Image recognition: Defense adversarial attacks using Generative Adversarial Network (GAN)

Speaker: Guofei Pang

Division of Applied Mathematics
Brown University

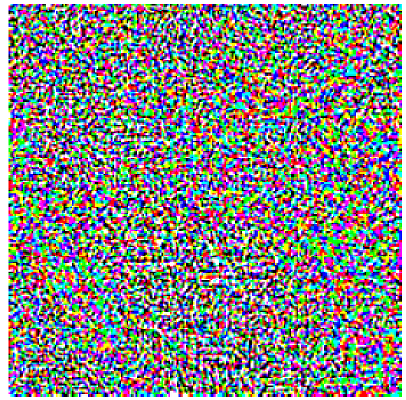
Presentation after reading the paper:
Ilyas, Andrew, et al. "The Robust Manifold Defense: Adversarial Training using Generative Models." arXiv preprint arXiv:1712.09196 (2017).

Outline

- Adversarial attacks
- Generative Adversarial Network (GAN)
- How to defense attacks using GAN
- Numerical results

Adversarial Attacks

$+ .007 \times$



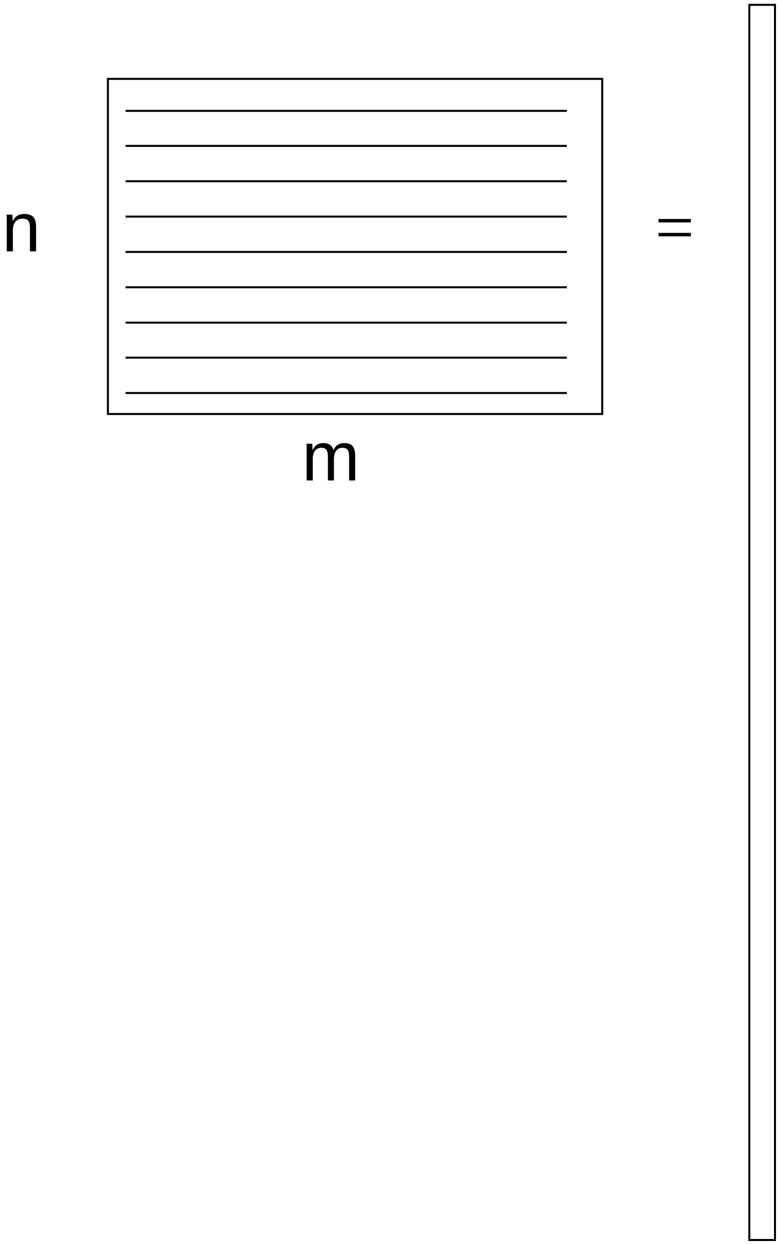
$=$

$\boldsymbol{x}$	$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$	$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"panda"	"nematode"	"gibbon"
57.7% confidence	8.2% confidence	99.3 % confidence

Adversarial Attacks

P(man) > 0.99

P(woman) > 0.99



n*m

Image as a vector:

$$\mathbf{x} = \{\mathbf{x_j}\}, \quad \mathbf{j} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{n} * \mathbf{m}$$

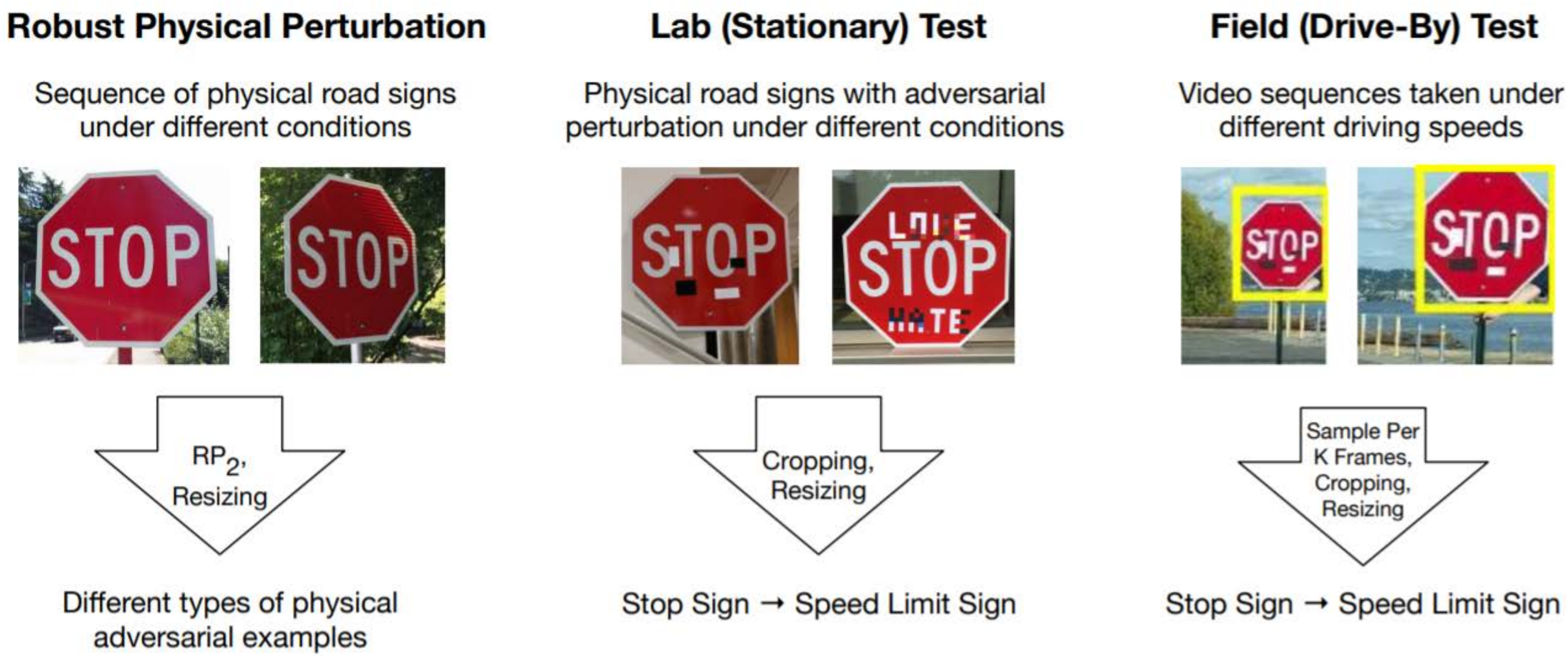
Adversarial Attacks

$$||\mathbf{x_1} - \mathbf{x_2}||_2 < \mathbf{e0} \quad \Longrightarrow \quad |\mathbf{C(x_1) - C(x2)}| > \mathbf{f0}$$

Adversarial examples for a classifier C():

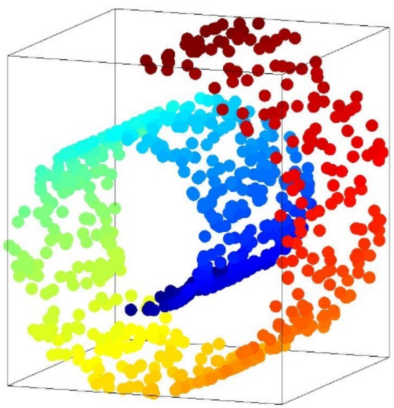
- A pair of input x_1 and x_2
- A person says they are of the same class
- But a classifier will they are completely different!

Adversarial Attacks



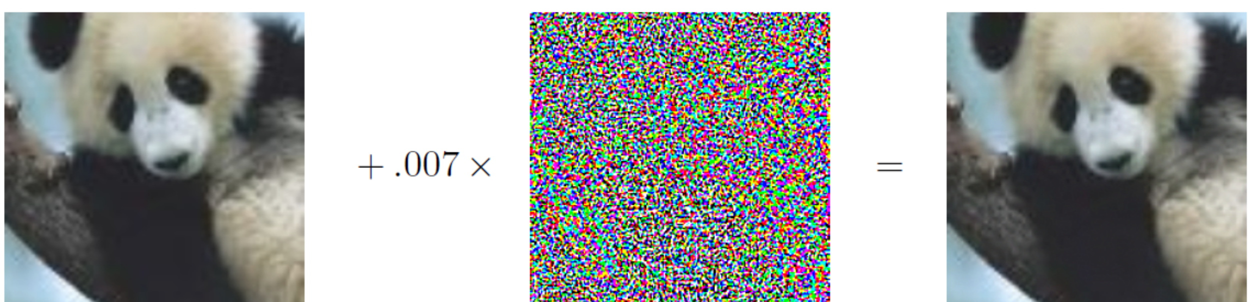
Why does classifier become fool for these examples?

Why does classifier become fool for these examples?



An intuition from the authors:

- Natural image: Low-dimensional manifold
- Noisy image: High-dimensional manifold
- High dimensionality is tough for classifier.



Generative adversarial network (GAN)

- **x and x’ have similar PDF**
- **G() has learned the underlying distribution of image dataset after training GAN**
- **The DNN G() is a nonlinear mapping from low-dimensional space, z, to high-dimensional space, x’**

Original image **x**

GAN

Synthetic image /Generative model

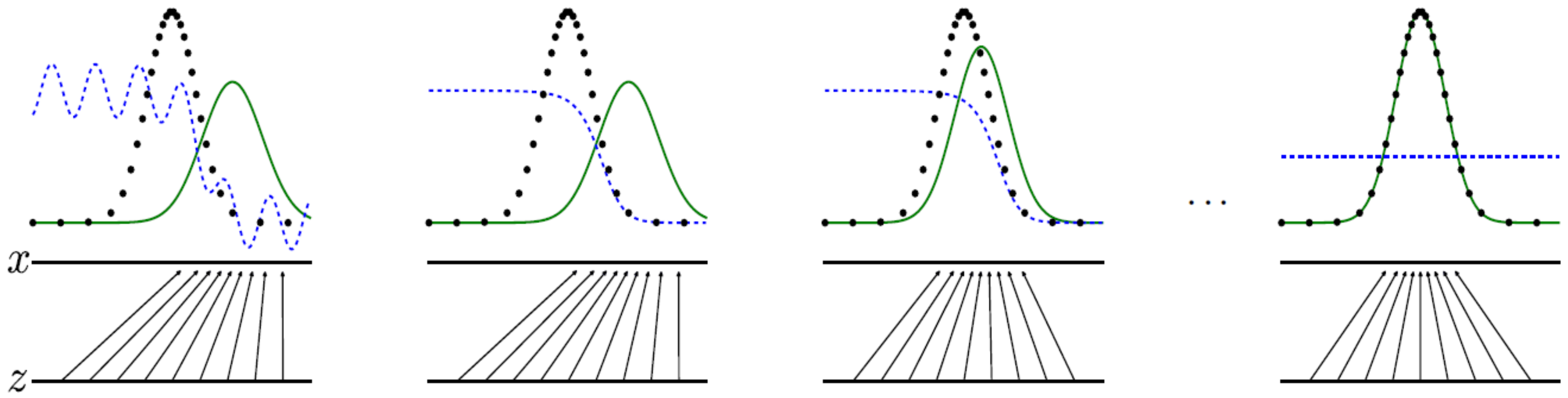
x’=G(z)

Generator **G(**z**)**

Noisy input z, say, z – N(0,**I**)

Generative adversarial network (GAN)

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$



- **Convergence state: $p_{data}(x)=p_G(x)$**
- Green solid line: probability density function (PDF) of the generator G()
- Black dotted line: PDF of original image x, i.e., $p_{data}(x)$
- Blue dash line: PDF of discriminator D()

Generative adversarial network (GAN)

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

How to defense attacks using GAN

- **G() is pre-trained and has learned the underlying distribution of the training (image) dataset after training GAN**

Invert and Classify

Synthetic image **x'=G(**z*)
(**Preserve low-dimensional manifold**)

Classifier C()

Original image **x**
(Could include high-dimensional manifold when noise enters)

$$z^* = \arg\min_z \|G(z) - x\|_2$$

- **G() is pre-trained and has learned the underlying distribution of the training (image) dataset after training GAN**

Enhanced Invert and Classify

Synthetic image **x'=G(**z*)
(Preserve low-dimensional manifold)

Classifier C()
(**retrain the classifier**)

Upper bound of attack magnitude

Classification loss

$$\inf_{\theta} \mu \boxed{\left(\sup_{z,z'} \| C_\theta(G(z)) - C_\theta(G(z')) \|_2^2 \right)} + (1-\mu) \boxed{\left(\frac{1}{N} \sum_{i=1}^{N} f\left(y^{(i)}, C_\theta(x^{(i)}) \right) \right)}$$
$$\text{s.t. } \|G(z) - G(z')\|_2^2 \le \eta^2.$$

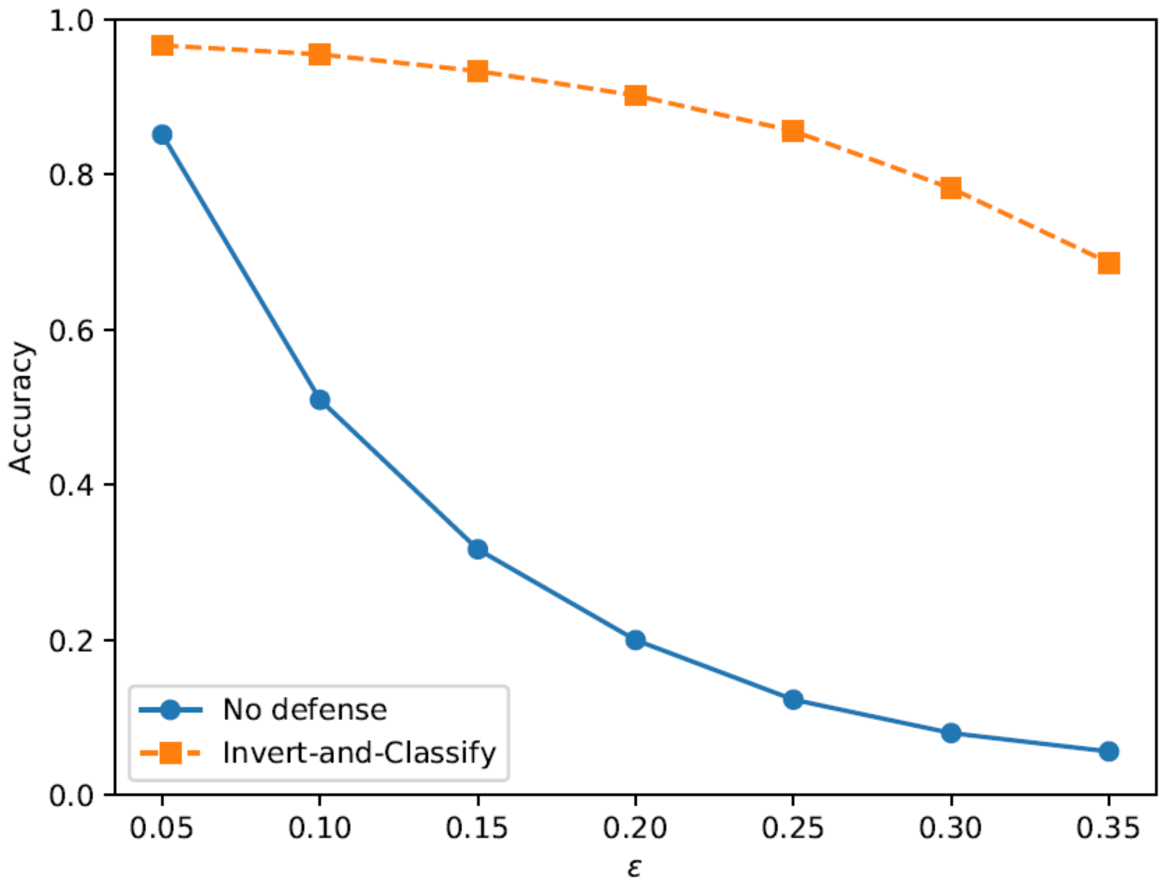
$$z^* = \arg\min_z \|G(z) - x\|_2$$

First-order classifier attacks for handwritten digit classification

$$\delta = +\epsilon \cdot \mathrm{sign}(\nabla_X L(y, C_\theta(X))|_{X=x}),$$

Numerical results

First-order classifier attacks for handwritten digit classification



$$\delta = +\epsilon \cdot \mathrm{sign}(\nabla_X L(y, C_\theta(X))|_{X=x}),$$

Numerical results

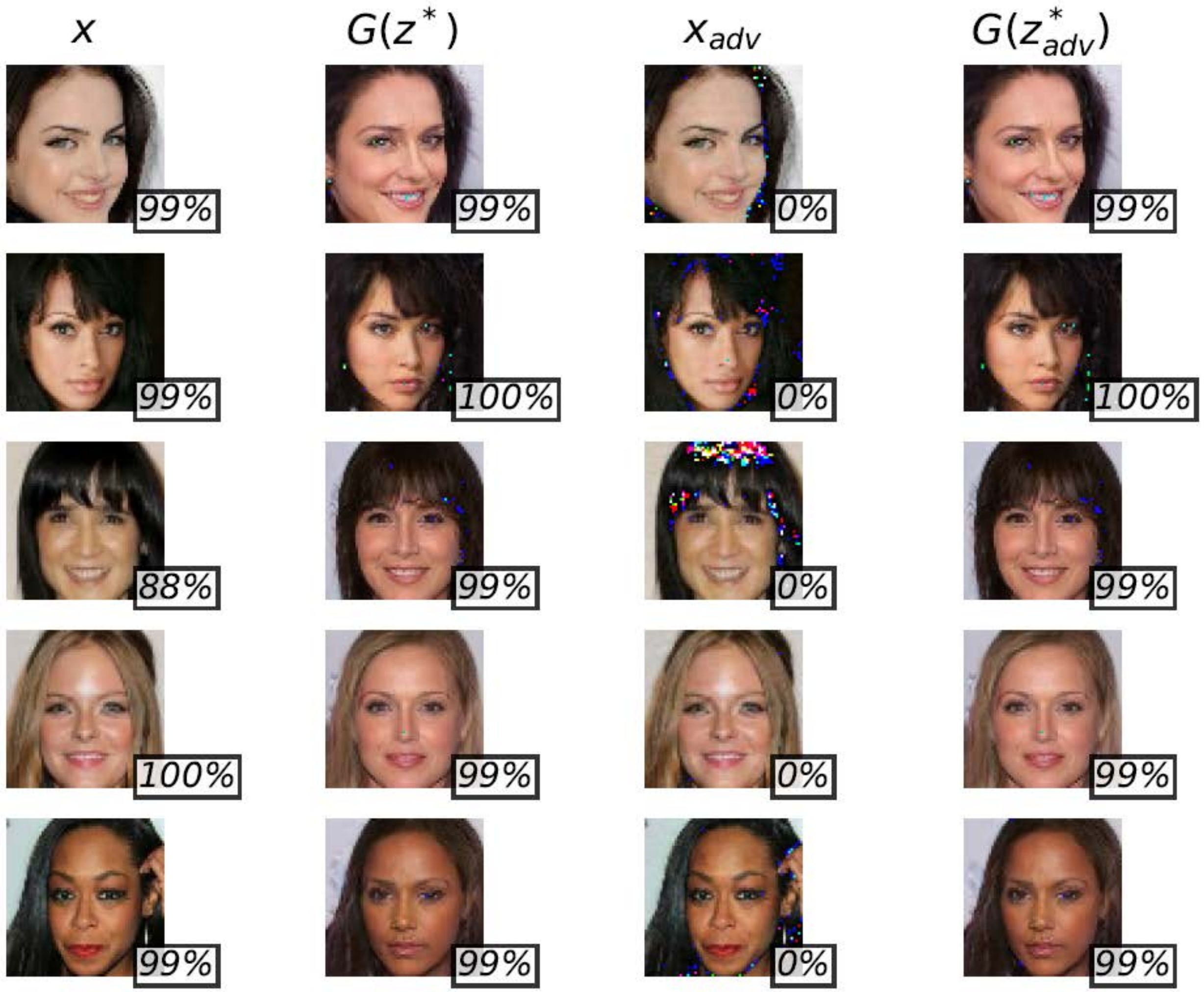
First-order classifier attacks for handwritten digit classification

ϵ	No defense	Invert and Classify
Clean Data	97%	84%
FGSM ($\epsilon = 0.05$)	1%	82%
FGSM ($\epsilon = 0.1$)	0%	80%
FGSM ($\epsilon = 0.2$)	0%	73%
Carlini-Wagner ℓ_2	0%	77%
Carlini-Wagner ℓ_0	0%	65%
Carlini-Wagner ℓ_∞	0%	66%

$$\delta = +\epsilon \cdot \mathrm{sign}(\nabla_X L(y, C_\theta(X))|_{X=x}),$$

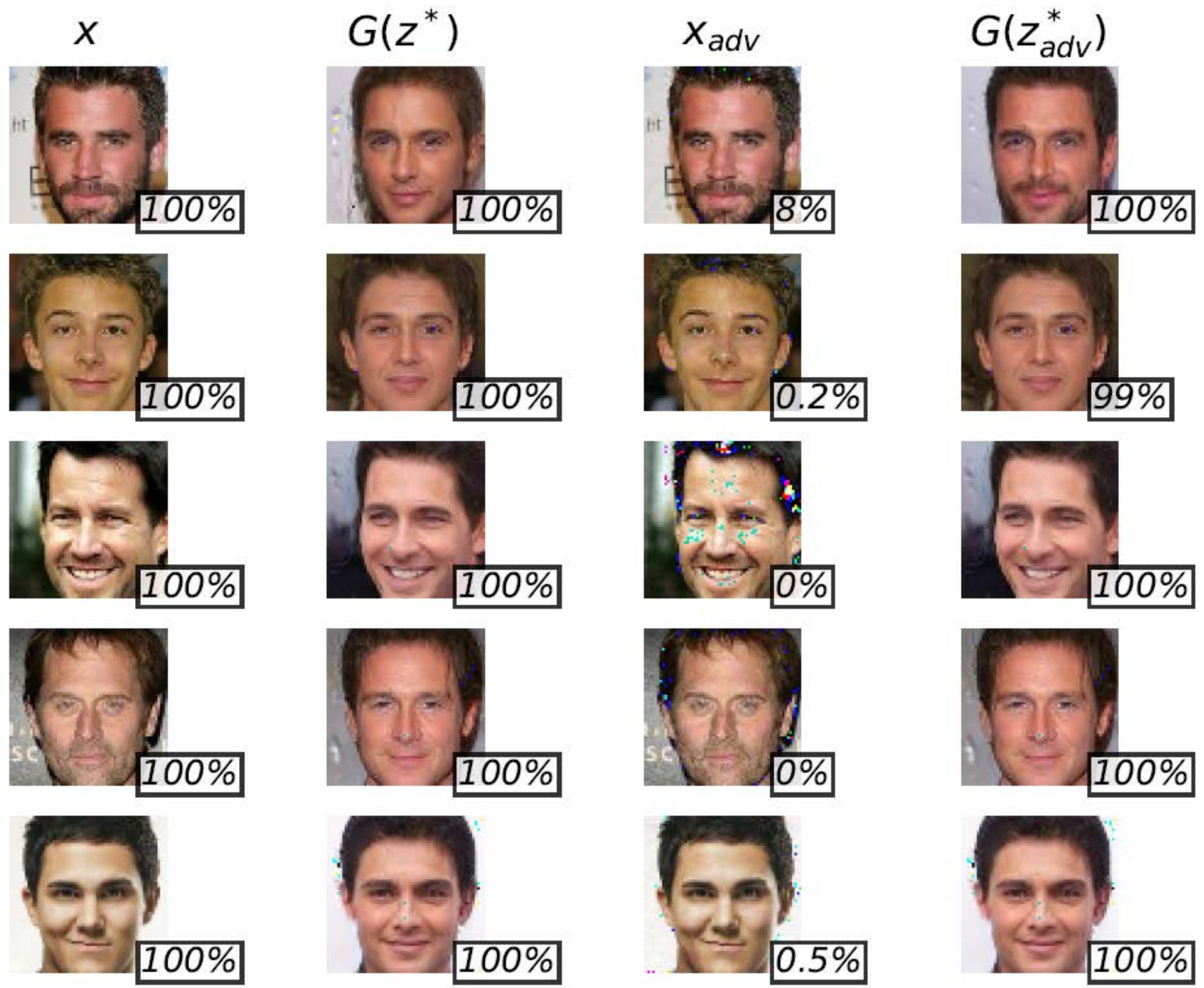
Numerical results

First-order classifier attacks for gender classification



Numerical results

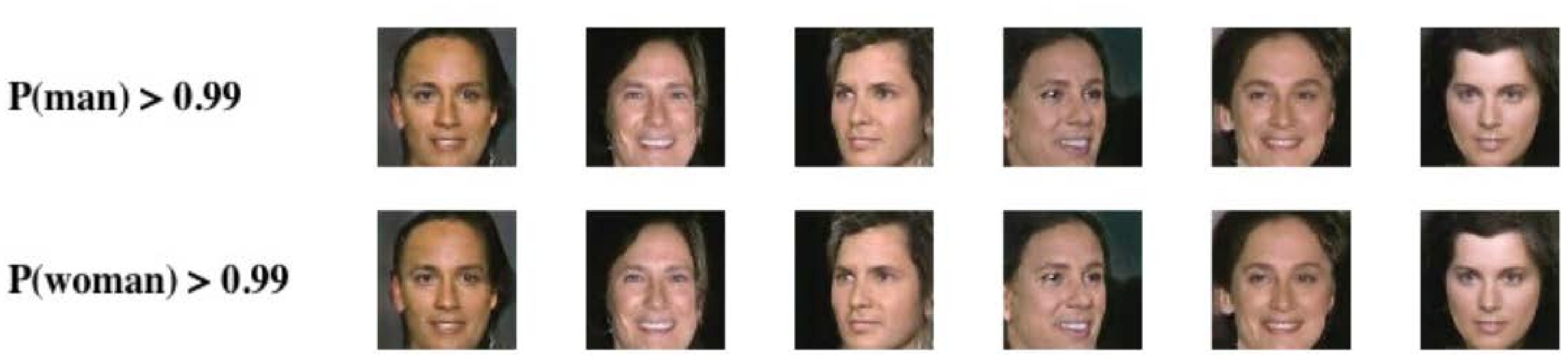
First-order classifier attacks for gender classification



Numerical results

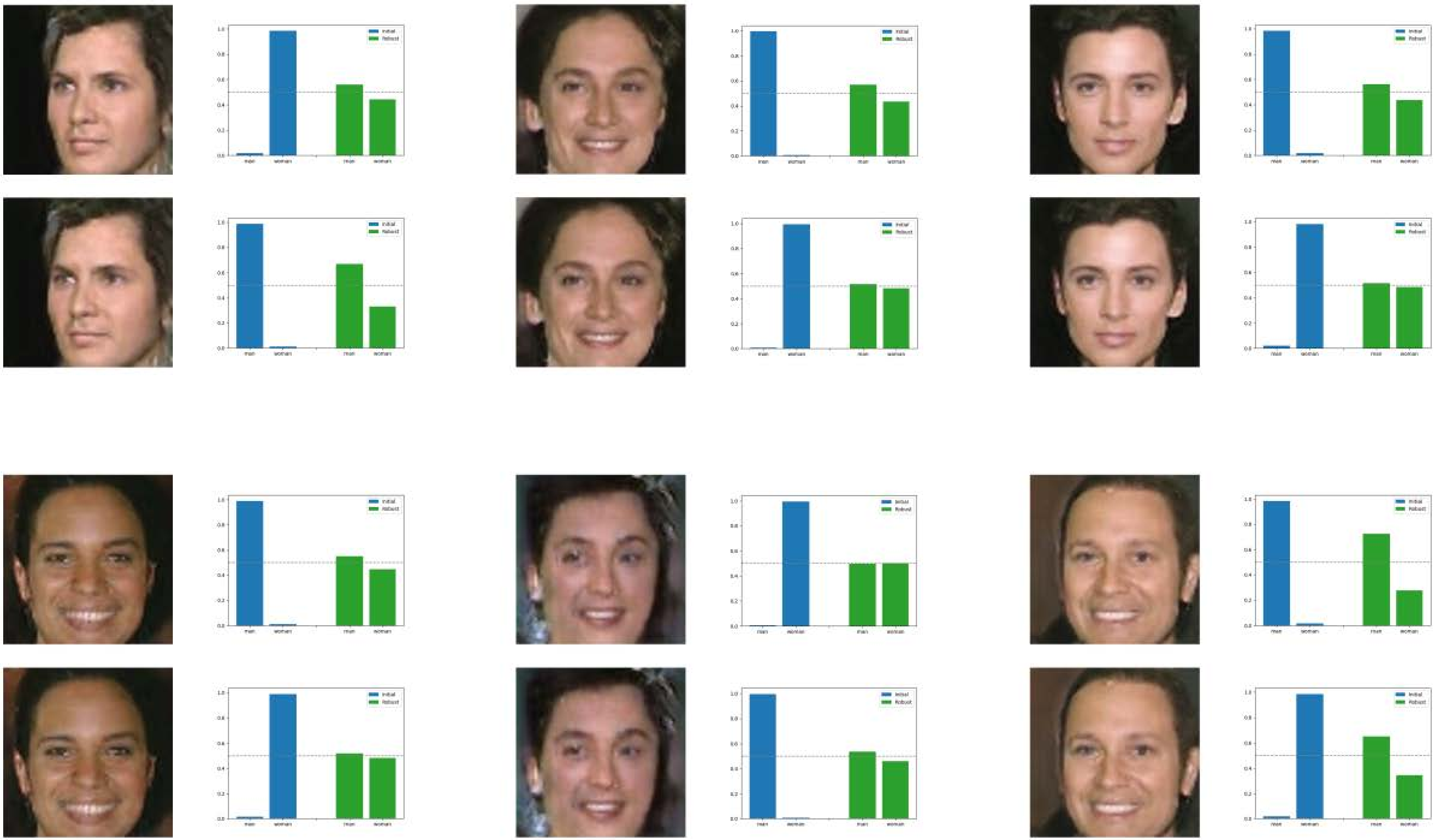
Substitute model attacks

Results from Invert and Classify

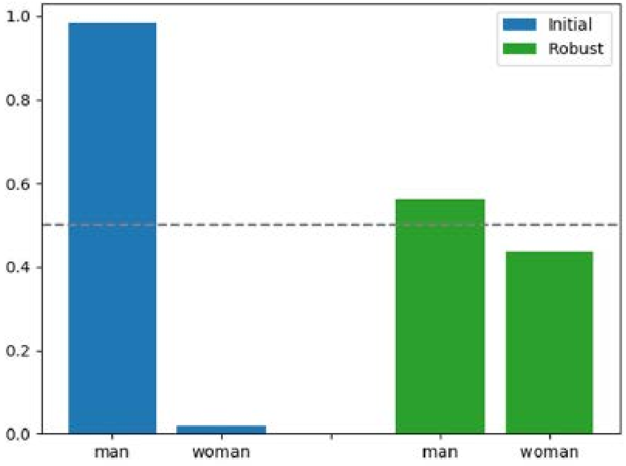


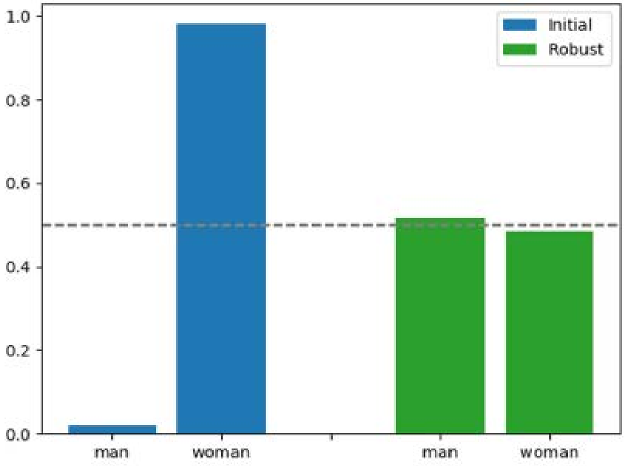
Comparison between
Invert and Classify and Enhanced Invert and Classify

Numerical results

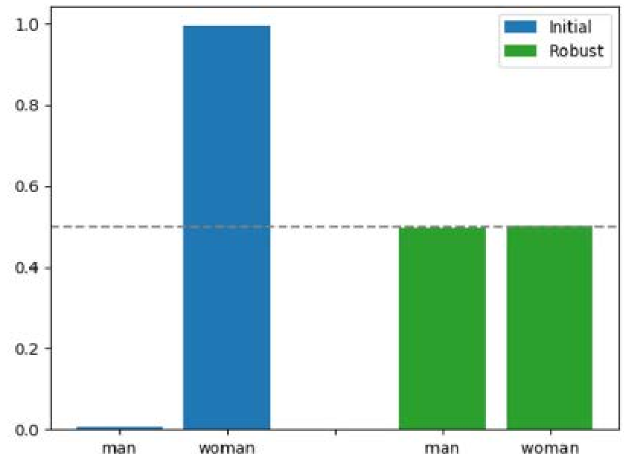


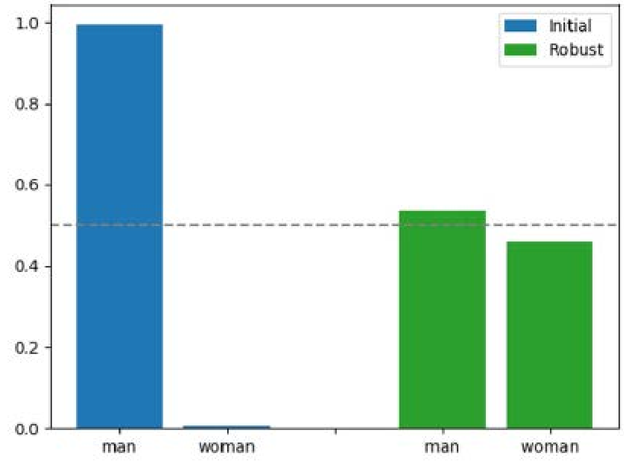
Numerical results



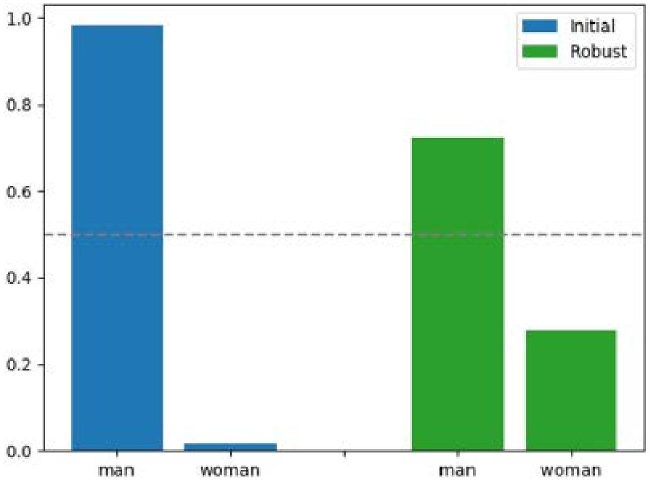


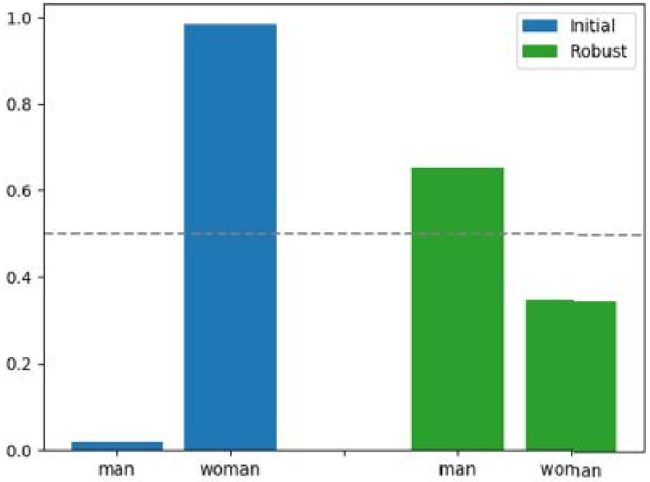
Numerical results





Numerical results





- GAN for regression problems?
- GAN versus other neural networks?
- One defense strategy for all types of attacks?