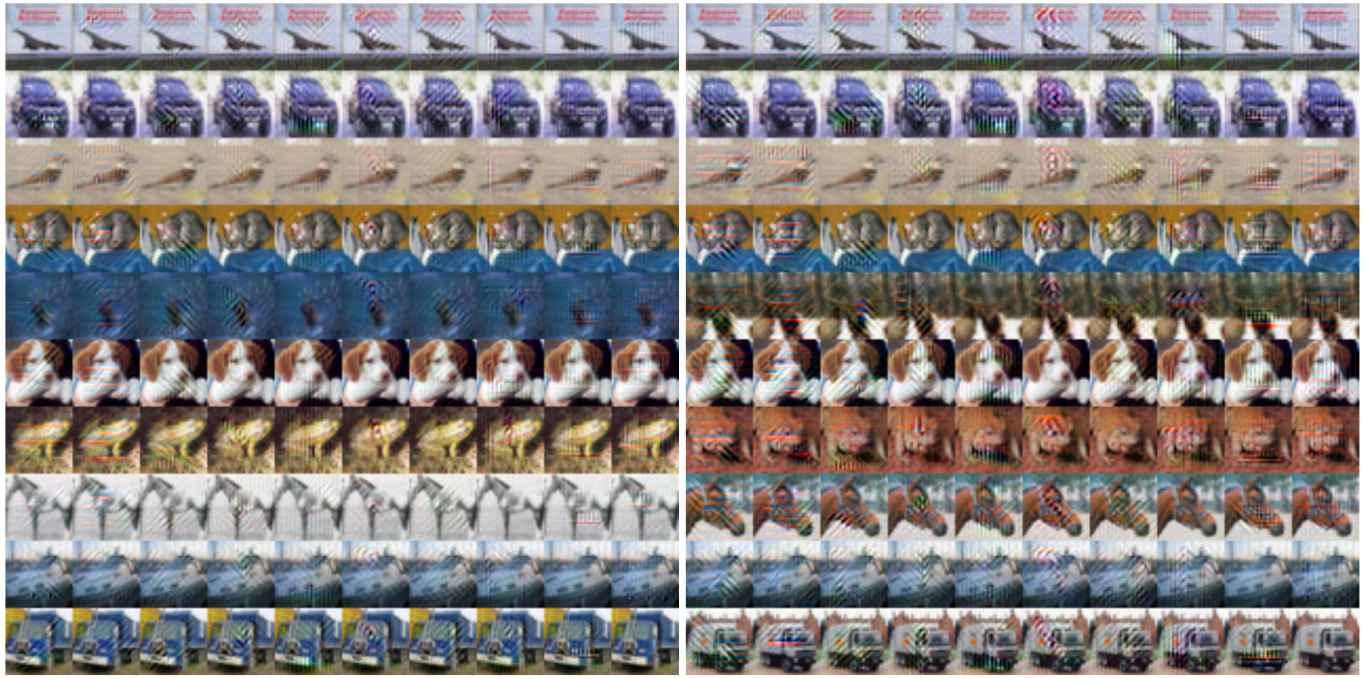
Semi-white box attack on MNIST



Black-box attack on MNIST

The perturbed images are very close to the original ones. The original images lie on the diagonal.

Generating Adversarial Examples with Adversarial Networks



(a) Semi-whitebox setting

(b) Black-box setting

The perturbed images are very close to the original ones. The original images lie on the diagonal.

Poodle Ambulance Basketball Electric guitar

(a) Strawberry

(b) Toy poodle

(c) Buckeye

(d) Toy poodle

Attack Effectiveness Under Defenses

Data	Model	Defense	FGSM	Opt.	AdvGAN
MNIST	A	Adv.	4.3%	4.6%	**8.0%**
		Ensemble	1.6%	4.2%	**6.3%**
		Iter.Adv.	4.4%	2.96%	**5.6%**
	B	Adv.	6.0%	4.5%	**7.2%**
		Ensemble	2.7%	3.18%	**5.8%**
		Iter.Adv.	**9.0%**	3.0%	6.6%
	C	Adv.	2.7%	2.95%	**18.7%**
		Ensemble	1.6%	2.2%	**13.5%**
		Iter.Adv.	1.6%	1.9%	**12.6%**
CIFAR	ResNet	Adv.	13.10%	11.9%	**16.03%**
		Ensemble.	10.00%	10.3%	**14.32%**
		Iter.Adv	22.8%	21.4%	**29.47%**
	Wide ResNet	Adv.	5.04%	7.61%	**14.26%**
		Ensemble	4.65%	8.43%	**13.94 %**
		Iter.Adv.	14.9%	13.90%	**20.75%**

Attack success rate of adversarial examples generated by AdvGAN in semi-whitebox setting under defenses on MNIST and CIFAR-10

Attack Effectiveness Under Defenses

Black-Box Leaderboard (Original Challenge)

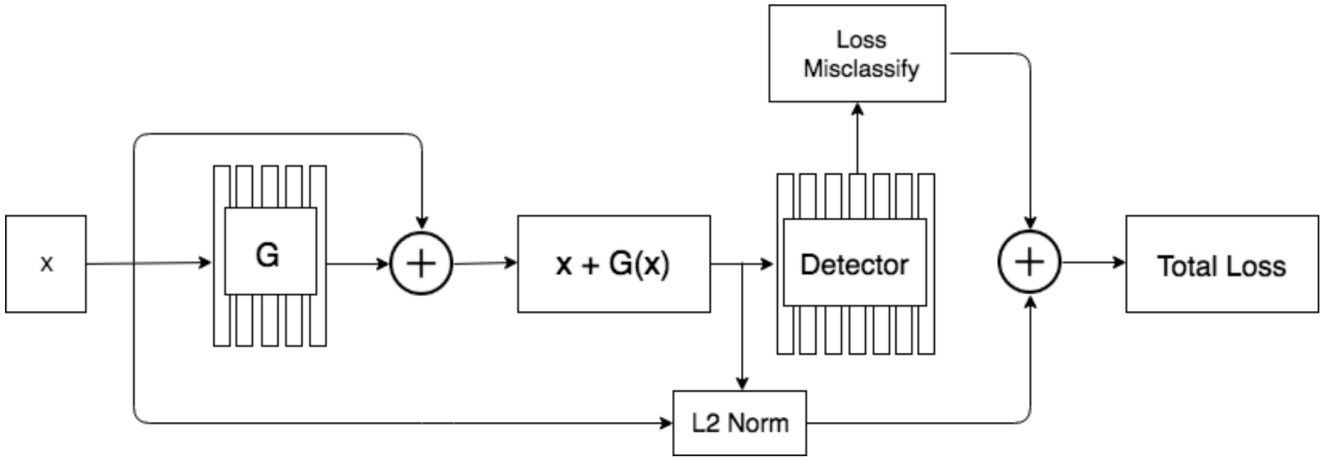
Attack	Submitted by	Accuracy	Submission Date
AdvGAN from "Generating Adversarial Examples with Adversarial Networks"	AdvGAN	**92.76%**	Sep 25, 2017
PGD against three independently and adversarially trained copies of the network	Florian Tramèr	93.54%	Jul 5, 2017
FGSM on the CW loss for model B from "Ensemble Adversarial Training [...]"	Florian Tramèr	94.36%	Jun 29, 2017
FGSM on the CW loss for the naturally trained public network	(initial entry)	96.08%	Jun 28, 2017
PGD on the cross-entropy loss for the naturally trained public network	(initial entry)	96.81%	Jun 28, 2017
Attack using Gaussian Filter for selected pixels on the adversarially trained public network	Anonymous	97.33%	Aug 27, 2017
FGSM on the cross-entropy loss for the adversarially trained public network	(initial entry)	97.66%	Jun 28, 2017
PGD on the cross-entropy loss for the adversarially trained public network	(initial entry)	97.79%	Jun 28, 2017

Takeaways

- Adversarial examples and generative adversarial networks are different
- We can integrate them together to work better
- Generative models can indeed synthesize new types of adversarial examples
- Adversarial retraining based defense is not enough

Similar work

- Adversarial Attacks on Face Detectors using Neural Net based Constrained Optimization



$$L_G(x, x') = \|x - x'\|_2^2 + \lambda \sum_{i=1}^{N} (Z(x'_i)_{\text{background}} - Z(x'_i)_{\text{face}})^+$$

Difference: attacking detector, face detection task

Similar work

- Generative Adversarial Examples

$$\mathcal{L} = \mathcal{L}_0 + \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2$$

$$\mathcal{L}_0 \triangleq -\log p(f(g_\theta(z, y_{\text{source}})) = y_{\text{target}})$$

$$\mathcal{L}_1 \triangleq \frac{1}{m} \sum_{i=1}^{m} \max\{|z_i - z_i^0|, \epsilon\}$$

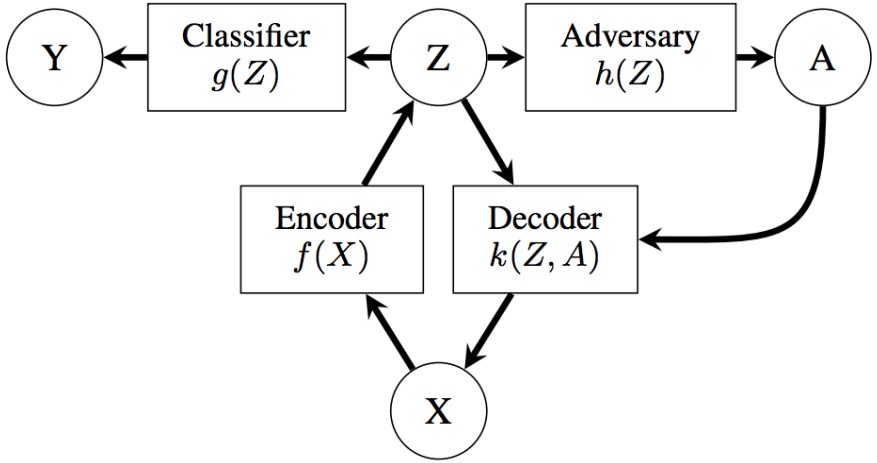
$$\mathcal{L}_2 \triangleq -\log p(c_\phi(g_\theta(z, y_{\text{source}})) = y_{\text{source}})$$

Results We found that with the perturbased-based examples attacking [16], annotators can correctly identify the adversarial images with a 92.9% success rate, but can only correctly identify adversarial examples from our attack with a 76.8% success rate. Against [17], the success rates are 87.6% and 78.2% respectively. It's important to note that we expect this gap to increase as both better generative models, and better defenses are developed, such that we may expect generative adversarial examples to eventually completely dominate the perterbation-based attacks in evading human detection.

Difference: human subject experiments

Similar work

- Learning Adversarially Fair and Transferable Representations
  - Advocate representation learning as the key to mitigating unfair prediction
  - Propose and explore adversarial representation learning as a natural method of ensuring third parties act fairly

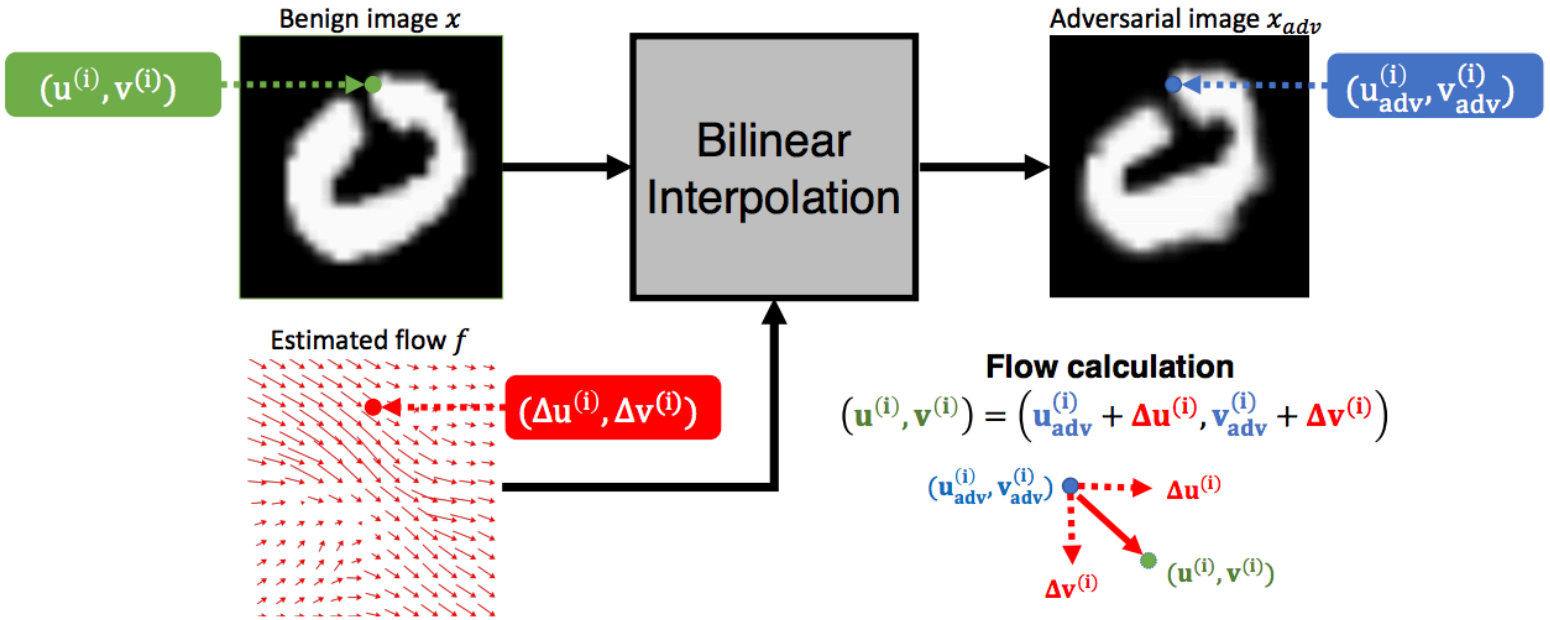


Difference: explore new the fairness of machine learning from adversarial learning aspect; nice definition of fairness and theoretic analysis

Spatially Transformed Adversarial Examples

- Realistic attacks are possible with generative models
- What if we do not directly manipulate the value of pixels?
- What else can we modify? (2D, 3D)
- Potential topic: how to attack 3D point clouds?

Spatially Transformed Adversarial Examples



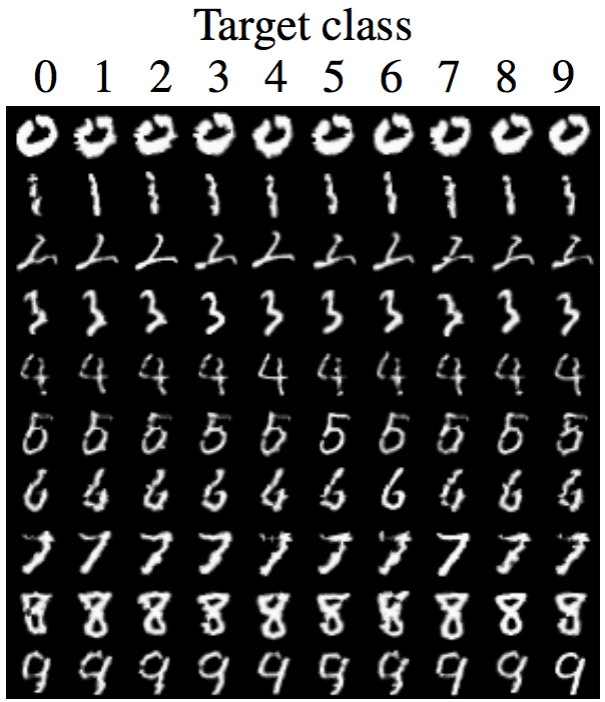
$$f^* = \underset{f}{\operatorname{argmin}} \quad \mathcal{L}_{adv}(x, f) + \tau \mathcal{L}_{flow}(f),$$

$$\mathcal{L}_{adv}(x, f) = \max(\max_{i \neq t} g(\mathbf{x}_{\text{adv}})_i - g(\mathbf{x}_{\text{adv}})_t, \kappa)$$

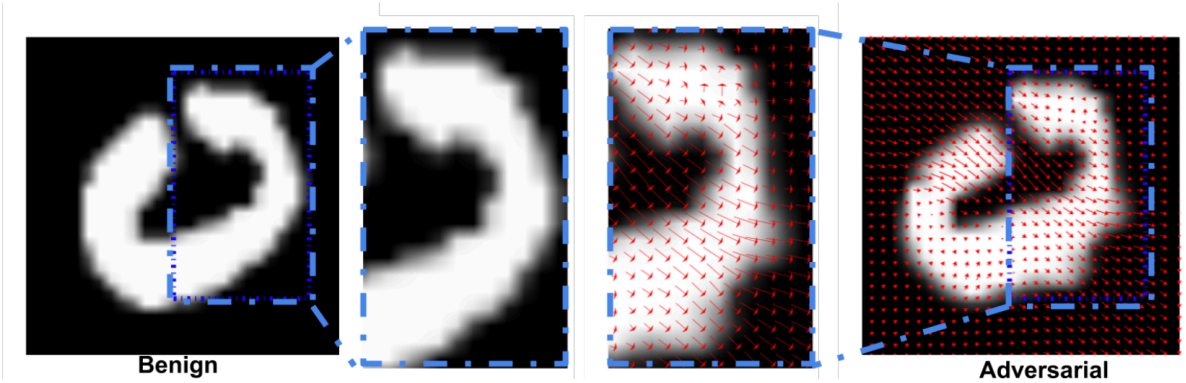
$$\mathcal{L}_{flow}(f) = \sum_{p}^{\text{all pixels}} \sum_{q \in \mathcal{N}(p)} \sqrt{||\Delta u^{(p)} - \Delta u^{(q)}||_2^2 + ||\Delta v^{(p)} - \Delta v^{(q)}||_2^2}.$$

$$\mathbf{x}^{(i)}_{\text{adv}} = \sum_{q \in \mathcal{N}(u^{(i)}, v^{(i)})} \mathbf{x}^{(q)} (1 - |u^{(i)} - u^{(q)}|)(1 - |v^{(i)} - v^{(q)}|)$$

Examples generated by stAdv



Adversarial examples generated by stAdv on MNIST
The ground truth images are shown in the diagonal



Flow visualization on MNIST. The digit "0'' is misclassified as "2''.

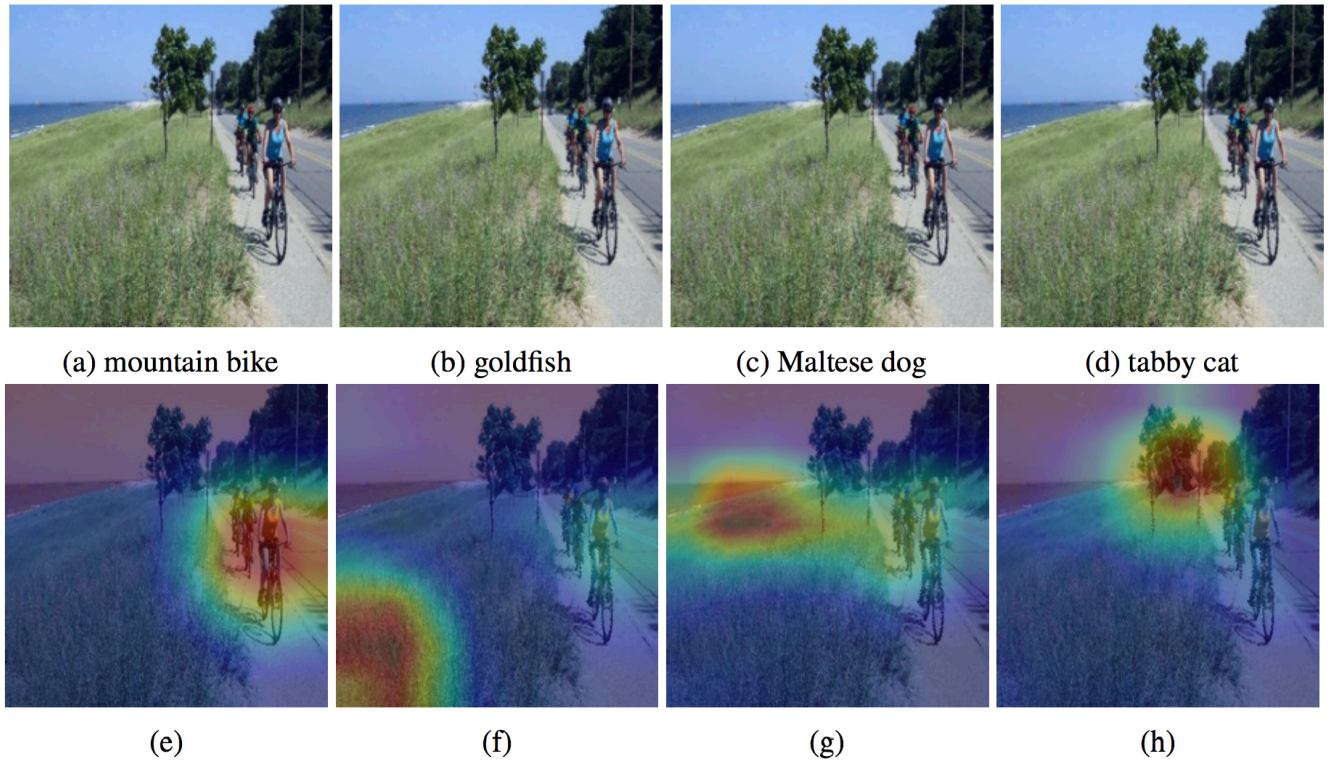
Attack Effectiveness Under Defenses

Model	Def.	FGSM	C&W.	stAdv
A	Adv.	4.3%	4.6%	**32.62%**
	Ens.	1.6%	4.2%	**48.07%**
	PGD	4.4%	2.96%	**48.38%**
B	Adv.	6.0%	4.5%	**50.17%**
	Ens.	2.7%	3.18%	**46.14%**
	PGD	9.0%	3.0%	**49.82%**
C	Adv.	3.22%	0.86%	**30.44%**
	Ens.	1.45%	0.98%	**28.82%**
	PGD	2.1%	0.98%	**28.13%**

Model	Def.	FGSM	C&W.	stAdv
ResNet32	Adv.	13.10%	11.9%	**43.36%**
	Ens.	10.00%	10.3%	**36.89%**
	PGD	22.8%	21.4%	**49.19%**
wide ResNet34	Adv.	5.04%	7.61%	**31.66%**
	Ens.	4.65%	8.43%	**29.56%**
	PGD	14.9%	13.90%	**31.6%**

Attack success rate of adversarial examples generated by stAdv against different models under standard defense on MNIST and CIFAR-10

Attention of Networks



CAM attention visualization for ImageNet inception_v3 model. (a) the original image and (b)-(d) are stAdv adversarial examples targeting different classes. Row 2 shows the attention visualization for the corresponding images above.

inception_v3 model

(a) Benign (b) FGSM (c) C&W (d) StAdv

Adversarial trained inception_v3 model

(e) Benign (f) FGSM (g) C&W (h) StAdv

CAM attention visualization for ImageNet inception_v3 model. Column 1 shows the CAM map corresponding to the original image. Column 2-4 show the adversarial examples generated by different methods. (a) and (e)-(g) are labeled as the ground truth "cinema", while (b)-(d) and (h) are labeled as the adversarial target "missile."

Takeaways

- Instead of manipulating the pixel values, we can also move the position of pixels to generate adversarial examples for 2D images
- For 3D, you can add points, what else?
- It is impossible to tell/detect adversarial perturbation from network attention
- A lot of diverse adversarial examples can be explored

Potential Final Project Topics

- Attacks against general machine learning models such as 3D reconstruction, BERT, and RL systems.
- Detection against attacks such as Deepfake.
- GWAS for AI
- Theoretically understanding of generative models from the game theoretic perspective
- Applications of GANs (GAN Zoo)
- Provable robustness for classifiers against different types of perturbation
- Differential private graphs, and robust graph neural networks
- Privacy analysis for generative models
- Robust reinforcement learning
- Improve model robustness with unlabeled data via semi-supervised learning
- Robustness testing for different deep neural networks architectures
- Robust autoML
- Semantic Forensics
- Design an ensemble model which guarantees the diversity of the individual classifiers and therefore improve robustness