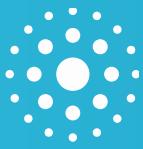




Master DSAI *More Deep Learning* 2023-2024

*Adversarial Examples, Generative Adversarial Networks
(GANs)*

Frederic Precioso



ADVERSARIAL EXAMPLES



Morphing



Amazing but...be careful of the adversaries (as any other ML algorithms)

Intriguing properties of neural networks

C. Szegedy, w. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I.

Goodfellow, R. Fergus

arXiv preprint arXiv:1312.6199

2013

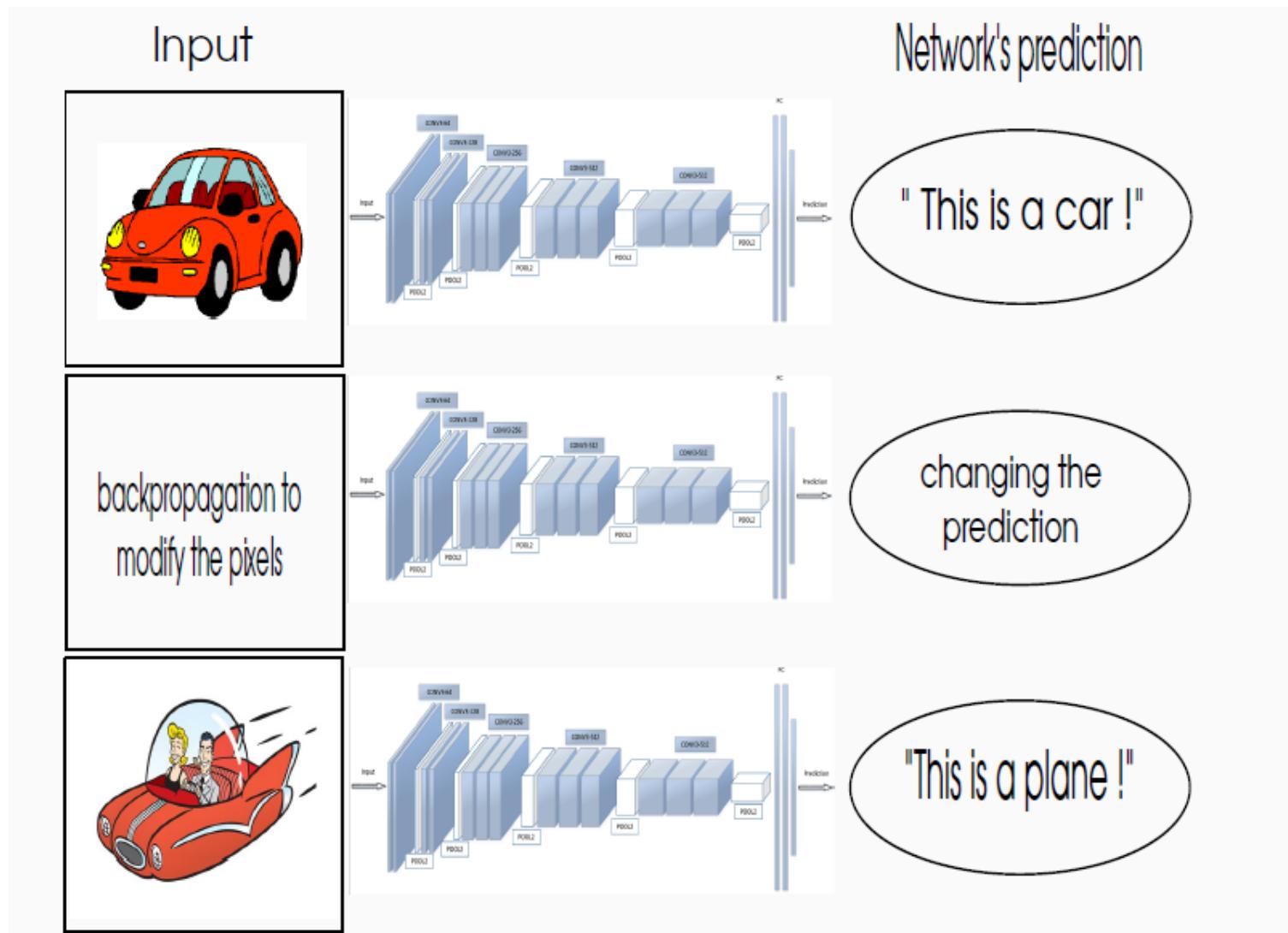
[1312.6199] Intriguing properties of neural networks - arXiv.org

<https://arxiv.org/abs/1312.6199> › cs - Traduire cette page

de C Szegedy - 2013 - Cité 449 fois - Autres articles

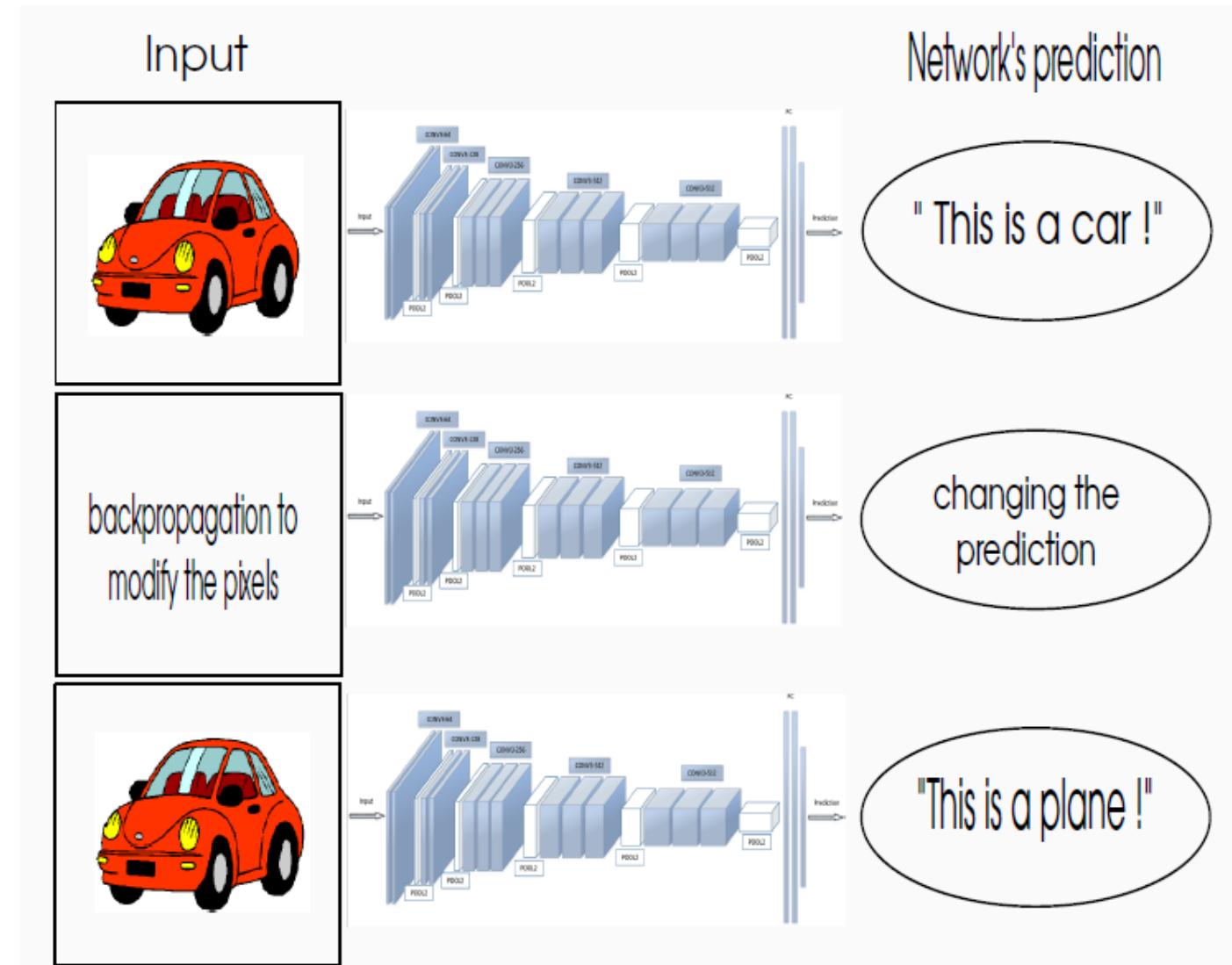
21 déc. 2013 - In this paper we report two such **properties**. First, we ... Second, we find that deep **neural networks** learn input-output mappings that are fairly ...

Amazing but...be careful of the adversaries (as any other ML algorithms)





Amazing but...be careful of the adversaries (as any other ML algorithms)



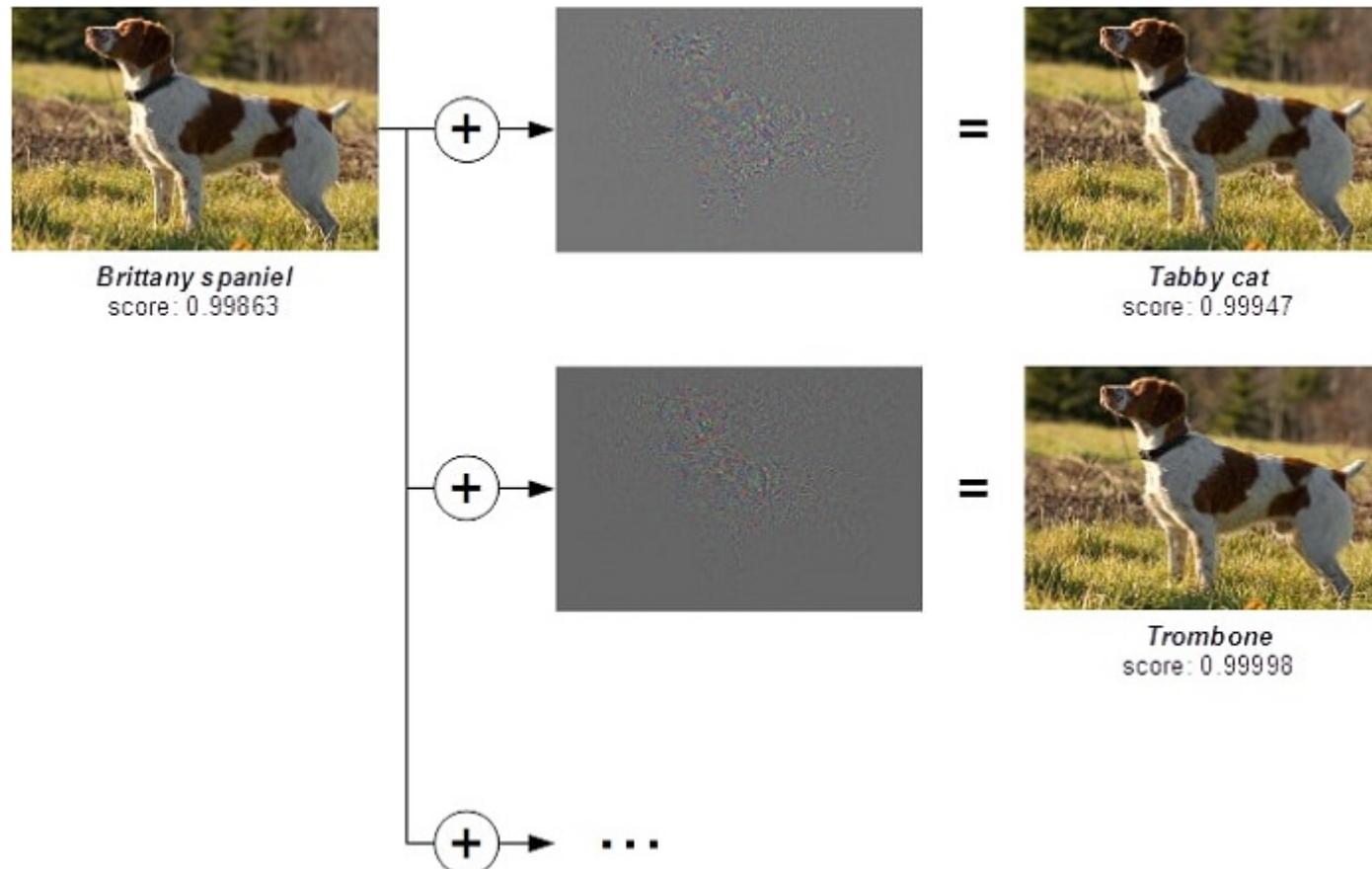


Amazing but...be careful of the adversaries (as any other ML algorithms)

$$\begin{array}{ccc} \text{panda} & + .007 \times & \text{nematode} \\ \mathbf{x} & & \text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y)) \\ & & 8.2\% \text{ confidence} \\ \text{"panda"} & & \mathbf{x} + \\ 57.7\% \text{ confidence} & & \text{esign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y)) \\ & & \text{"gibbon"} \\ & & 99.3 \% \text{ confidence} \end{array}$$

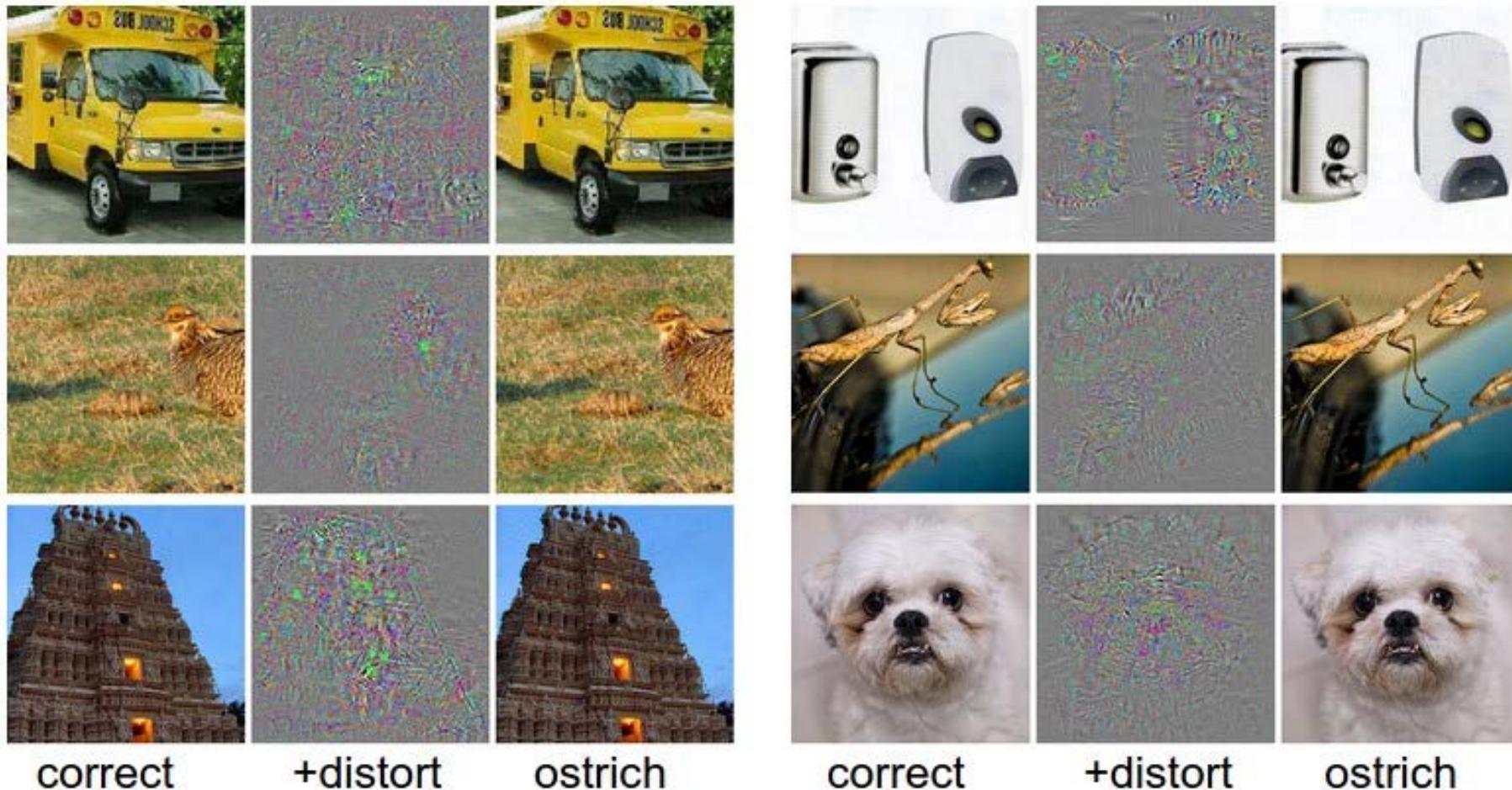
Andrey Karpathy blog, <http://karpathy.github.io/2015/03/30/breaking-convnets/>

Amazing but...be careful of the adversaries (as any other ML algorithms)



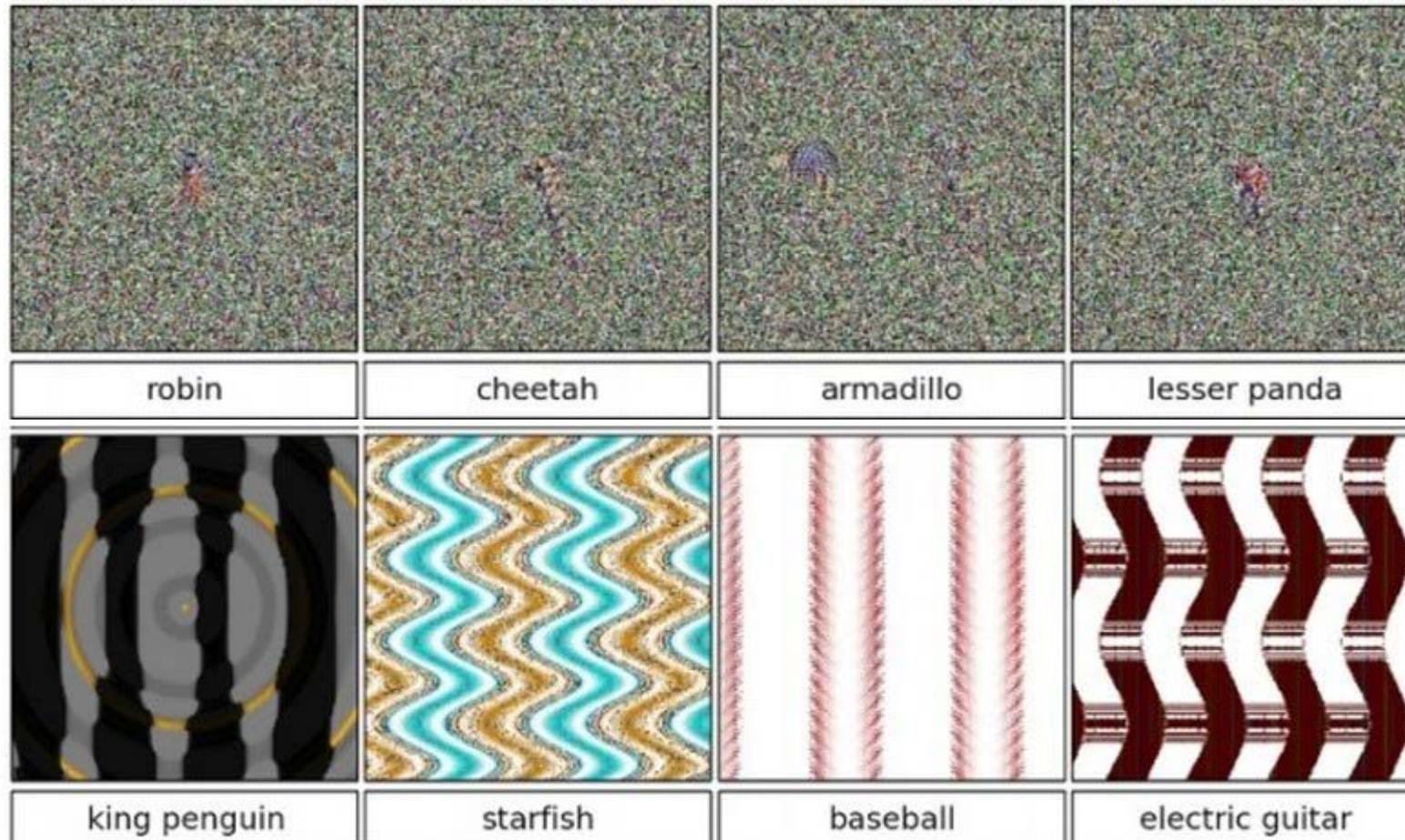
From Thomas Tanay

Amazing but...be careful of the adversaries (as any other ML algorithms)





Amazing but...be careful of the adversaries (as any other ML algorithms)



Adversarial example: Definition

Definition: \hat{x} is called adversarial iff:

- given image x
- low distortion $\|x - \hat{x}\| < \epsilon$, ($\epsilon > 0$, few pixels)
- given network's probabilities $f_\theta(x)$
- **Different predictions!** $\text{argmax}f_\theta(x) \neq \text{argmax}f_\theta(\hat{x})$

Adversarial examples: Algorithms

The Fast Gradient Sign Method attack (FGSM)[Goodfellow 2015]:

The FGSM is a single step attack, i.e., the perturbation is added in a single-step instead of adding it over a loop (Iterative attack). Essentially, FGSM computes the gradients of a loss function with respect to the input image and then uses the sign of the gradients to create a new image (i.e., the adversarial image) that maximizes the loss. Mathematically, the attacking scheme can be demonstrated as:

$$adv_x = x + \epsilon * sign(\nabla_x J(\theta, x, y))$$

where:

adv_x : Our output adversarial image

x : The original input image

y : The ground-truth label of the input image

ϵ : Small value we multiply the signed gradients by to ensure the perturbations are small enough that the human eye cannot detect them but large enough that they fool the neural network

θ : Our neural network model

J : The loss function

Adversarial examples: Algorithms

The Basic Iterative Method attack (BIM)[Kurakin 2017]: The BIM applying the same step as FGSM multiple times with small step size and clip the pixel values of intermediate results after each step to ensure that they are in an ϵ -neighborhood of the original image, i.e., within the range $[X_{ij} - \epsilon, X_{ij} + \epsilon]$, X_{ij} being the pixel value of the previous image. This method is also referred to as Iterative-FGSM or IFGSM. Mathematically, the attacking scheme can be demonstrated as:

$$\stackrel{dv}{\tau}, y_{true})) \} \quad X_0^{adv} = X, \quad X_{N+1}^{adv} = Clip_{X, \epsilon} \{ X_N^{adv} + \alpha sign(\nabla_X J(X_N^a)) \}$$

where:

X : Clean input image

X^{adv} : Adversarial Image at i^{th} step

J : Loss function

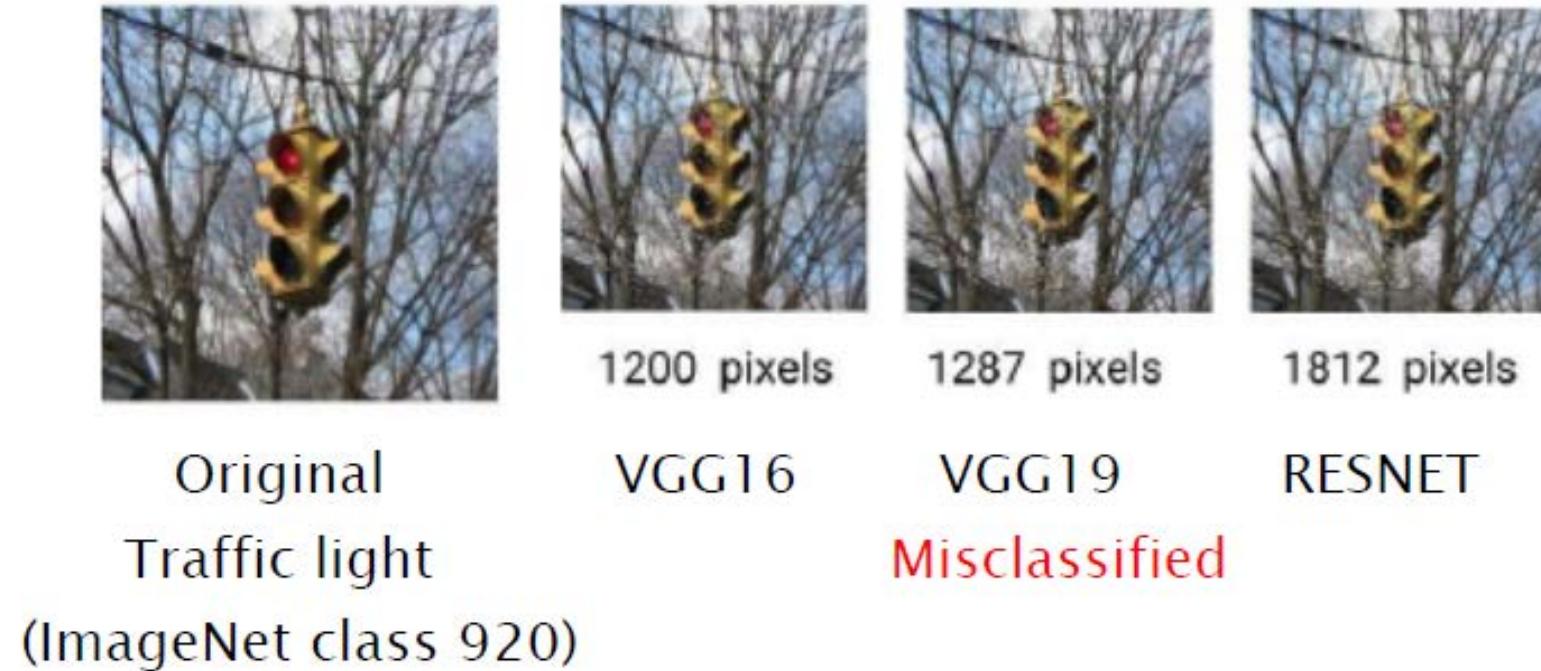
y_{true} : Model output for x

ϵ : Tunable parameter

α : Step size



Amazing but...be careful of the adversaries (as any other ML algorithms)



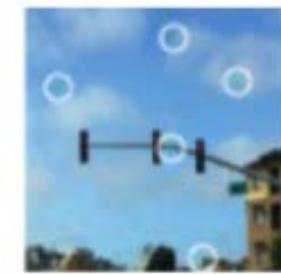
State-of-the art deep neural networks on ImageNet



Amazing but...be careful of the adversaries (as any other ML algorithms)



Red Light Modified to
Green after 18 white pixels.
Probability: 59%



Red Light Modified to
Green after 9 green pixels.
Probability: 50.9%



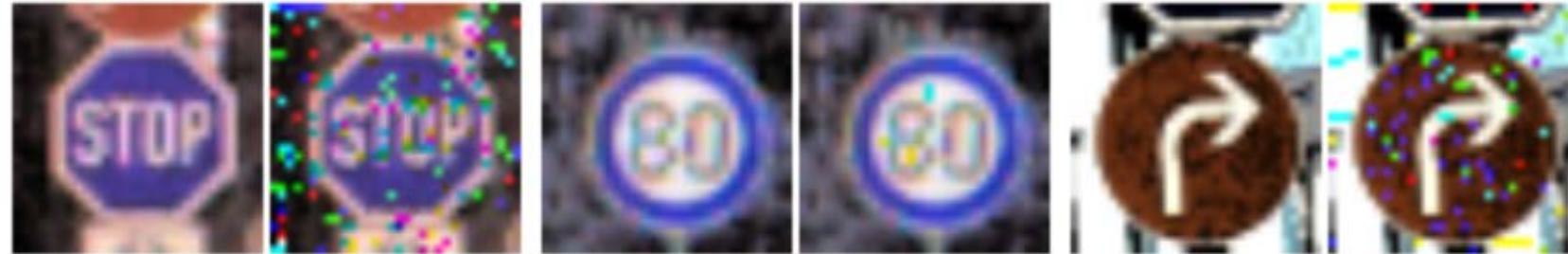
Red Light Modified to
Green after 9 green pixels.
Probability: 53%



No Light Modified to Green
after 4 green pixels.
Probability: 51.9%



Amazing but...be careful of the adversaries (as any other ML algorithms)



stop

30m
speed
limit80m
speed
limit30m
speed
limitgo
rightgo
straight

Confidence 0.999964

0.99



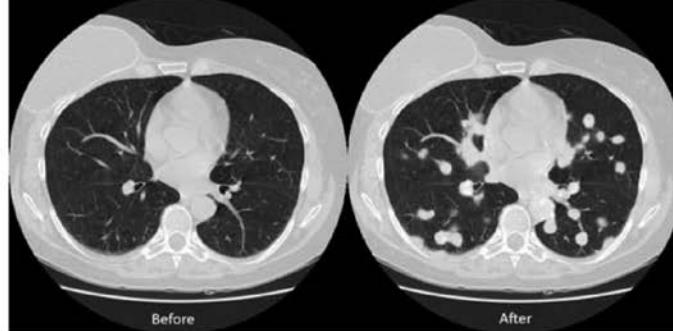
Amazing but...be careful of the adversaries (as any other ML algorithms)

SUBSCRIBE TOPIC INDEX Current Issue Digital Editions Article Archive eNewsletter Product Directories Events Jobs

home | subscribe | comment | resources | reprints | writers' guidelines

News

HACKERS CAN FOOL RADIOLOGISTS AND AI SOFTWARE BY MANIPULATING LUNG CANCER SCANS



Before After

Hackers can access a patient's 3D medical scans to add or remove malignant lung cancer and deceive both radiologists and AI algorithms that are used to aid diagnosis, according to a [new study](#) published by Ben-Gurion University (BGU) of the Negev cybersecurity researchers. [Click here](#) for a video of the attack.

A 3D CT scan combines a series of X-ray images taken from different angles around the body and uses computer processing to create cross sectional slices of the bones, blood vessels, and soft tissues. CT images provide more detailed information than standard X-rays and are used to diagnose cancer, heart disease, infectious diseases, and more. An MRI scan is similar, but uses powerful magnetic fields instead of ionizing radiation to diagnose bone, joint, ligament, and cartilage conditions.

Malicious attackers can tamper with the scans to deliberately cause a misdiagnosis for insurance fraud, ransomware, cyberterrorism, or even murder. Attackers can even automate the entire process in a malware that can infect a hospital's network.

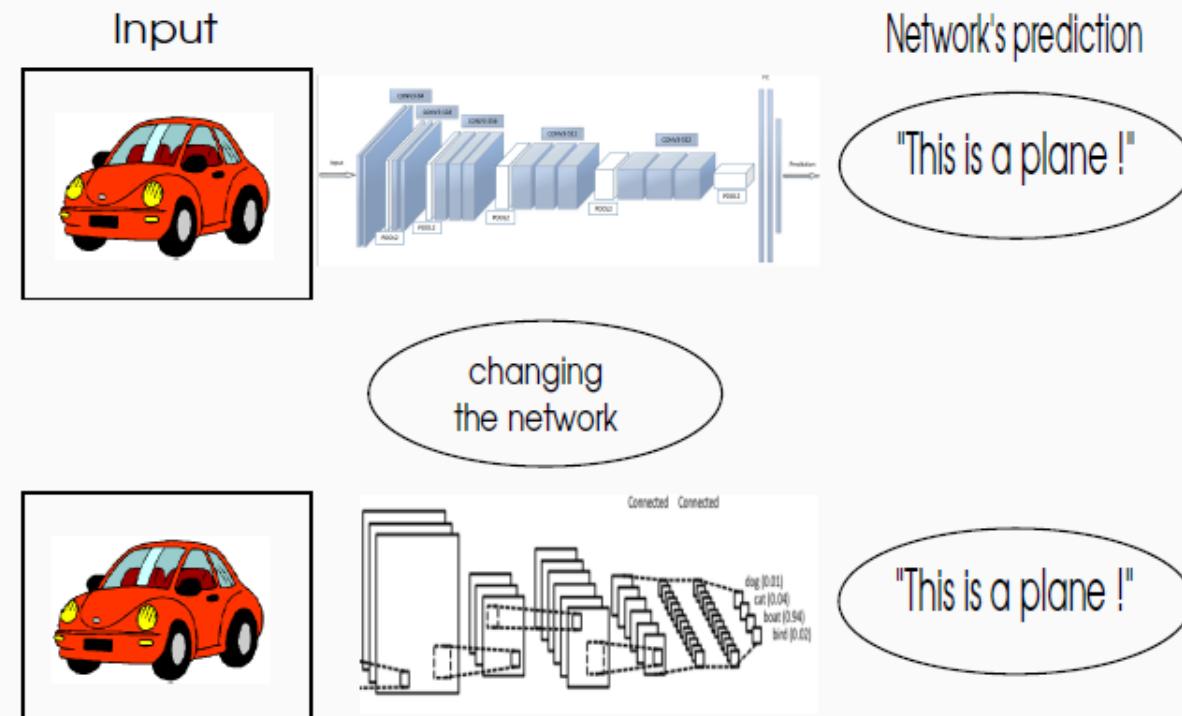
"Our research shows how an attacker can realistically add or remove medical conditions from CT and MRI scans," says Yisroel Mirsky, PhD, lead researcher in the BGU department of software and information systems engineering and project manager and cybersecurity researcher at BGU's National Cyber Security Research Center. "In particular we show how easily an attacker can

Tweets by @RadiologyToday



Amazing but...be careful of the adversaries (as any other ML algorithms)

- \neq outliers
- regularization: correct one... find another
- high confidence predictions
- Transferability



Amazing but...be careful of the adversaries (as any other ML algorithms)

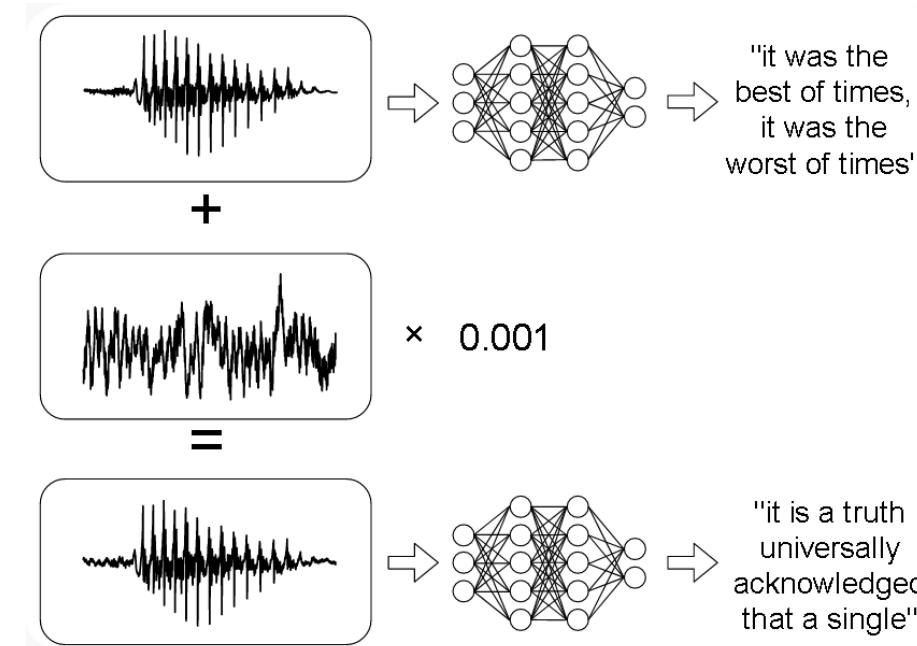
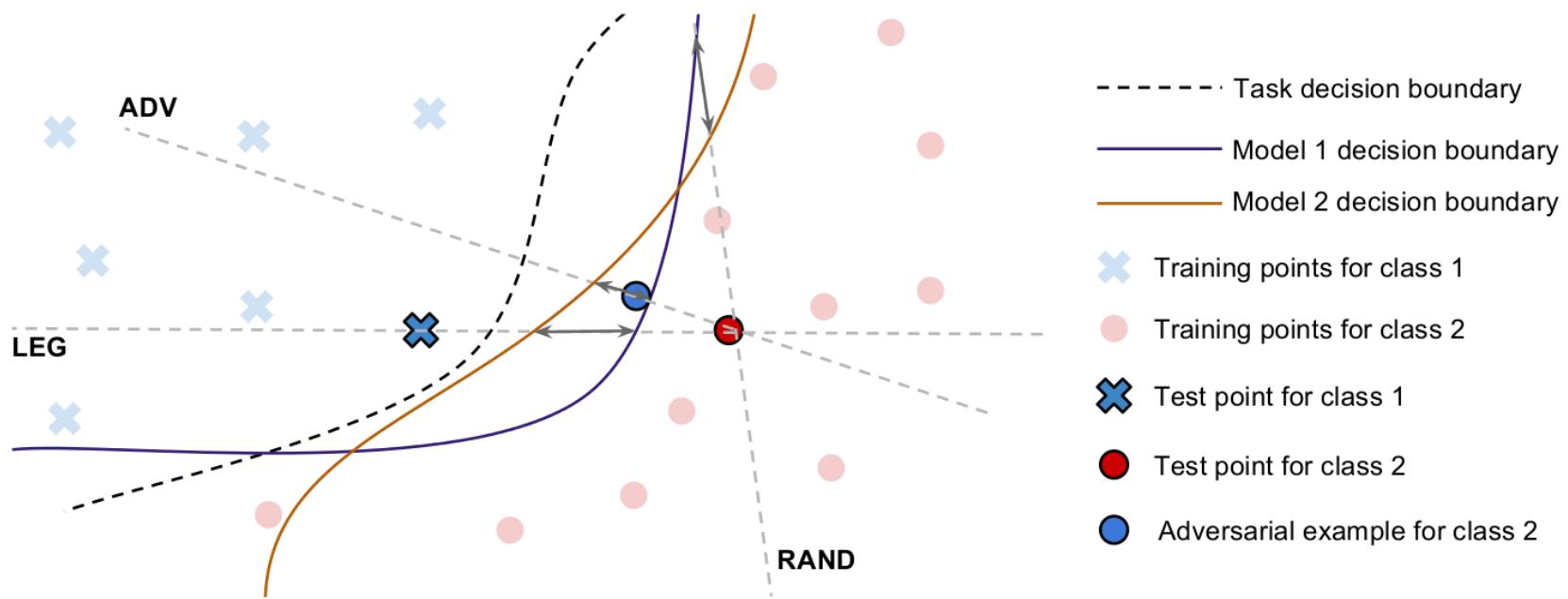


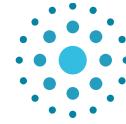
Figure from our paper: given any waveform, we can modify it slightly to produce another (similar) waveform that transcribes as any different target phrase.

https://nicholas.carlini.com/code/audio_adversarial_examples/

Adversarial examples...



Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2017).
The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*.

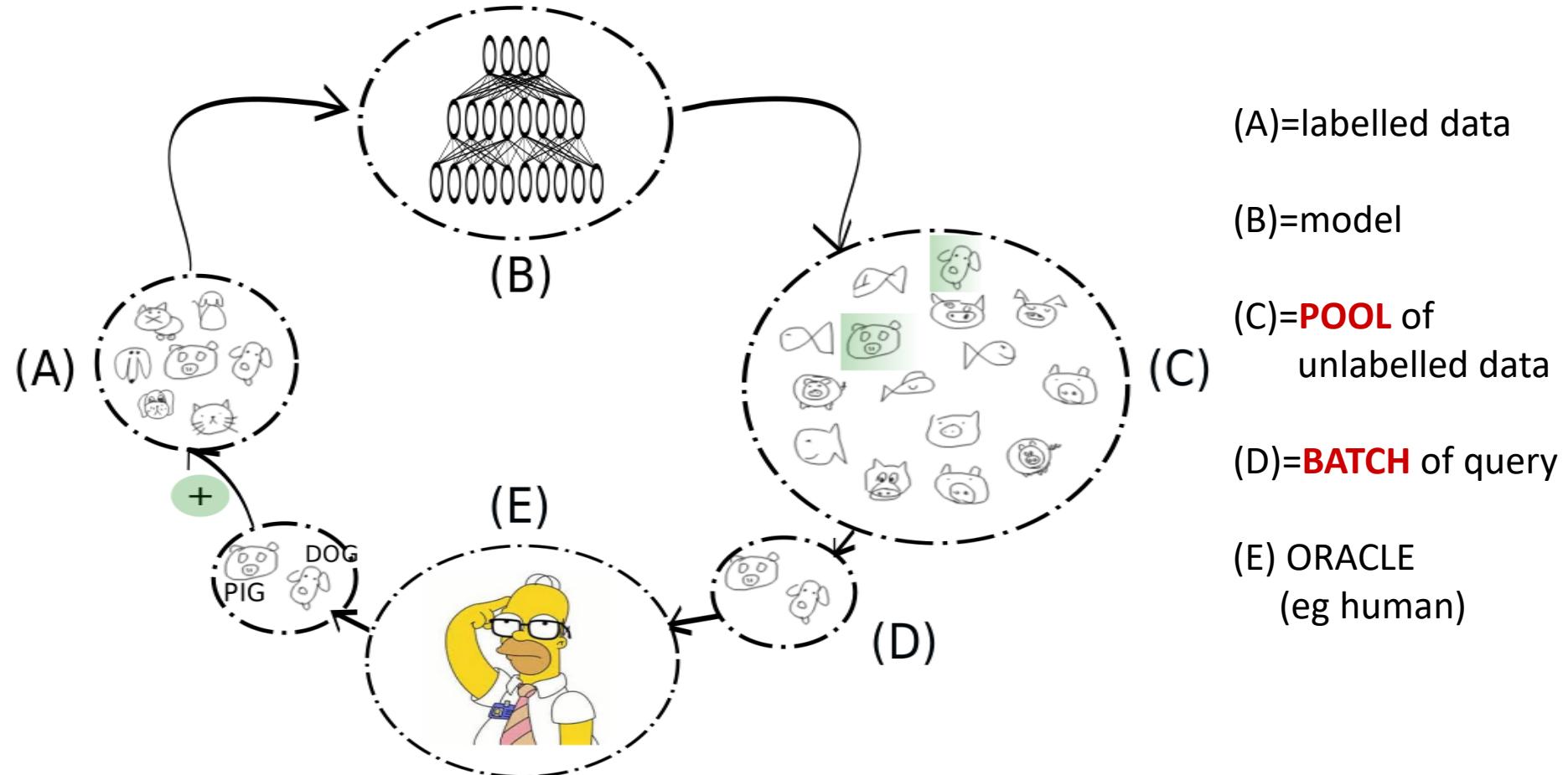


Can we benefit from adversarial examples and all their properties ?



ACTIVE Supervised Classification

univ-cotedazur.fr

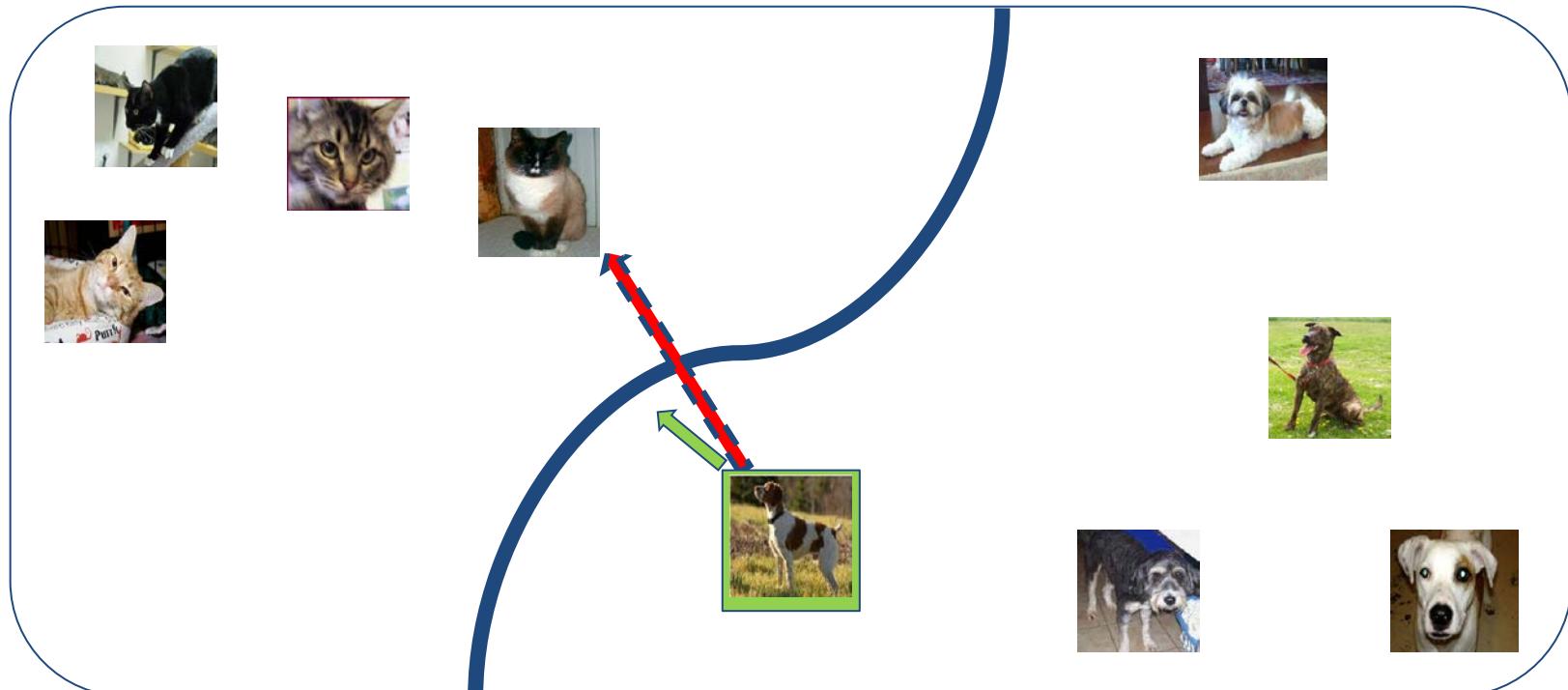




MARGIN BASED ACTIVE LEARNING?

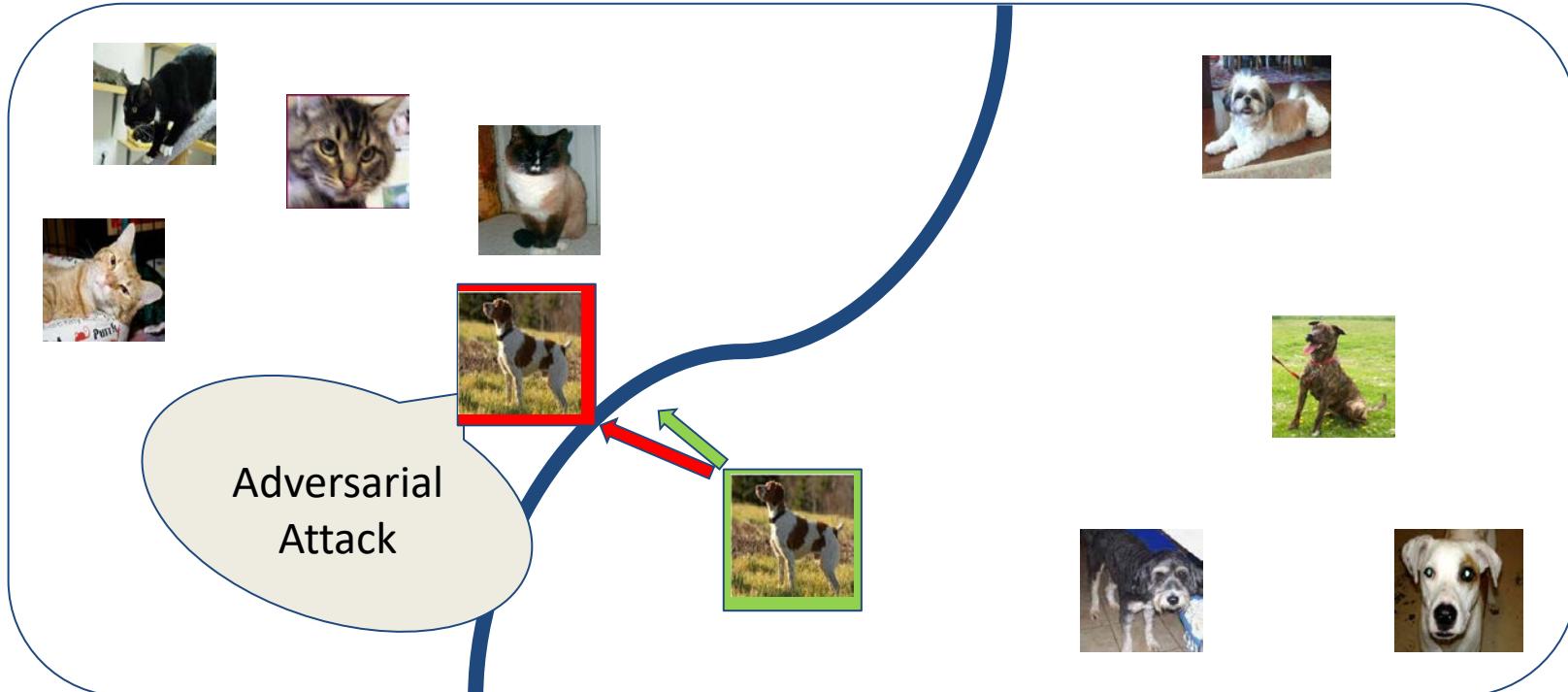
univ-cotedazur.fr

- query => close to the decision boundary
- topology of the decision boundary unknown
- how can we approximate such a distance for neural networks ?
- *“Dimension + Volume + Labels”*



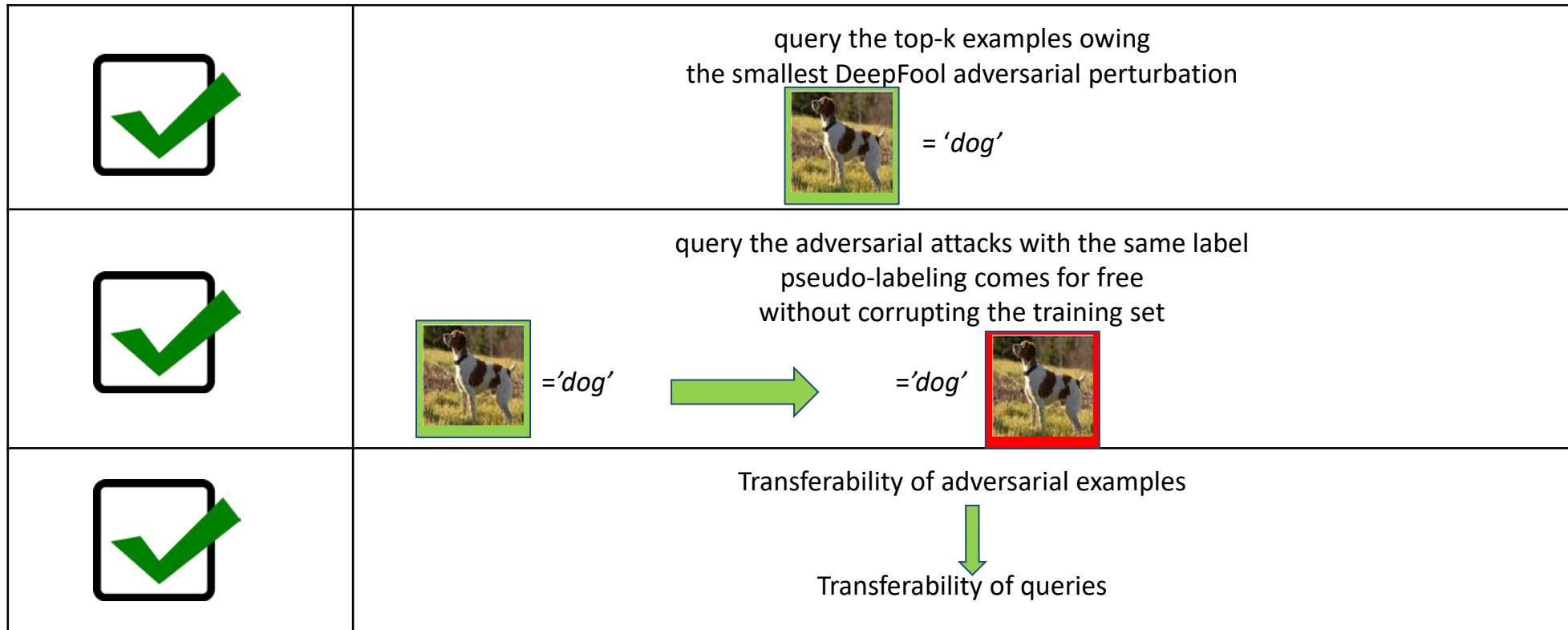
Adversarial attacks for MARGIN BASED ACTIVE LEARNING

- query → small adversarial perturbation





DeepFool Active Learning (DFAL)





DFAL EXPERIMENTS 1/3

univ-cotedazur.fr

- Query the top-k samples owing the smallest adversarial perturbation **(DFAL_0)**
- **BATCH= 10**
- **|MNIST| = 60 000 |QuickDraw| = 444 971**

# annotations	Accuracy (%)				
	100	500	800	1000	All
DFAL_0	85.08	95.89	97.79	98.13	-
BALD	53.73	91.47	94.32	94.32	-
CEAL	50.87	90.69	90.69	90.69	-
CORE-SET	78.80	96.68	97.46	97.88	-
EGL	37.92	91.84	93.99	93.99	-
uncertainty	45.57	88.36	94.27	94.60	-
RANDOM	69.79	91.96	94.05	94.46	98.98

% of Test accuracy
MNIST (VGG8)

# annotations	Accuracy (%)				
	100	500	800	1000	All
DFAL_0	78.62	91.35	92.44	93.14	-
BALD	82.00	89.94	91.92	92.87	-
CEAL	64.45	79.66	85.73	88.65	-
CORE-SET	66.71	89.93	92.28	92.62	-
EGL	63.12	86.80	90.06	90.06	-
uncertainty	52.77	88.05	89.31	91.03	-
RANDOM	78.28	88.13	89.71	89.94	96.75

% of Test accuracy
QuickDraw (VGG8)



DFAL EXPERIMENTS 2/3

- Pseudo labeling the adversarial samples of the queries
- **BATCH= 10**
- **|MNIST| = 60 000 |QuickDraw| = 444 971**

# annotations	Accuracy (%)				
	100	500	800	1000	All
DFAL	84.28	96.90	97.98	98.59	-
DFAL_0	85.08	95.89	97.79	98.13	-
BALD	53.73	91.47	94.32	94.32	-
CEAL	50.87	90.69	90.69	90.69	-
CORE-SET	78.80	96.68	97.46	97.88	-
EGL	37.92	91.84	93.99	93.99	-
uncertainty	45.57	88.36	94.27	94.60	-
RANDOM	69.79	91.96	94.05	94.46	98.98

% of Test accuracy
MNIST (VGG8)

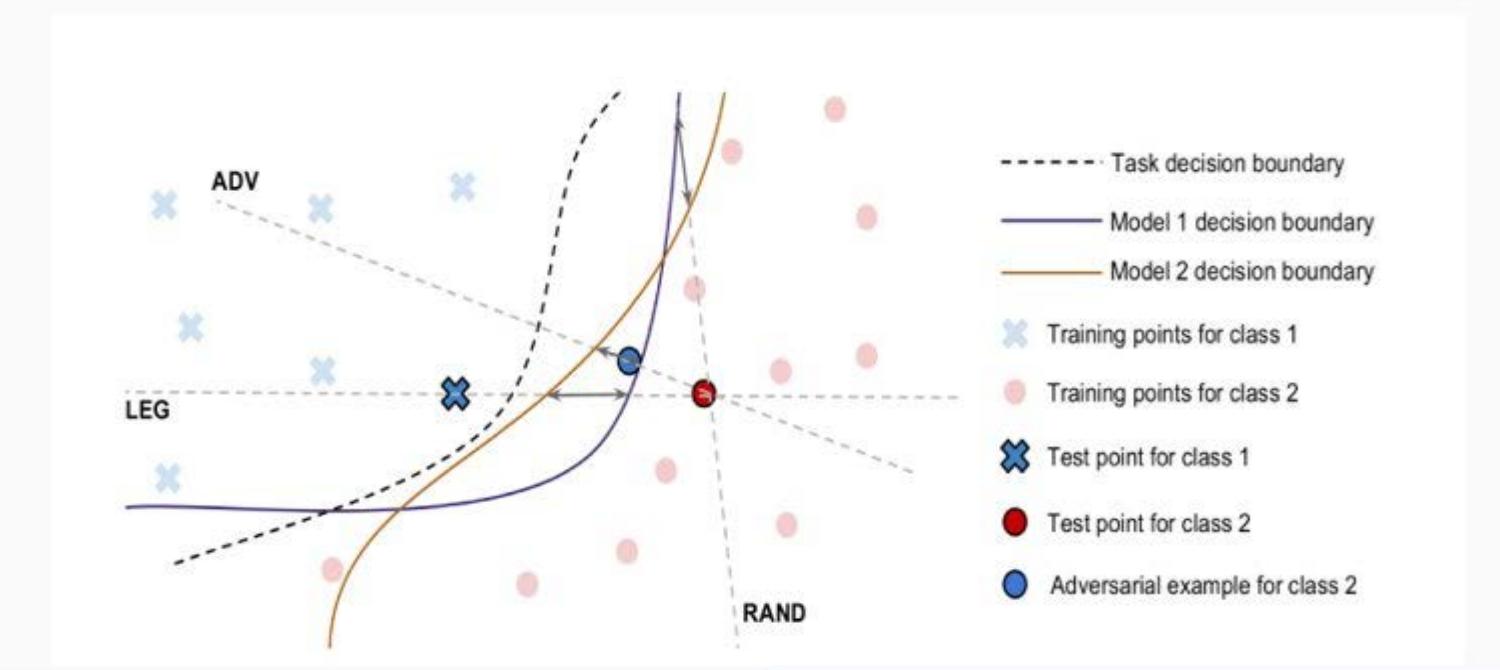
# annotations	Accuracy (%)				
	100	500	800	1000	All
DFAL	84.23	91.52	93.16	93.91	-
DFAL_0	78.62	91.35	92.44	93.14	-
BALD	82.00	89.94	91.92	92.87	-
CEAL	64.45	79.66	85.73	88.65	-
CORE-SET	66.71	89.93	92.28	92.62	-
EGL	63.12	86.80	90.06	90.06	-
uncertainty	52.77	88.05	89.31	91.03	-
RANDOM	78.28	88.13	89.71	89.94	96.75

% of Test accuracy
QuickDraw (VGG8)

TRANSFERABILITY: The ultimate threat of adversarial examples [Tramèr, 2017]

Adversarial space: contiguous, at least 2 dimensional. Dimension is proportional to the ratio increase in loss / perturbation

Different models with similar class boundary distances





DFAL EXPERIMENTS 3/3

univ-cotedazur.fr

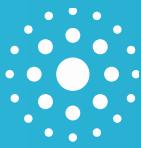
- Transferability of non targeted adversarial attacks
- 1000 queries
- Model Selection
- **|MNIST| = 60 000 |ShoeBag| = 184 792**

	DFAL	CORE-SET	RANDOM
LeNet5 → VGG8	97.80	96.90	94.46
VGG8 → LeNet5	97.93	97.40	95.31

% of Test accuracyMNIST

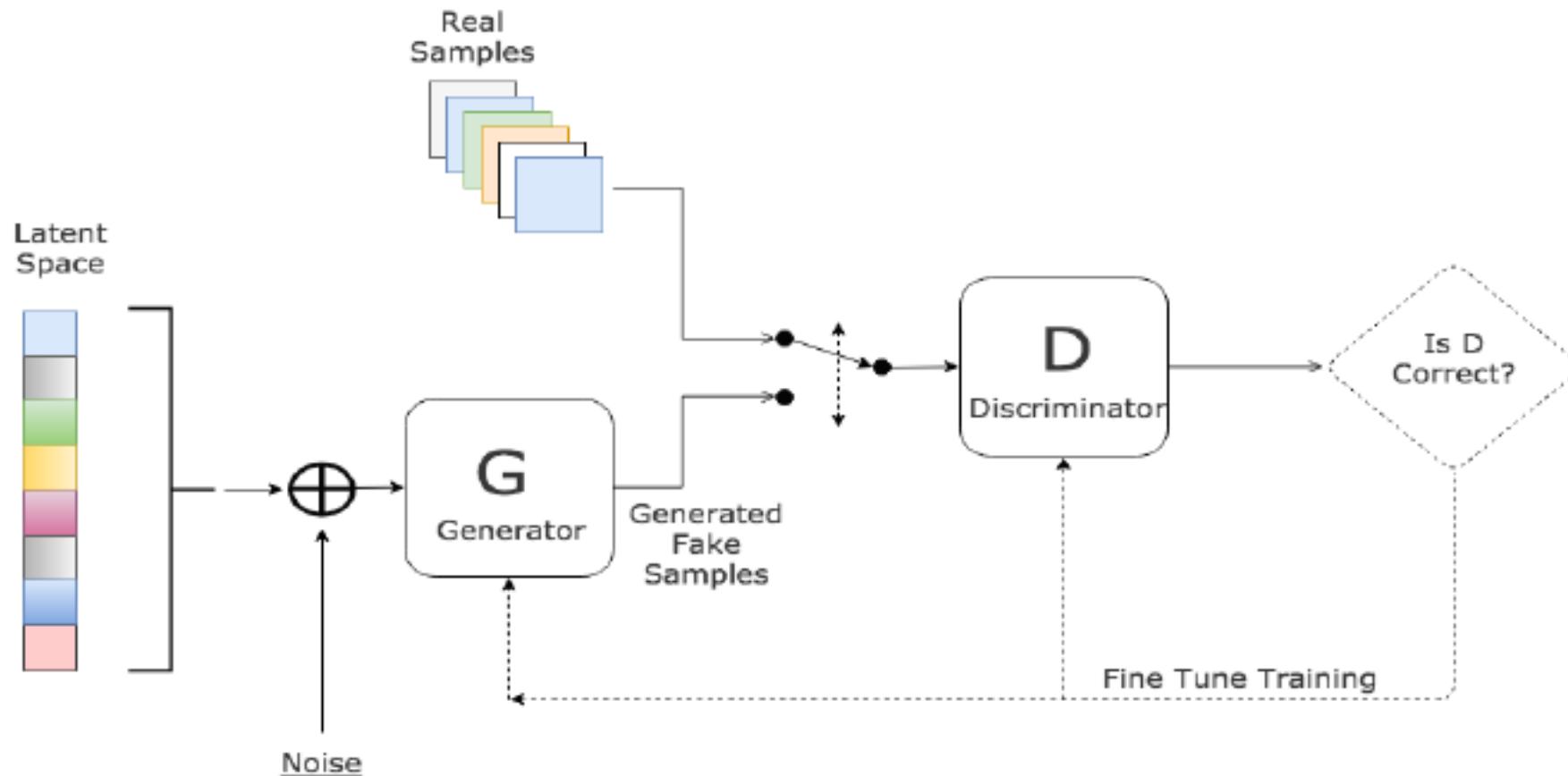
	DFAL	CORE-SET	RANDOM
LeNet5 → VGG8	99.40	99.12	97.08
VGG8 → LeNet5	98.75	98.50	98.07

% of Test accuracyShoe Bag



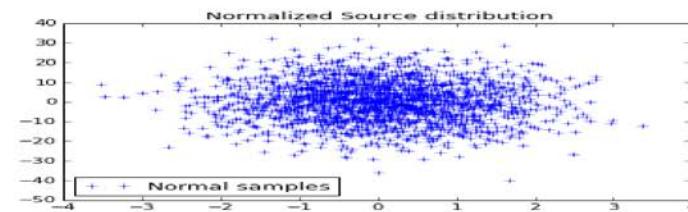
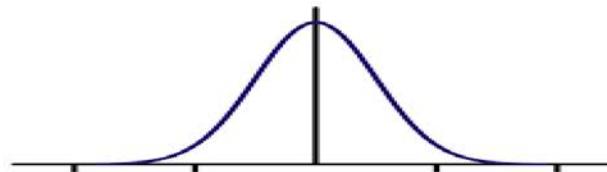
GENERATIVE ADVERSARIAL NETWORKS

How to solve Adversarial Examples? Generative Adversarial Networks



Generative Learning

- **Generative Models:** Learning how to sample from an approximate distribution close to the groundtruth



- Equation vs sampling
- RBM, VAE, Pixel CNN, **GANs** → sharpest samples

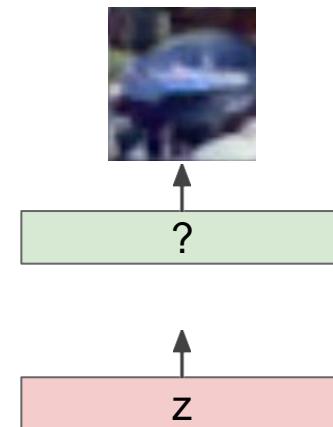
Generative Adversarial Networks

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

- Problem: Want to sample from complex, high-dimensional training distribution.
No direct way to do this!
- Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.
- Q: What can we use to represent this complex transformation?

Output: Sample from
training distribution

Input: Random noise



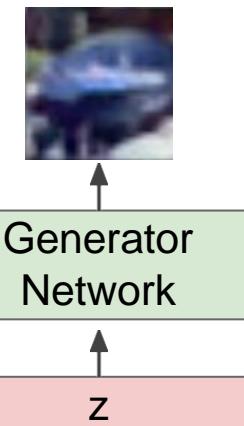
Generative Adversarial Networks

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

- Problem: Want to sample from complex, high-dimensional training distribution.
No direct way to do this!
- Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.
- Q: What can we use to represent this complex transformation?
- A: A neural network!

Output: Sample from training distribution

Input: Random noise

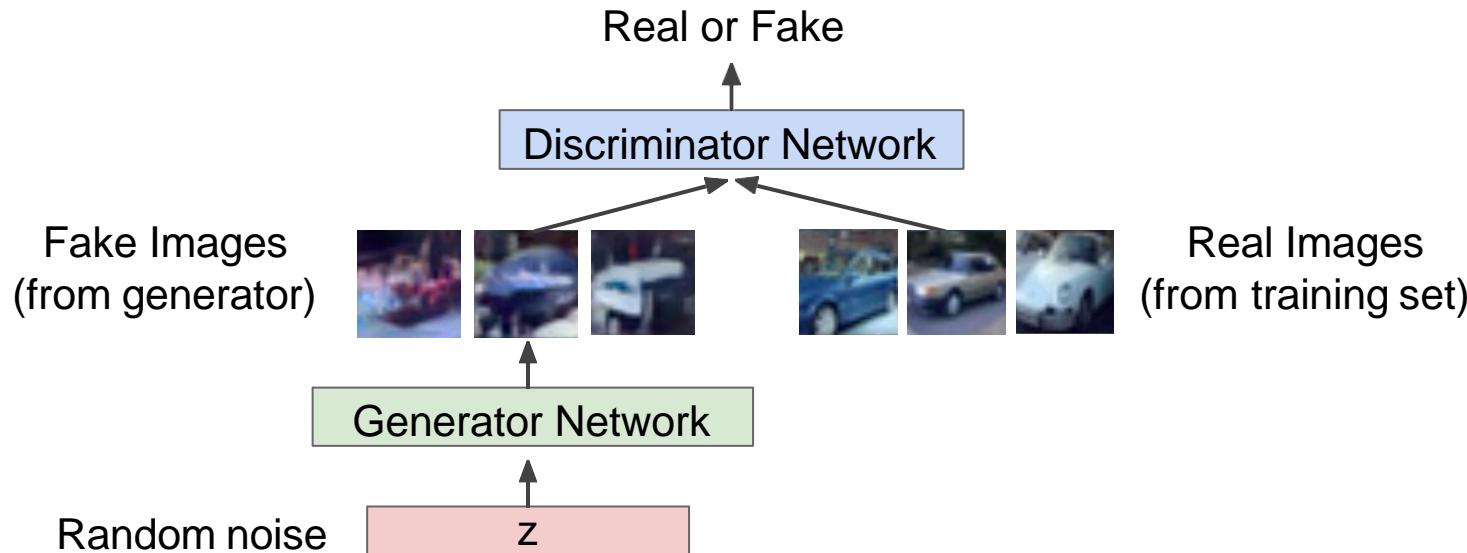


Training GANs: Two-player game

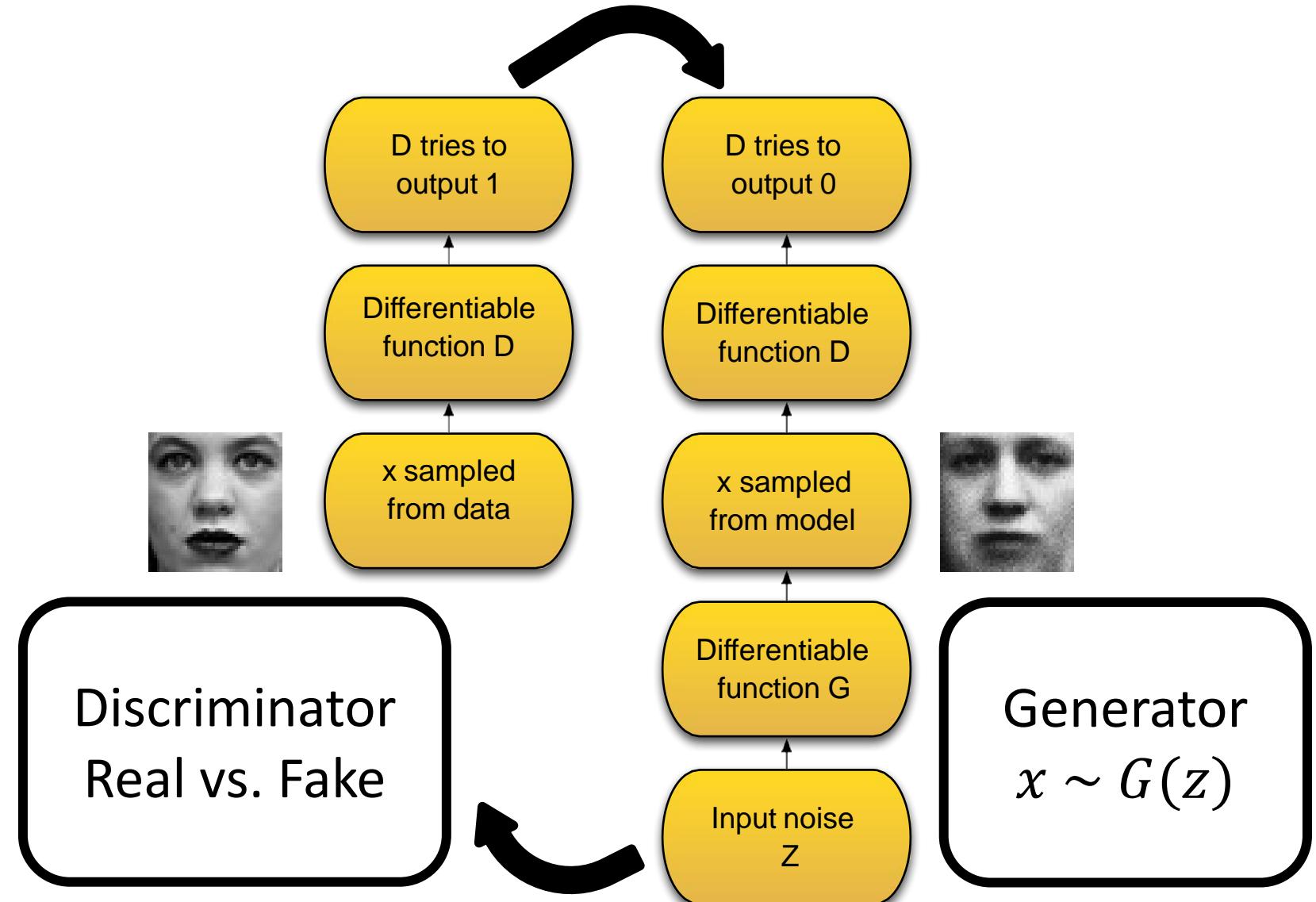
- **Generator network:** try to fool the discriminator by generating real-looking images
- **Discriminator network:** try to distinguish between real and fake images

Training GANs: Two-player game

- **Generator network:** try to fool the discriminator by generating real-looking images
- **Discriminator network:** try to distinguish between real and fake images



Adversarial Networks Framework



Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

- **Generator network:** try to fool the discriminator by generating real-looking images
- **Discriminator network:** try to distinguish between real and fake images
- Train jointly in **minimax game**
- Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Training GANs: Two-player game

- **Generator network:** try to fool the discriminator by generating real-looking images
- **Discriminator network:** try to distinguish between real and fake images
- Train jointly in **minimax game**
- Minimax objective function:

Discriminator outputs likelihood in (0,1) of real image

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data } x} + \mathbb{E}_{z \sim p(z)} \log \underbrace{(1 - D_{\theta_d}(G_{\theta_g}(z)))}_{\text{Discriminator output for generated fake data } G(z)} \right]$$

- Discriminator (θ_d) wants to **maximize objective** such that $D(x)$ is close to 1 (real) and $D(G(z))$ is close to 0 (fake)
- Generator (θ_g) wants to **minimize objective** such that $D(G(z))$ is close to 1 (discriminator is fooled into thinking generated $G(z)$ is real)

Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

- Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

- Alternate between:

1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

2. **Gradient descent** on generator

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Alternate between:

1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

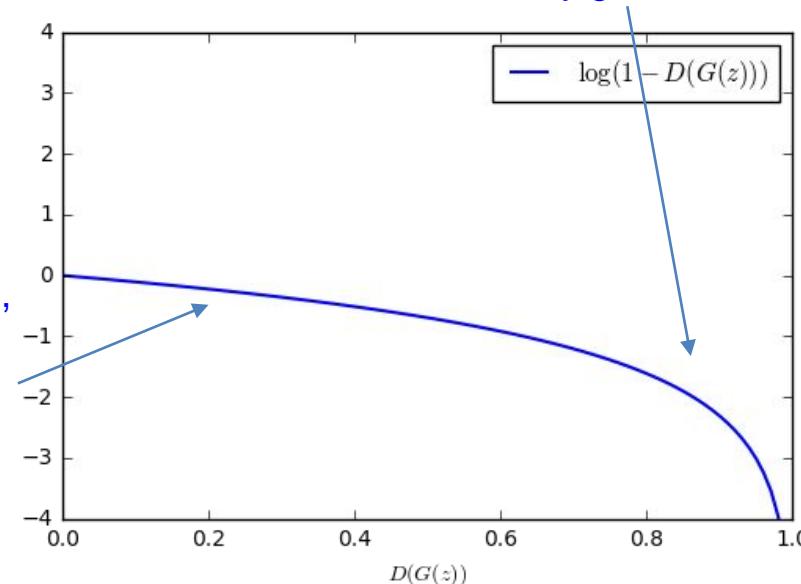
Gradient signal dominated by region where sample is already good

2. **Gradient descent** on generator

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

In practice, optimizing this generator objective does not work well!

When sample is likely fake, want to learn from it to improve generator, but gradient in this region is relatively flat!



Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Alternate between:

1. Gradient ascent on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

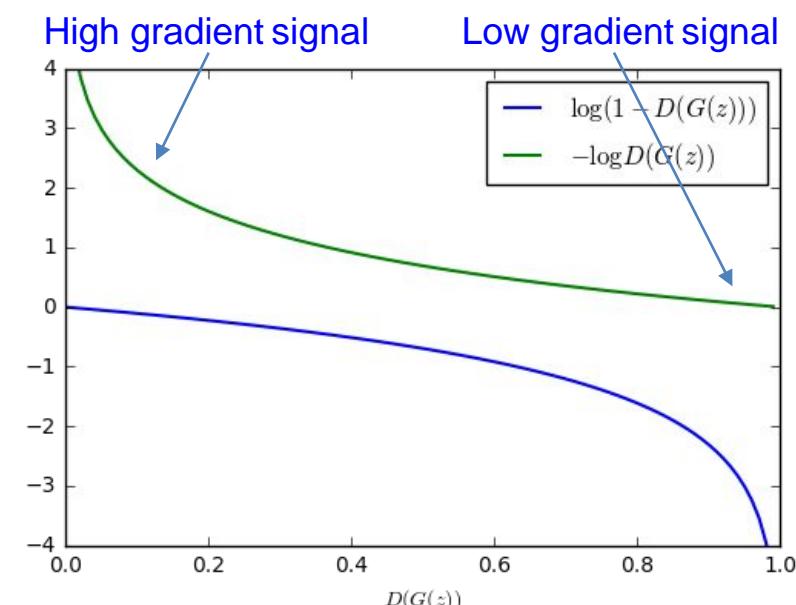
2. Instead: Gradient ascent on generator, different

objective

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$

Instead of minimizing likelihood of discriminator being correct, now maximize likelihood of discriminator being wrong.

Same objective of fooling discriminator, but now higher gradient signal for bad samples => works much better! Standard in practice.





Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Putting it together: GAN training algorithm

for number of training iterations **do**

for k steps **do**

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

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

end for

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

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

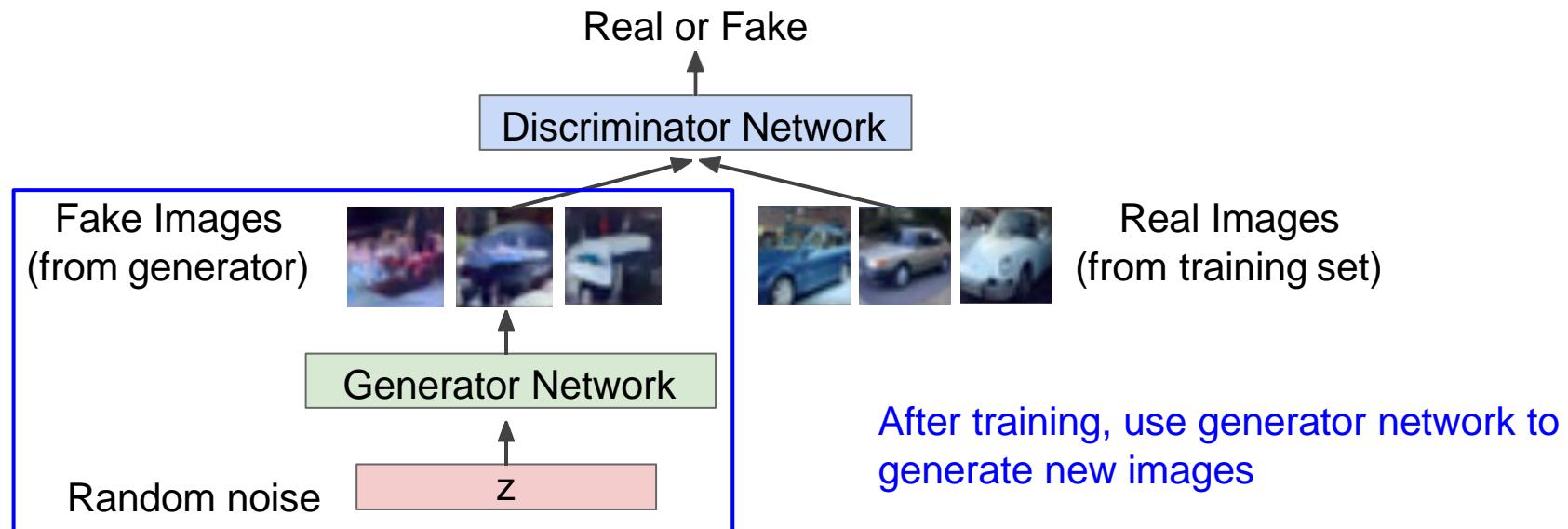
end for

Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images

Discriminator network: try to distinguish between real and fake images





GAN convergence

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \mathcal{V}(\mathbf{D}, \mathbf{G}) = \mathbb{E}_{x \sim p_{data}(x)} [\log(\mathbf{D}(x))] + \mathbb{E}_{z \sim p_z(z)} [1 - \log(\mathbf{D}(\mathbf{G}(z)))]$$

- If the discriminator is optimal: $\mathbf{D} \equiv \mathbf{D}^*$ (*NNs are universal approximators*)

$$\mathbf{D}^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_{model}(x)} \quad (1)$$

$$\mathcal{V}(\mathbf{D}^*, \mathbf{G}) = 2 * JSD(p_{data} || p_{model}) - 2 \ln(2) \quad (2)$$

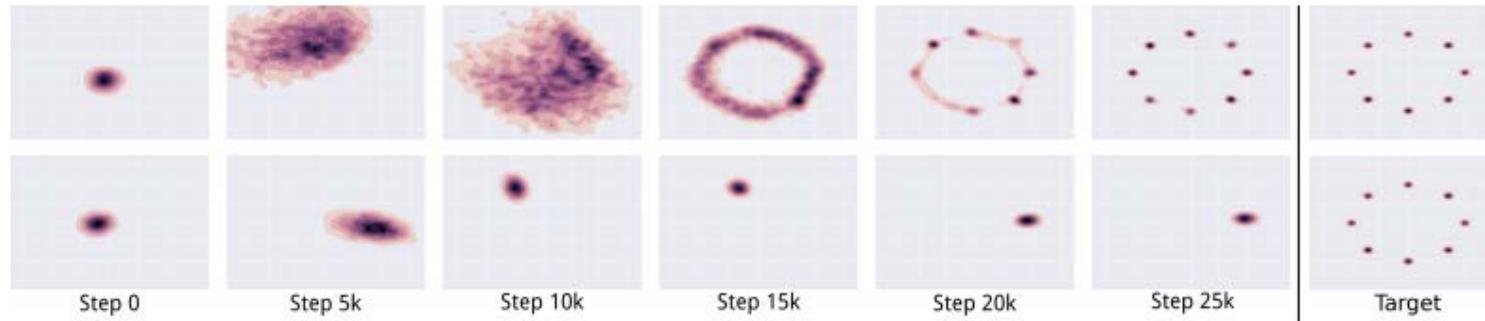
- with certain conditions, it converges asymptotically to the groundtruth distribution, which may not be the case for other maximum likelihood based generative models
- G has converged to p_{data} and D outputs $\frac{1}{2}$

The struggle is real

- GAN models often end up in local Nash equilibria that are associated with mode collapse or otherwise fail to model the target distribution
- cost function is non convex
- parameters are continuous
- high dimensional parameter space
- no way to measure the quality of the generation process (not even log likelihood)

Mode Collapse

- Multiple peaks, generator with poor diversity
- Gradient vanishing

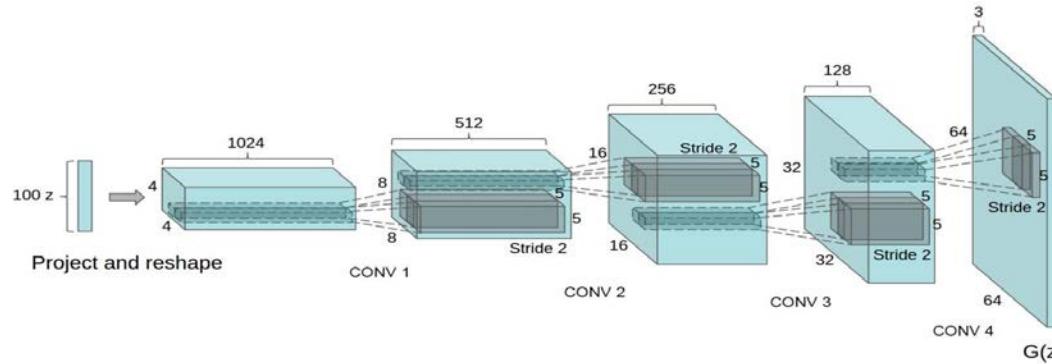


With mode collapse, the model only learns one or a few of the modes of a multimodal dataset. The data usually used to train GANs on typical problems have a very large number of modes, which makes mode collapse problematic.

- Multiple GANS: AdaGANs

Generative Adversarial Nets: Convolutional Architectures (DCGAN)

- Generator is an upsampling network with fractionally-strided convolutions
- Discriminator is a convolutional network

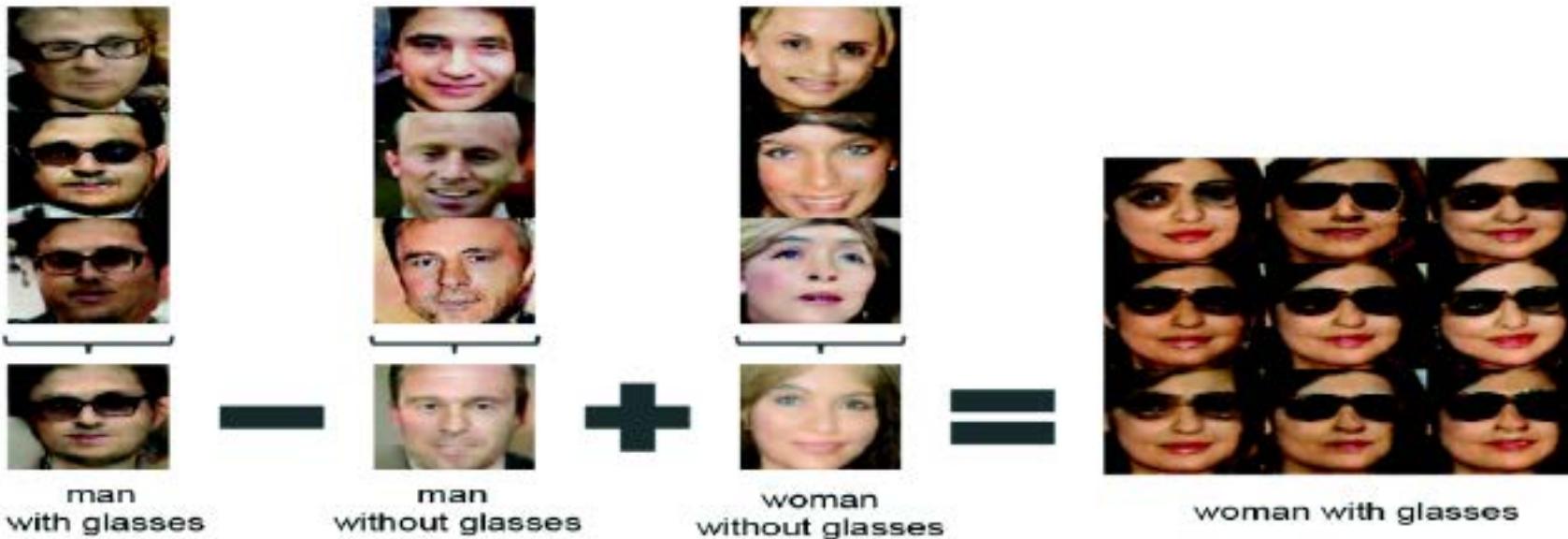


Architecture guidelines for stable Deep Convolutional GANs

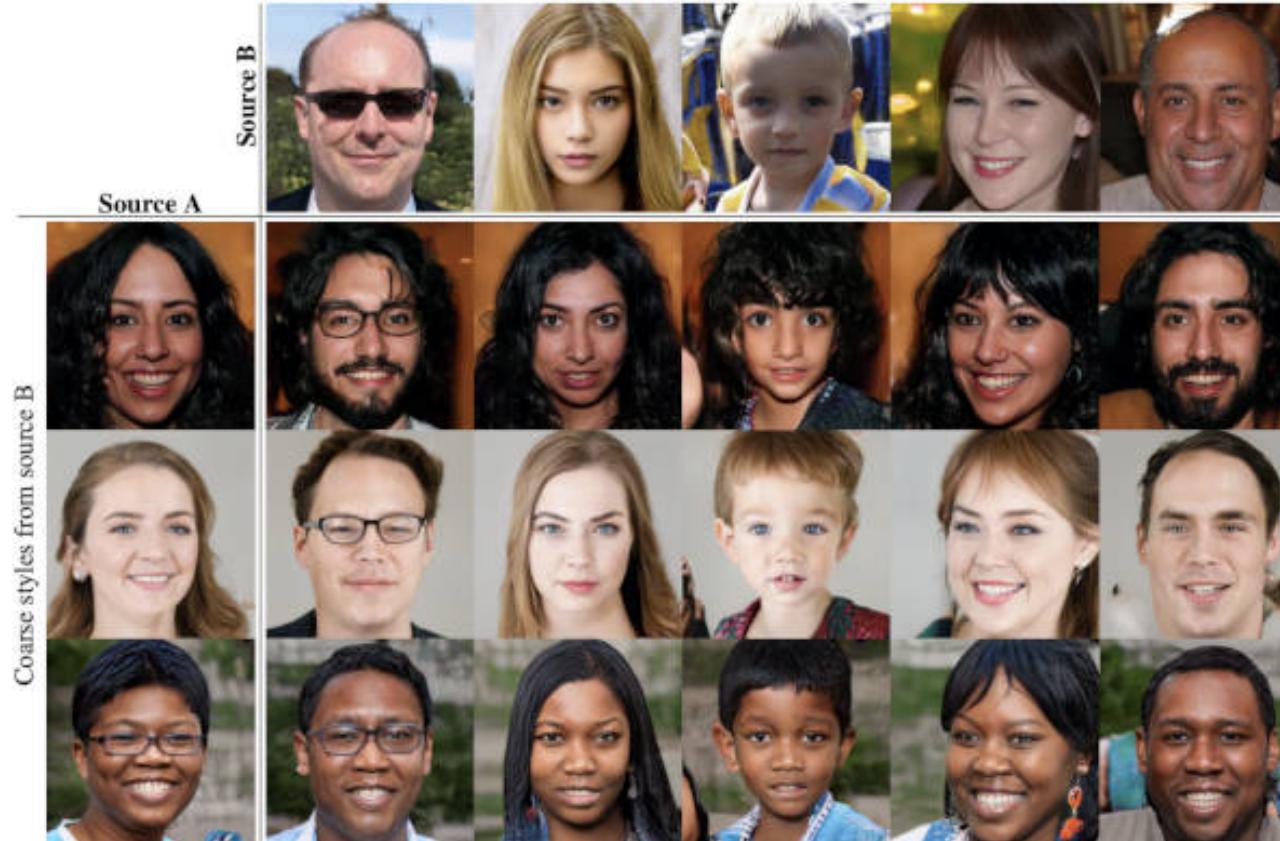
- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

It finally did not solve adversarial, but...

Results DCGAN



It finally did not solve adversarial, but...



(image from [Karras et al. 2019](#))



Tips and tricks

Tips & Tricks

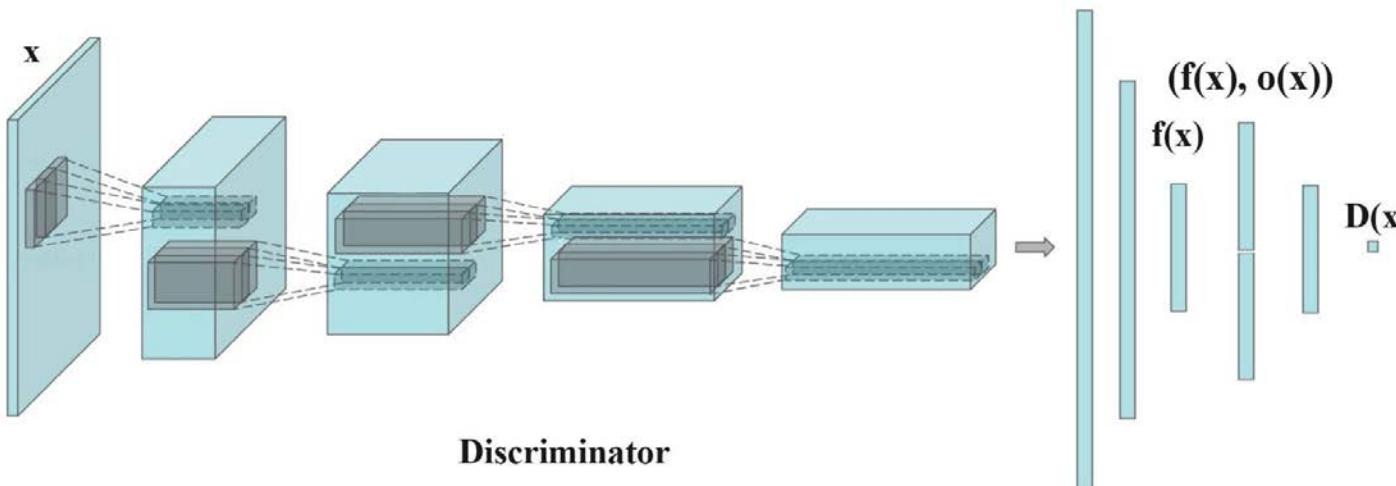
- Normalize your training data into range [-1, 1]
- Pick a bounded output activation function for G (sigmoid, tanh...)
- Do not sabotage your Discriminator: usually deeper than G and with an unbounded output activation
- Label smoothing for real training data
- **Feature matching (facial for instance)**
- Learning a conditional model often gives better samples (add a generated class)
- Do not balance D and G losses

Tips & Tricks

- Add BN in G (not the last layer, maybe not a good idea for D)
- minibatch of data X : $\mu = \text{mean}(X)$, $\sigma = \text{std}(X)$
- Add two extra parameters of the network γ , β
- $BN(x \in X)_{\gamma, \beta} = \gamma \frac{x - \mu}{\sqrt{\sigma + \epsilon}} + \beta$

Tips & Tricks

- Preventing mode collapsing: discriminant minibatch



When mode collapses, all images created looks similar.

⇒ feed real images and generated images into the discriminator separately in different batches and compute the similarity of the image x with images in the same batch.

If the mode starts to collapse, the similarity of generated images increases. The discriminator can use this score to detect generated images and penalize the generator if mode is collapsing.

The similarity $o(x_i)$ between the image x_i and other images in the same batch is computed and we append the similarity $o(x)$ in one of the dense layers in the discriminator to classify whether this image is real or generated.

GAN training is challenging

- Vanishing gradient – when discriminator is very good
- Mode collapse – too little diversity in the samples generated
- Lack of convergence because hard to reach Nash equilibrium
- Loss metric does not always correspond to image quality; Frechet Inception Distance (FID) is a decent choice

Wasserstein GAN

- New objective function: Earth Mover Distance

$$\mathbb{W}(p_{data} \mid p_{model}) = \inf_{\gamma \in \Pi(p_{data}, p_{model})} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|] \quad (4)$$

- Optimal Transport in the dual space: training on minibatch
- Discriminator: 1 Lipschitz function, weight clipping
- NNs are already Lipschitz function =)

$$\mathbb{W}(p_{data} \mid p_{model}) = \max_{\mathbf{D}, \|\mathbf{D}\|_L=1} \mathbb{E}_{x \sim p_{data}} [\mathbf{D}(x)] - \mathbb{E}_{z \sim p_z} [\mathbf{D}(\mathbf{G}(z))] \quad (5)$$



Alternative loss functions

Name	Paper Link	Value Function
GAN	Arxiv	$L_D^{GAN} = E[\log(D(x))] + E[\log(1 - D(G(z)))]$ $L_G^{GAN} = E[\log(D(G(z)))]$
LSGAN	Arxiv	$L_D^{LSGAN} = E[(D(x) - 1)^2] + E[D(G(z))^2]$ $L_G^{LSGAN} = E[(D(G(z)) - 1)^2]$
WGAN	Arxiv	$L_D^{WGAN} = E[D(x)] - E[D(G(z))]$ $L_G^{WGAN} = E[D(G(z))]$ $W_D \leftarrow clip_by_value(W_D, -0.01, 0.01)$
WGAN_GP	Arxiv	$L_D^{WGAN_GP} = L_D^{GAN} + \lambda E[VD(ax - (1 - \alpha G(z))) - 1]^2$ $L_G^{WGAN_GP} = L_G^{GAN}$
DRAGAN	Arxiv	$L_D^{DRAGAN} = L_D^{GAN} + \lambda E[VD(ax - (1 - \alpha x_p)) - 1]^2$ $L_G^{DRAGAN} = L_G^{GAN}$
CGAN	Arxiv	$L_D^{CGAN} = E[\log(D(x, c))] + E[\log(1 - D(G(z), c))]$ $L_G^{CGAN} = E[\log(D(G(z), c))]$
infoGAN	Arxiv	$L_{D,Q}^{infoGAN} = L_D^{GAN} - \lambda L_I(c, c')$ $L_G^{infoGAN} = L_G^{GAN} - \lambda L_I(c, c')$
ACGAN	Arxiv	$L_{D,Q}^{ACGAN} = L_D^{GAN} + E[P(class = c x)] + E[P(class = c G(z))]$ $L_G^{ACGAN} = L_G^{GAN} + E[P(class = c G(z))]$
EBGAN	Arxiv	$L_D^{EBGAN} = D_{AE}(x) + \max(0, m - D_{AE}(G(z)))$ $L_G^{EBGAN} = D_{AE}(G(z)) + \lambda \cdot PT$
BEGAN	Arxiv	$L_D^{BEGAN} = D_{AE}(x) - k_t D_{AE}(G(z))$ $L_G^{BEGAN} = D_{AE}(G(z))$ $k_{t+1} = k_t + \lambda(\gamma D_{AE}(x) - D_{AE}(G(z)))$

<https://github.com/hwalsuklee/tensorflow-generative-model-collections>

https://medium.com/@jonathan_hui/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490

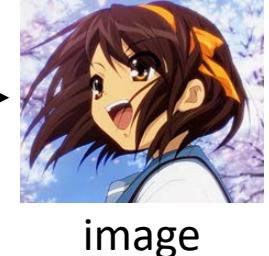


Three Categories of GANs

1. Typical GAN

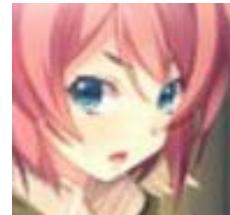

$$\begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.9 \end{bmatrix}$$

random vector



image

2. Conditional GAN



blue eyes,
red hair,
short hair
paired data

“Girl with
red hair”
text



image

3. Unsupervised Conditional GAN

domain x



domain y



x



Photo

Generator

y

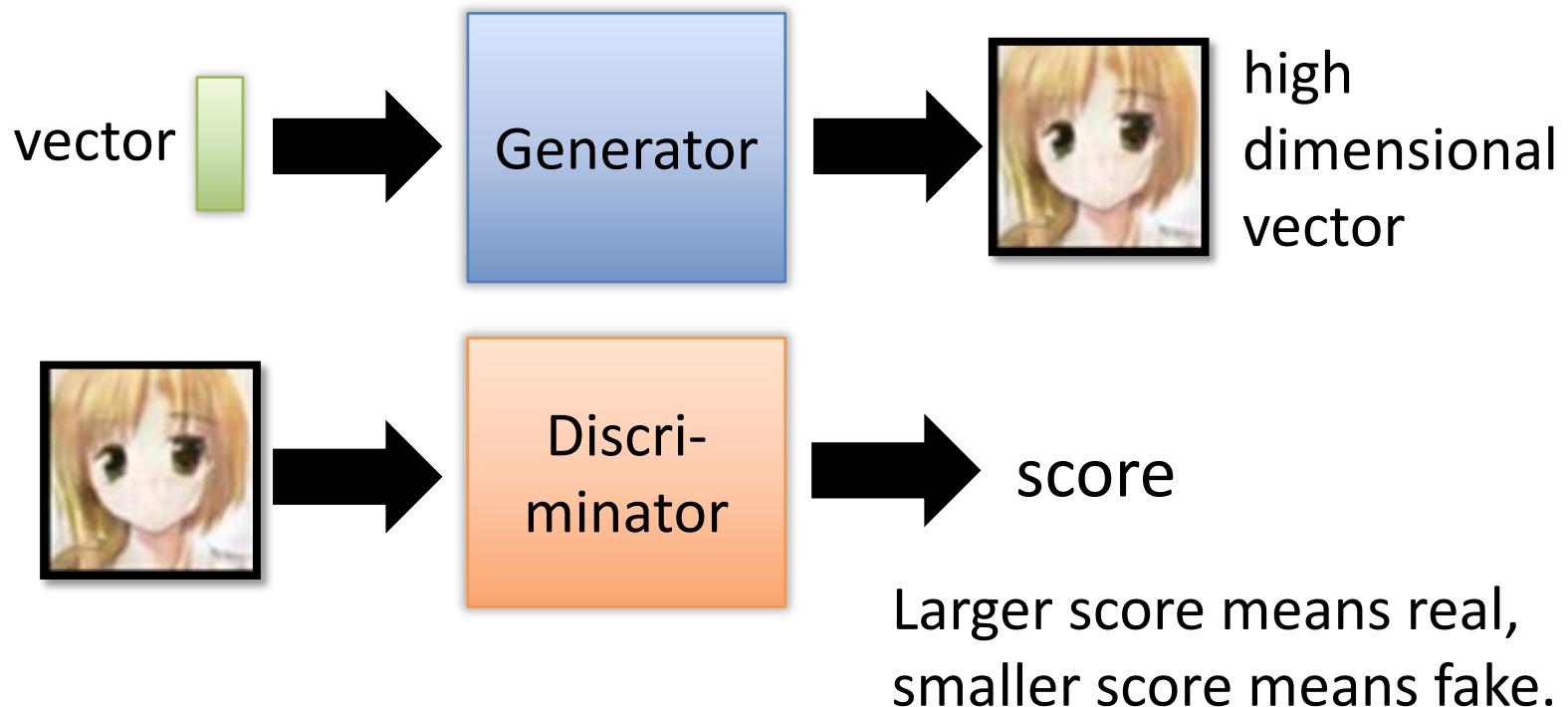


Vincent van
Gogh's style

unpaired data

Generative Adversarial Network (GAN)

- Anime face generation as example

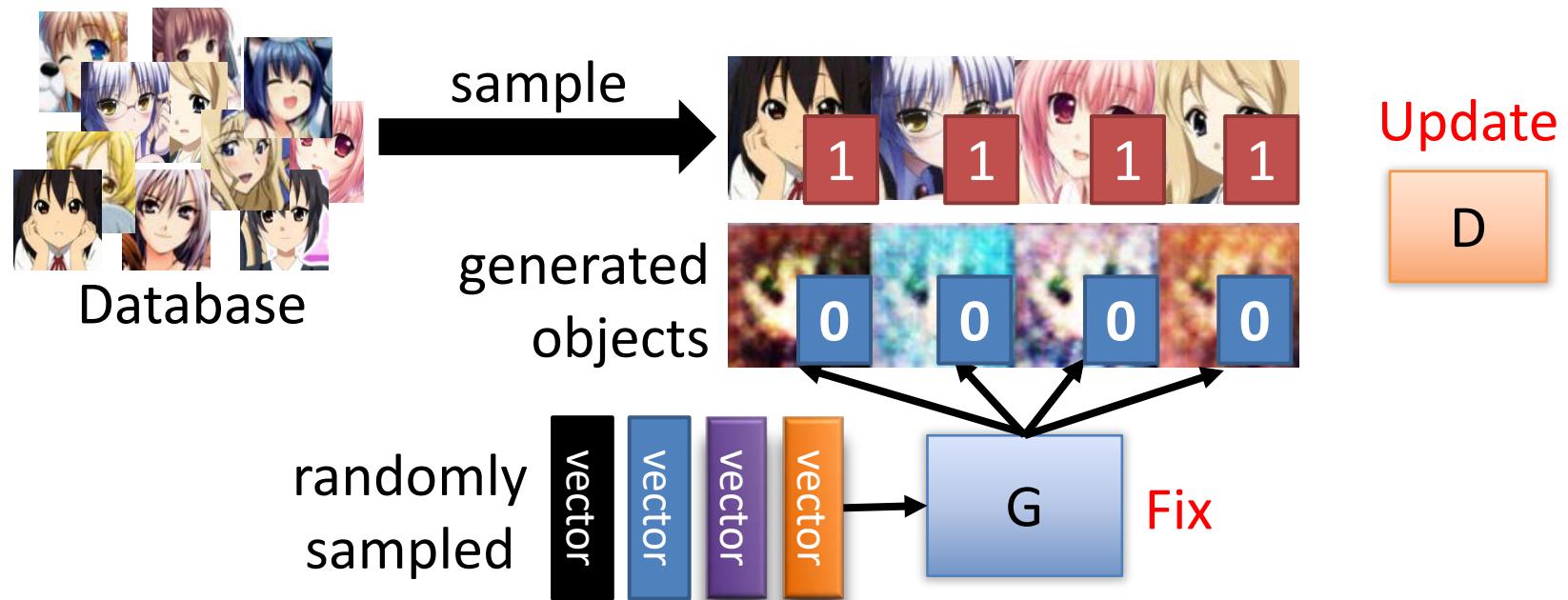


Algorithm

- Initialize generator and discriminator
- In each training iteration:

G D

Step 1: Fix generator G, and update discriminator D



Discriminator learns to assign high scores to real objects and low scores to generated objects.



Algorithm

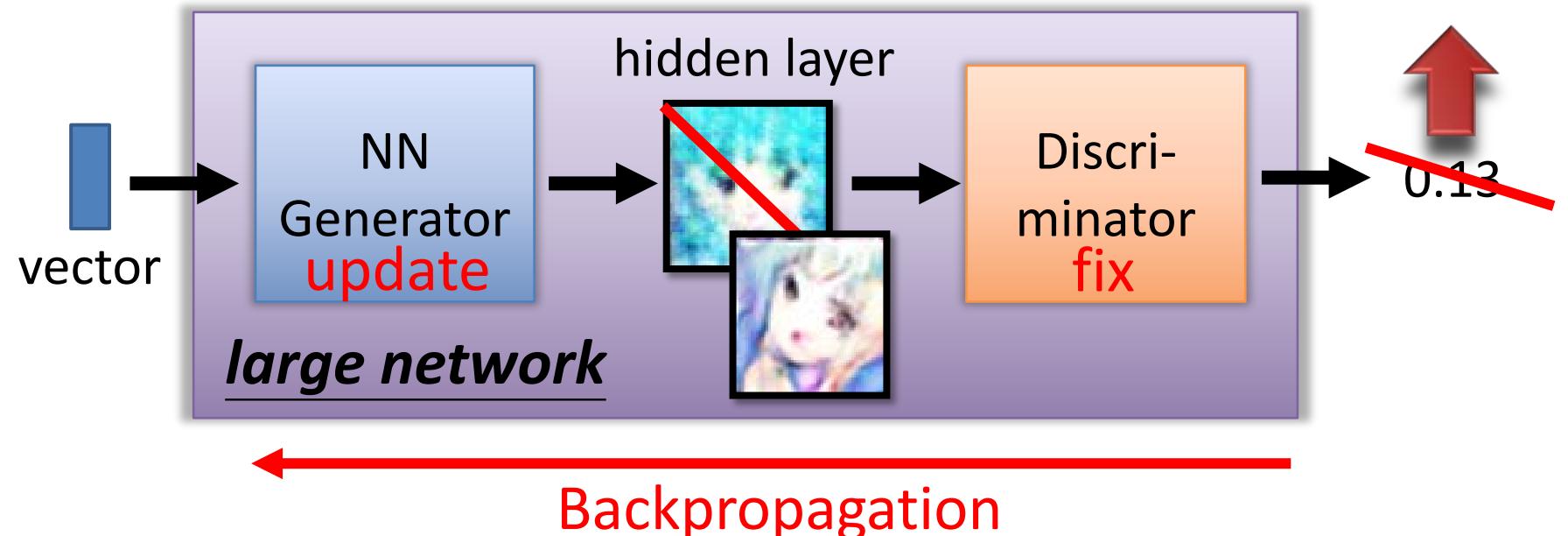
G

D

- Initialize generator and discriminator
- In each training iteration:

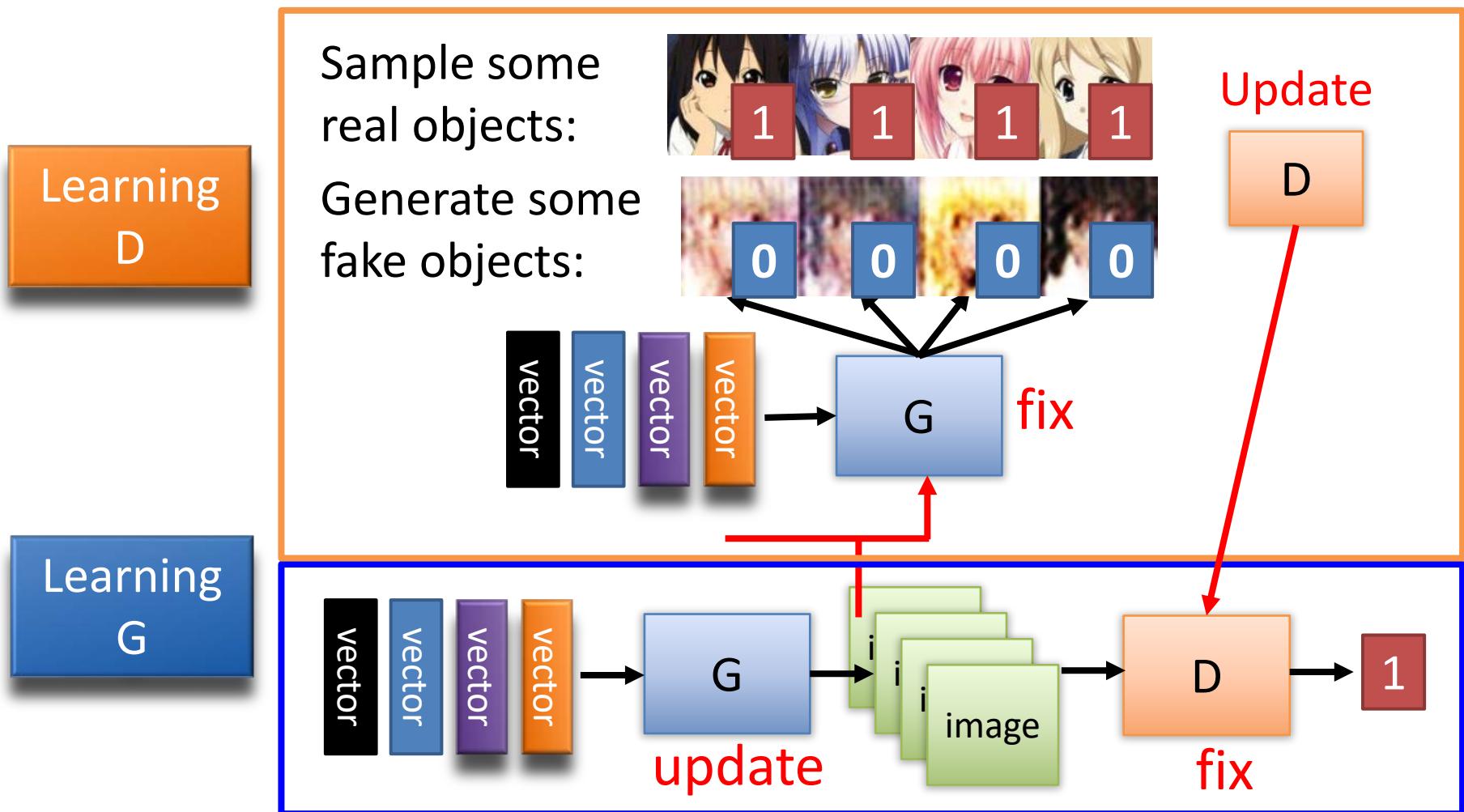
Step 2: Fix discriminator D, and update generator G

Generator learns to “fool” the discriminator



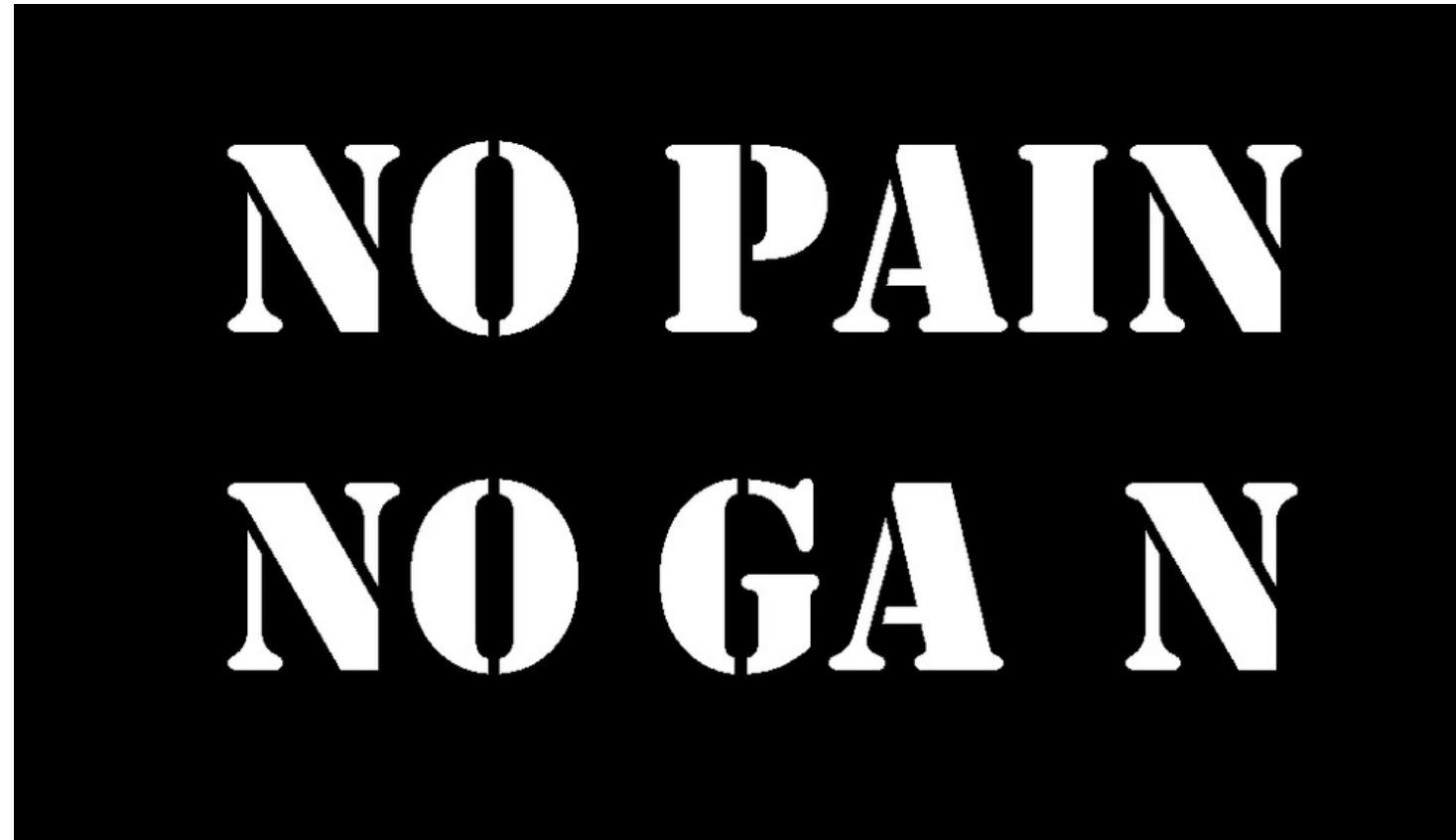
Algorithm

- Initialize generator and discriminator
- In each training iteration:



GAN is hard to train

- There is a saying



(Joke from facebook)

Three Categories of GANs

1. Typical GAN

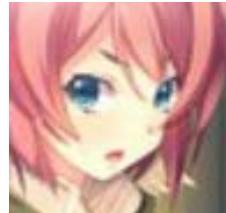

$$\begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.9 \end{bmatrix}$$

random vector



image

2. Conditional GAN



blue eyes,
red hair,
short hair
paired data

“Girl with
red hair”
text



image

3. Unsupervised Conditional GAN

domain x



domain y



x



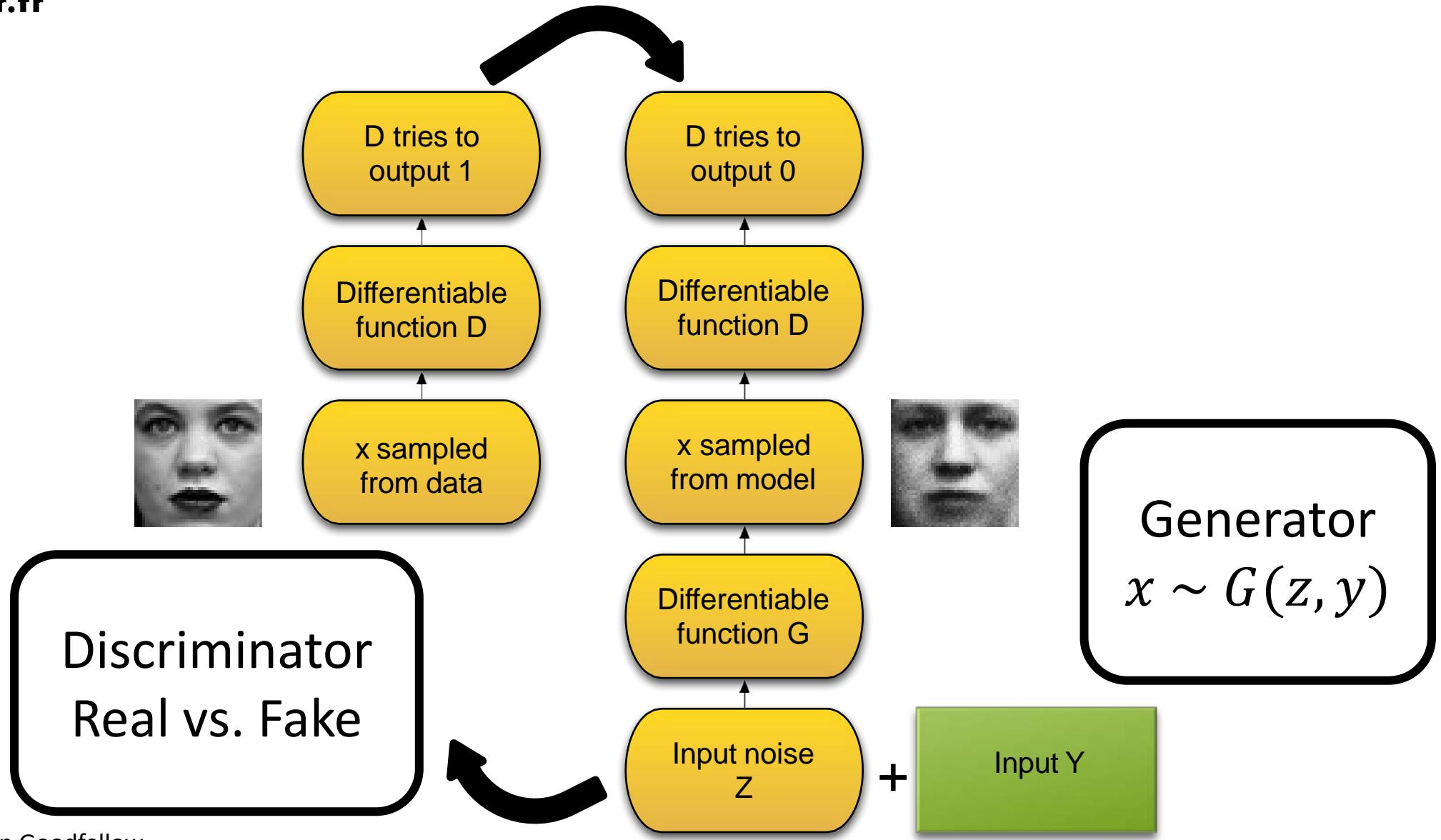
Photo

Generator

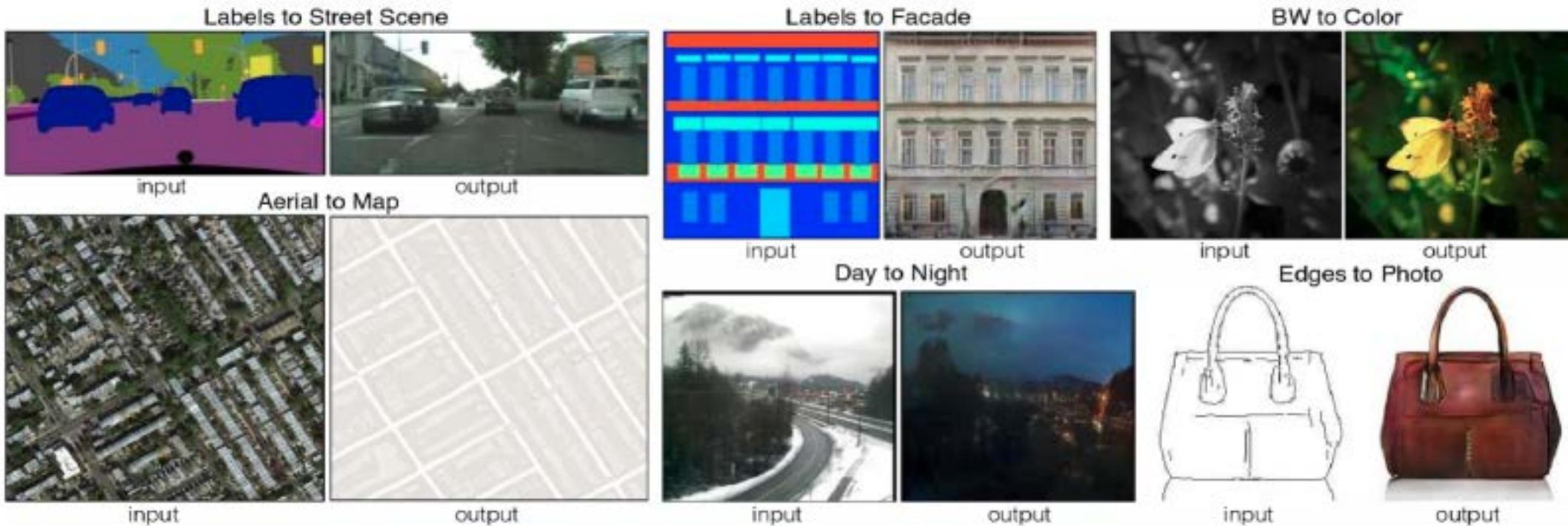


Vincent van
Gogh's style

unpaired data



It finally did not solve adversarial, but...





Text-to-Image

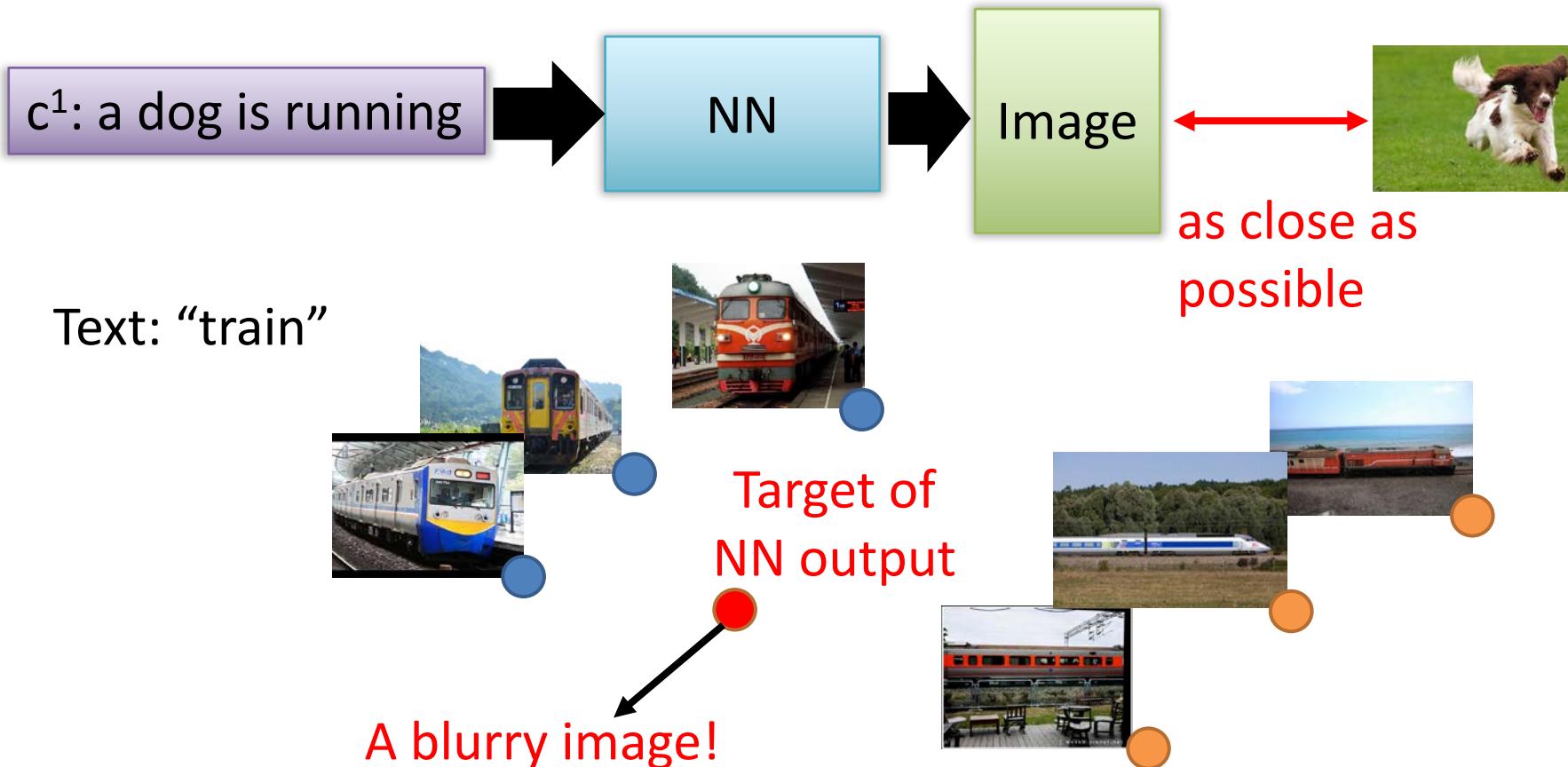
a dog is running



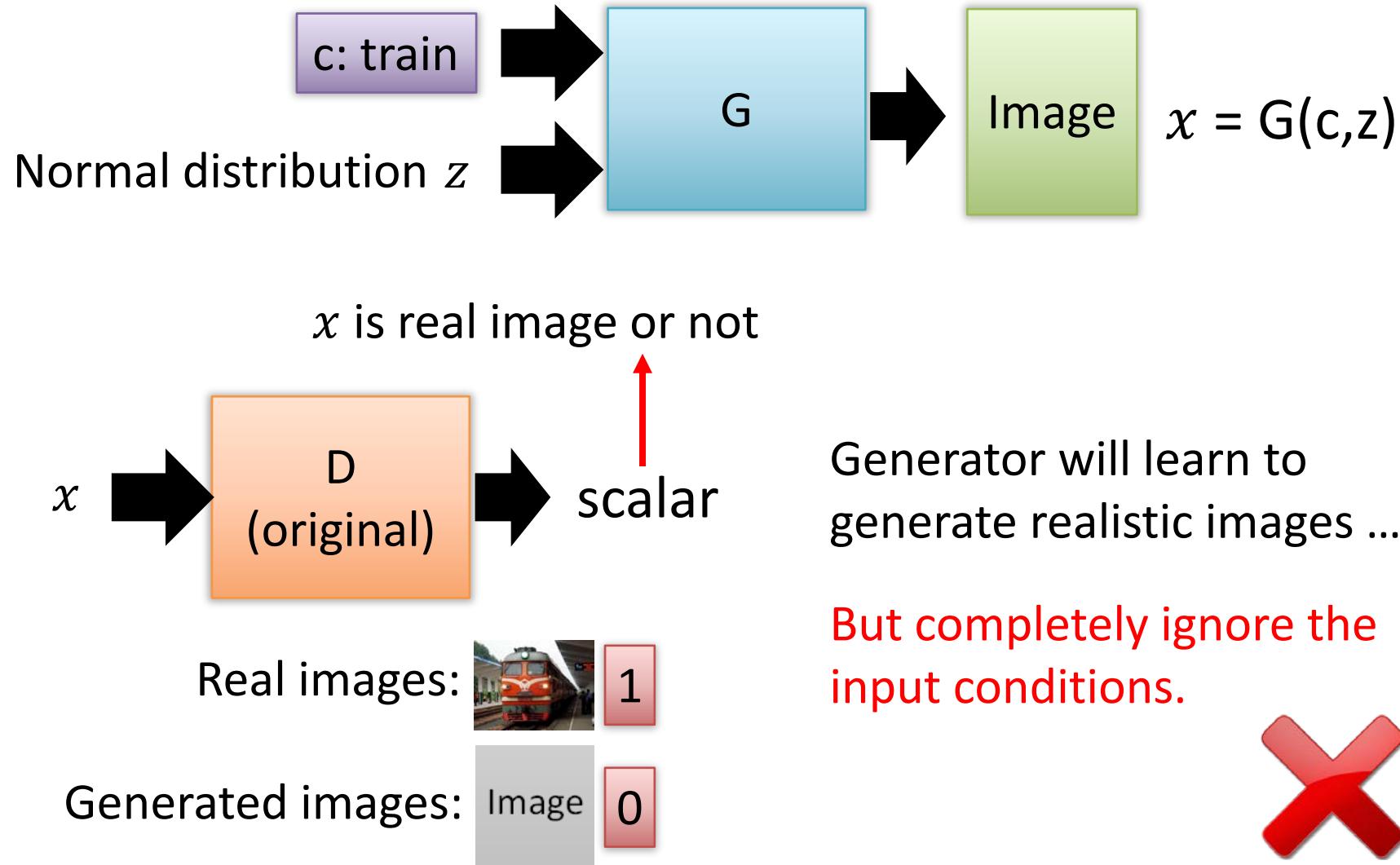
a bird is flying



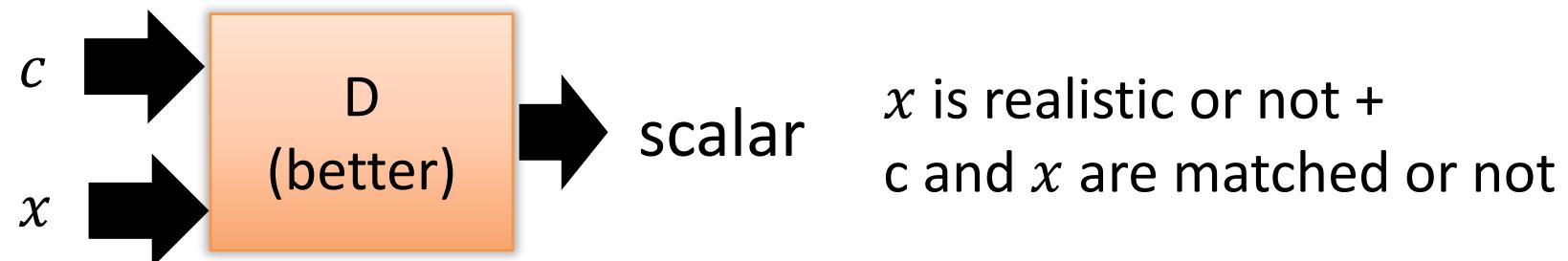
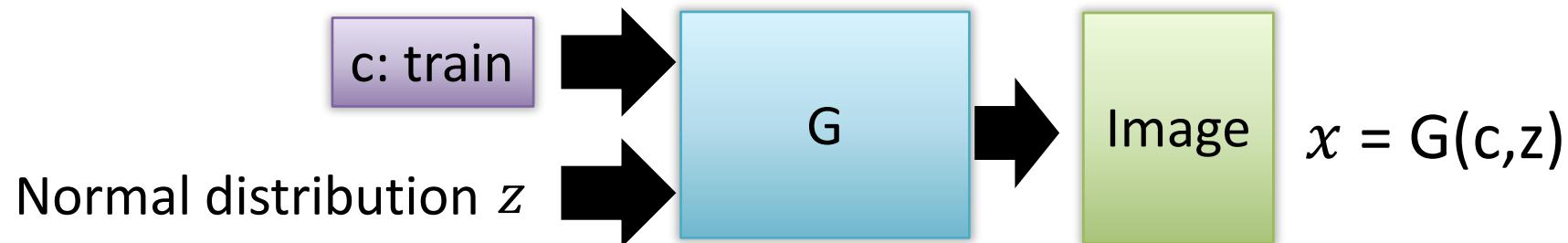
- Traditional supervised approach



Conditional GAN



Conditional GAN



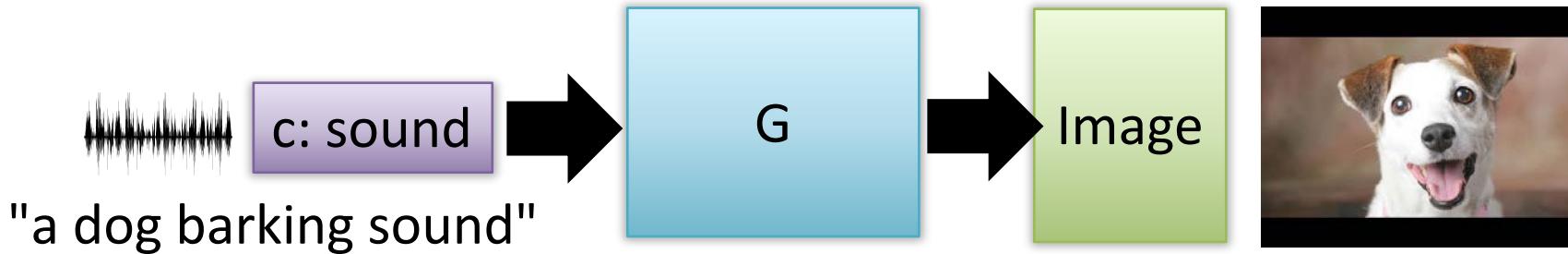
True text-image pairs:

(train , ) 1

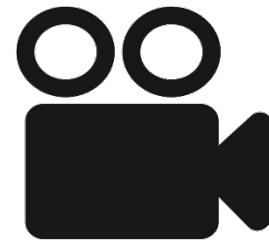
(cat , ) 0

(train , ) 0

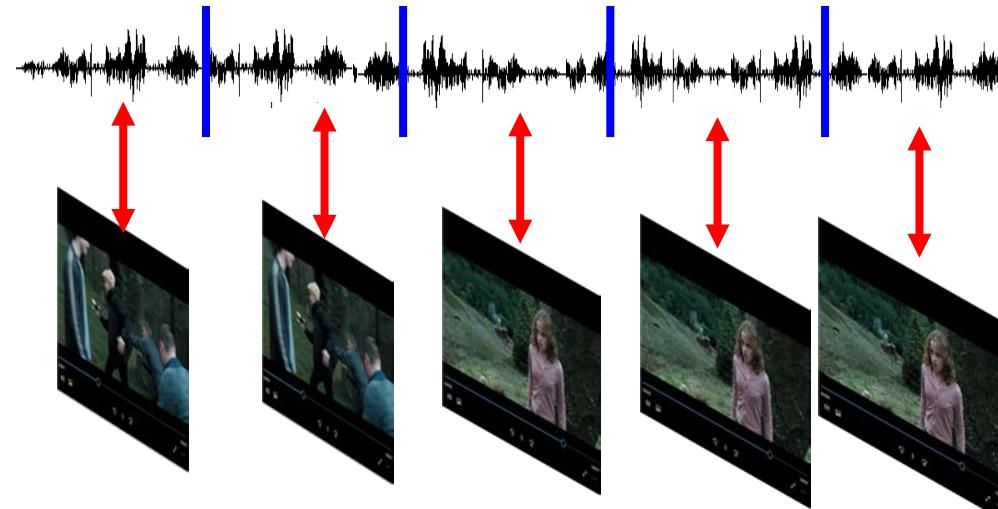
Conditional GAN - Sound-to-image



Training Data Collection



video



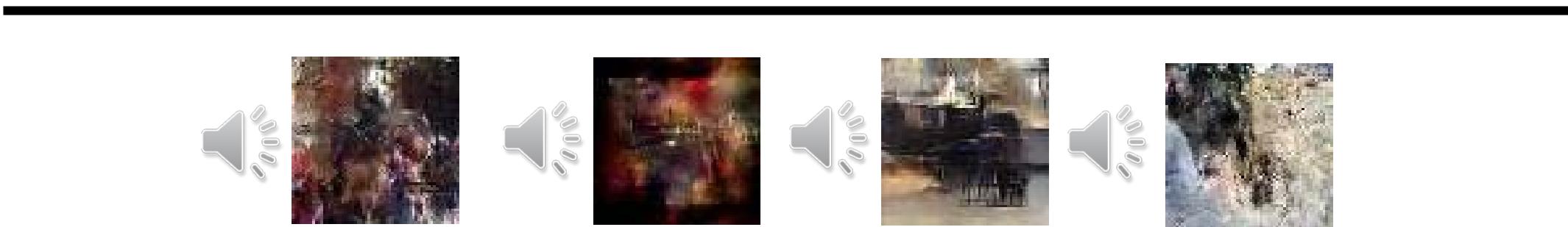


Conditional GAN

- Sound-to-image

- Audio-to-image

Louder



The images are generated by Chia-Hung Wan and Shun-Po Chuang.

[https://wjohn1483.github.io/
audio_to_scene/index.html](https://wjohn1483.github.io/audio_to_scene/index.html)

Conditional GAN - Image-to-label

Multi-label Image Classifier

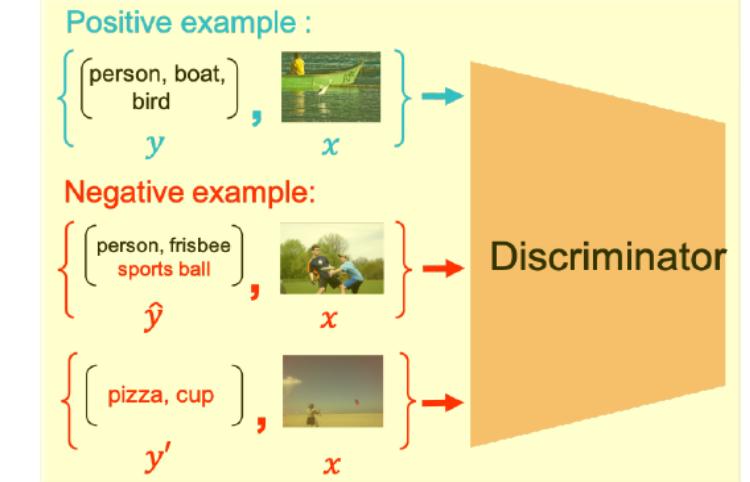
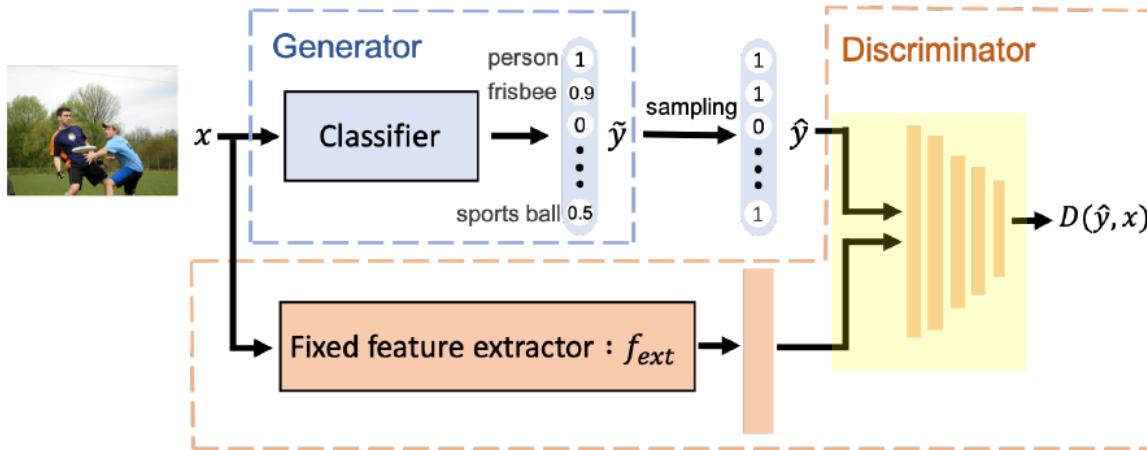


Input condition

person, sports ball,
baseball bat, baseball glove



Generated output



Conditional GAN - Image-to-label

The classifiers can have different architectures.

The classifiers are trained as conditional GAN.

[Tsai, et al., submitted to ICASSP 2019]

F1	MS-COCO	NUS-WIDE
VGG-16	56.0	33.9
+ GAN	60.4	41.2
Inception	62.4	53.5
+GAN	63.8	55.8
Resnet-101	62.8	53.1
+GAN	64.0	55.4
Resnet-152	63.3	52.1
+GAN	63.9	54.1
Att-RNN	62.1	54.7
RLSD	62.0	46.9

Conditional GAN - Image-to-label

The classifiers can have different architectures.

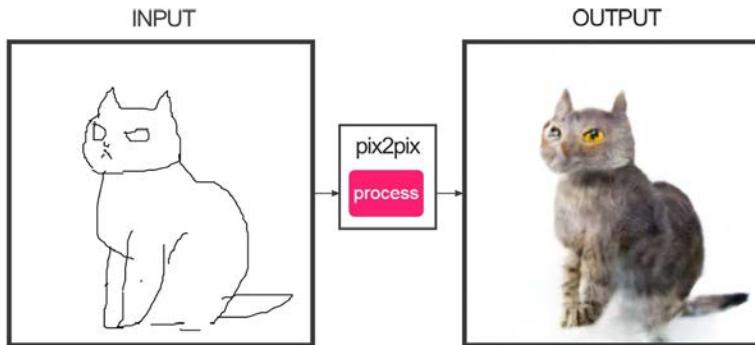
The classifiers are trained as conditional GAN.

Conditional GAN outperforms other models designed for multi-label.

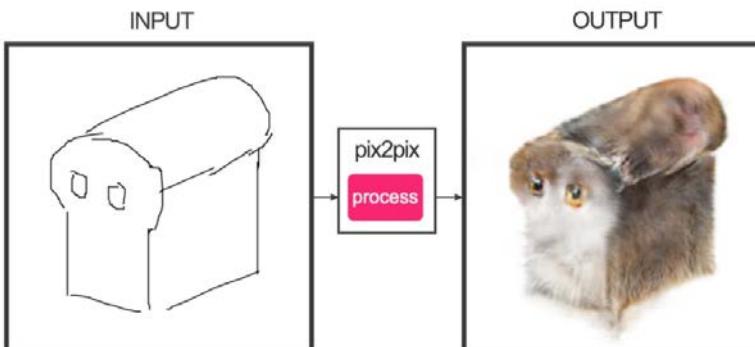
F1	MS-COCO	NUS-WIDE
VGG-16	56.0	33.9
+ GAN	60.4	41.2
Inception	62.4	53.5
+GAN	63.8	55.8
Resnet-101	62.8	53.1
+GAN	64.0	55.4
Resnet-152	63.3	52.1
+GAN	63.9	54.1
Att-RNN	62.1	54.7
RLSD	62.0	46.9

#edges2cats

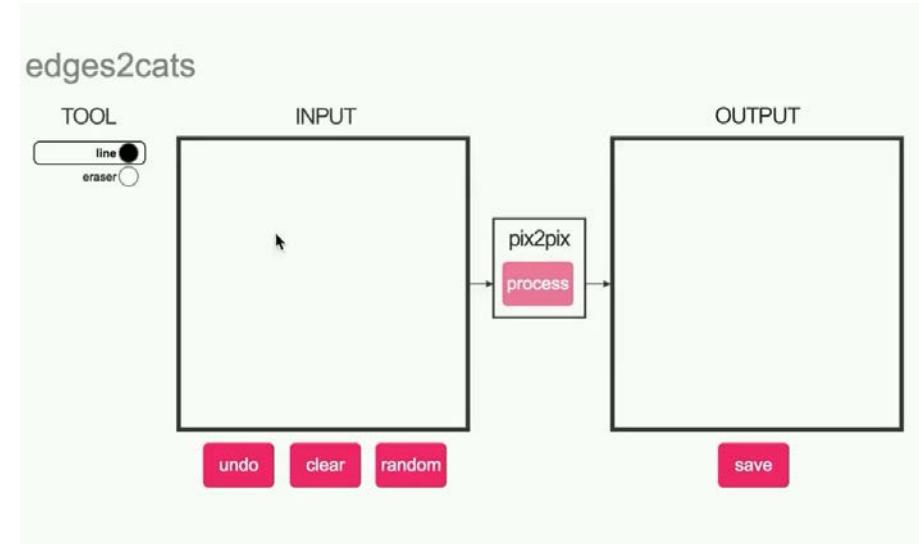
[Christopher Hesse]



@gods_tail



Ivy Tasi @ivymyt



@matthematician



Vitaly Vidmirov @vvid

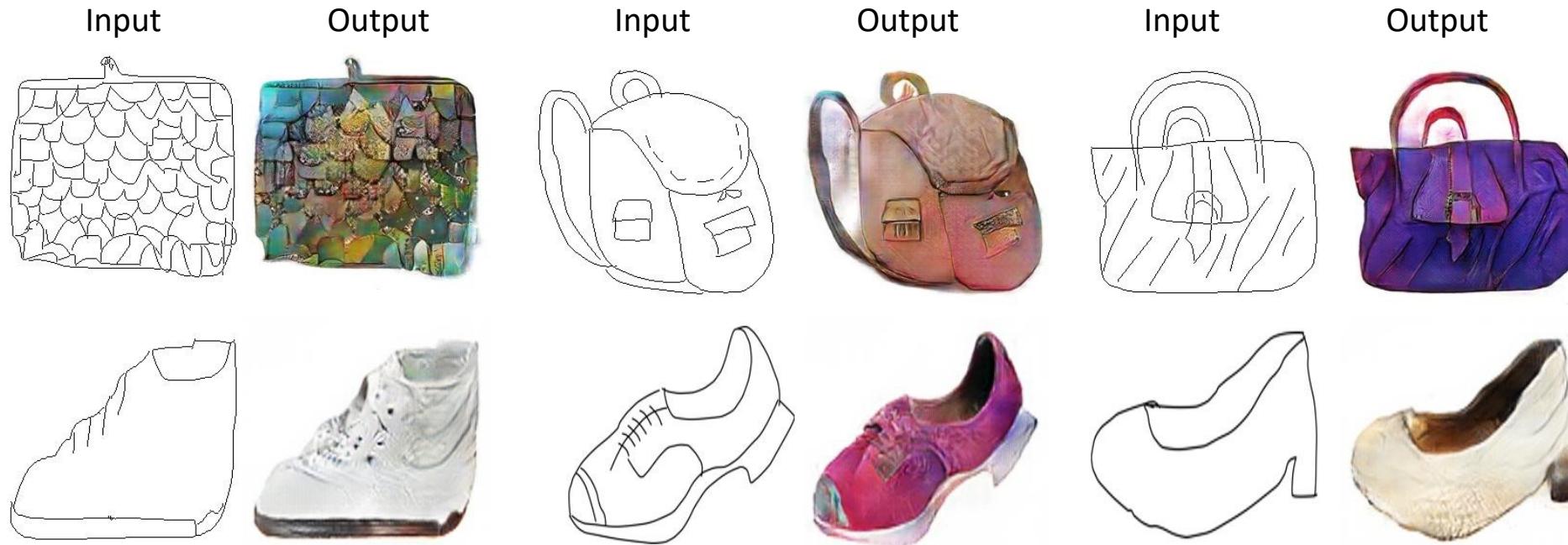
<https://affinelayer.com/pixsrv/>

Edges → Images



Edges from [Xie & Tu, 2015]

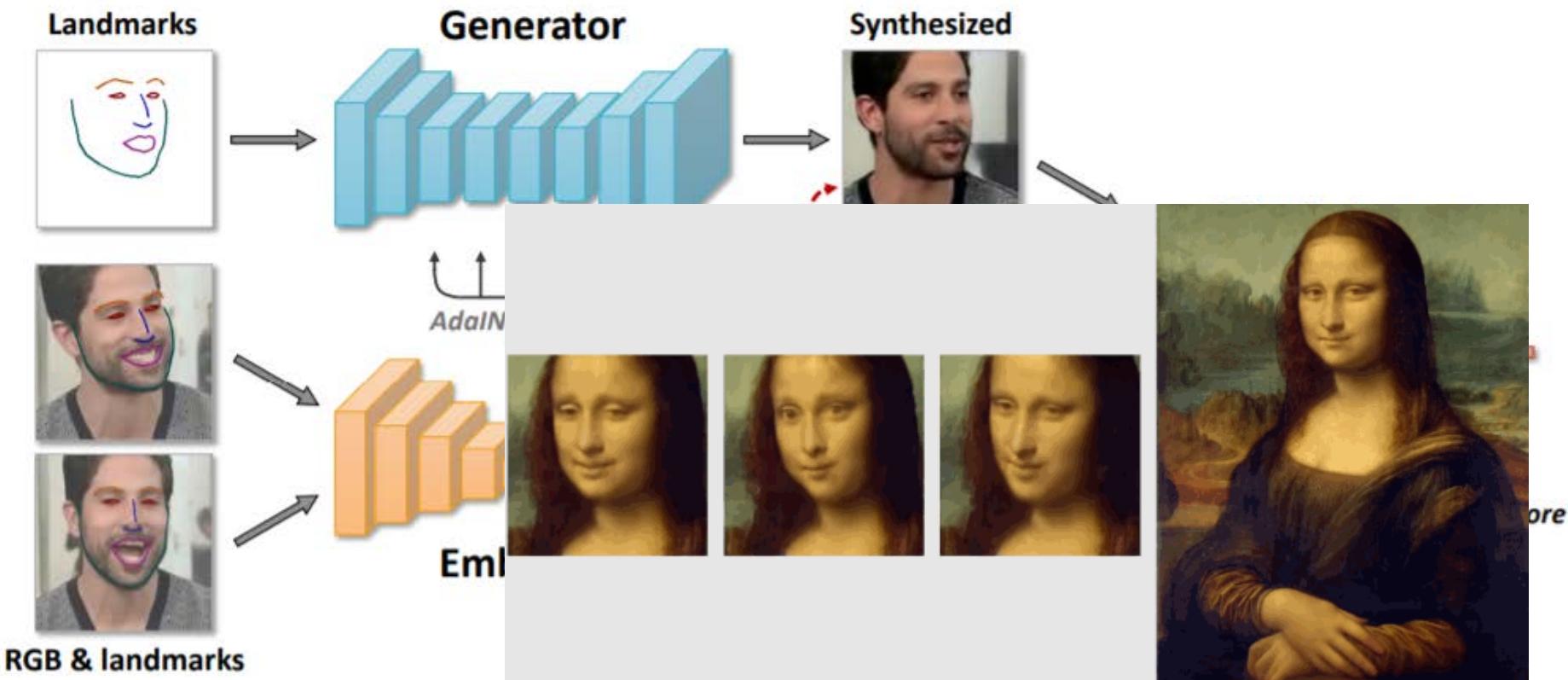
Sketches → Images



Trained on Edges → Images

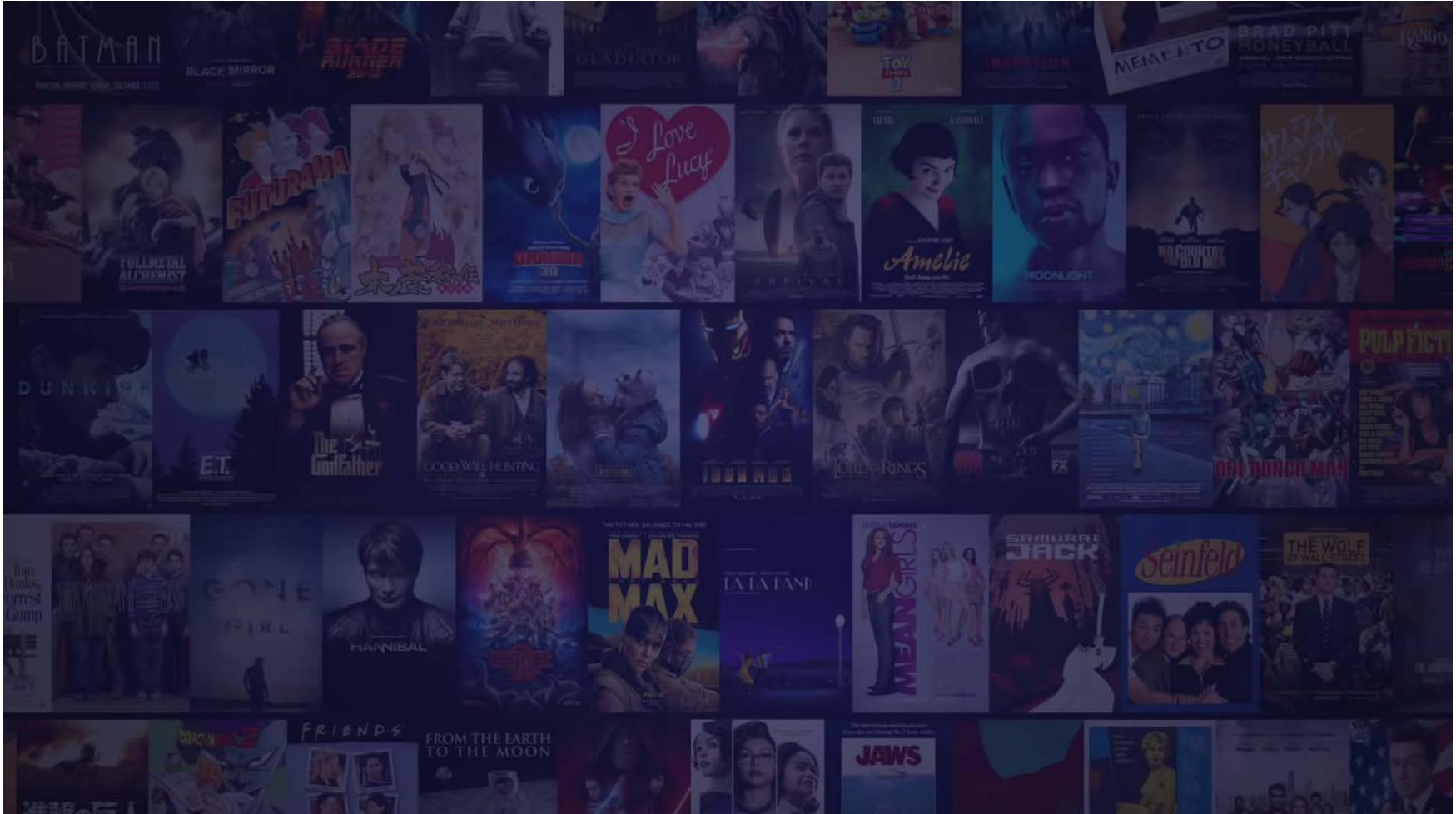
Data from [Eitz, Hays, Alexa, 2012]

Talking Head



<https://arxiv.org/abs/1905.08233>

Deepfake



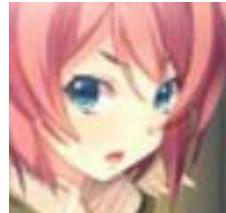
<https://www.youtube.com/watch?v=-QvIX3cY4lc>

Three Categories of GANs

1. Typical GAN



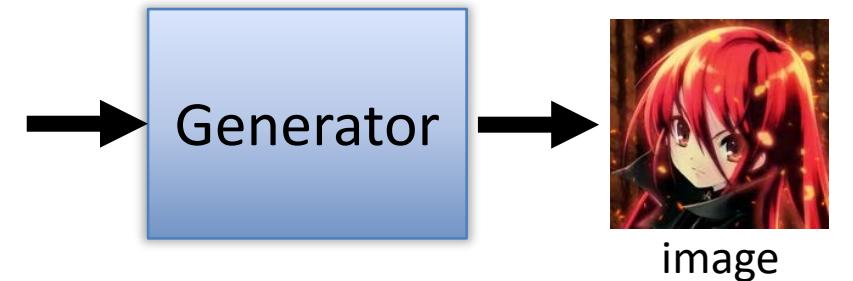
2. Conditional GAN



blue eyes,
red hair,
short hair

paired data

“Girl with
red hair”
text



3. Unsupervised Conditional GAN

domain x



domain y



unpaired data

x



Photo

Generator



Vincent van
Gogh's style

Cycle GAN



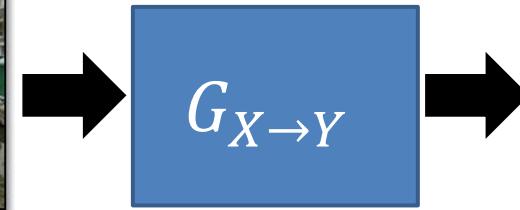
Domain X



Domain Y



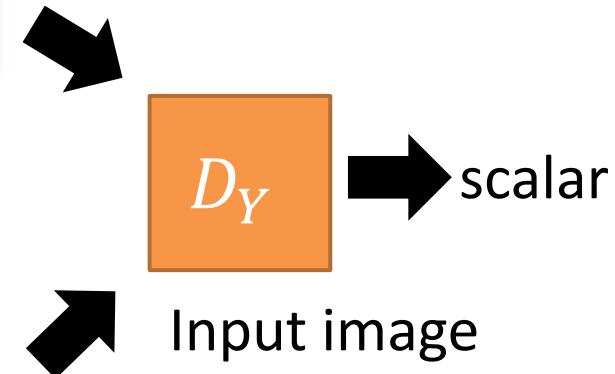
Domain X



Become similar
to domain Y



Domain Y



scalar

Input image
belongs to
domain Y or not

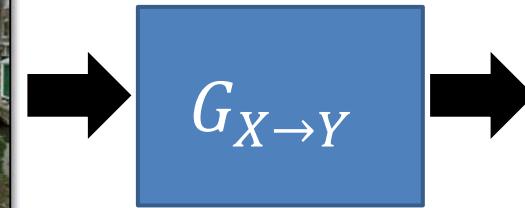
Cycle GAN



Domain X

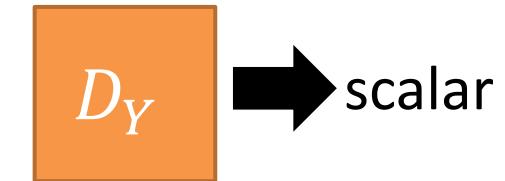


Domain Y



Become similar
to domain Y

Not what we want!



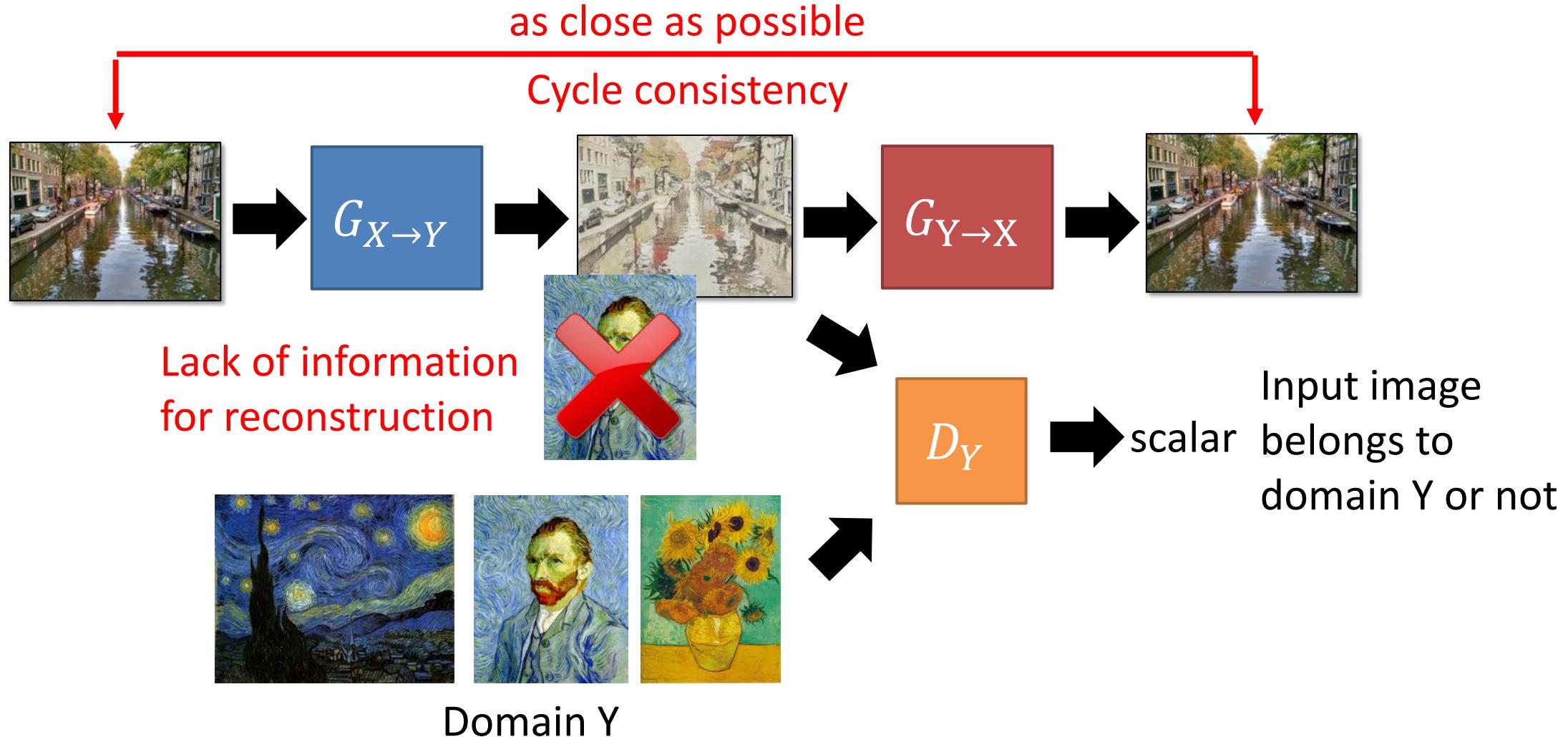
scalar

Input image
belongs to
domain Y or not

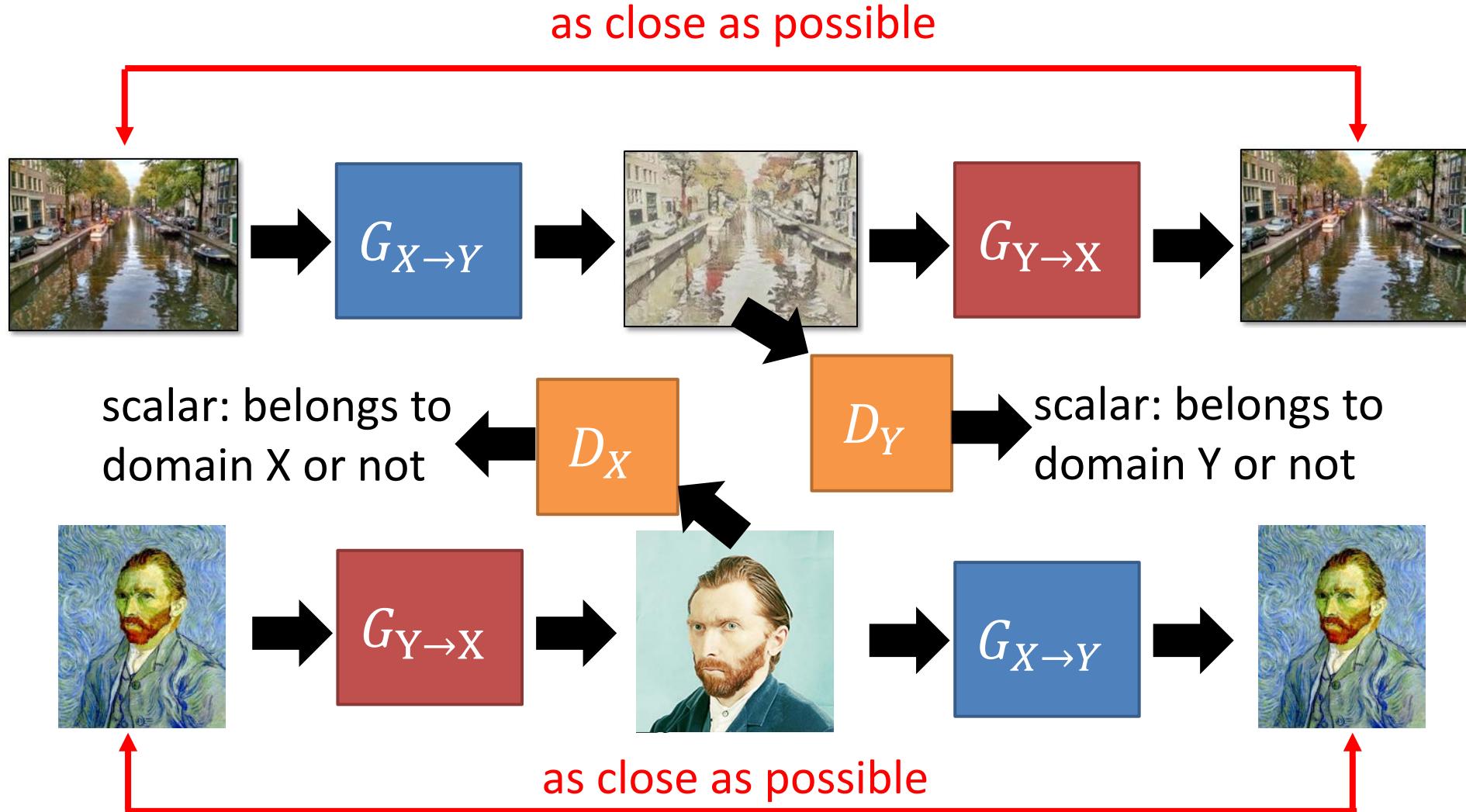


Domain Y

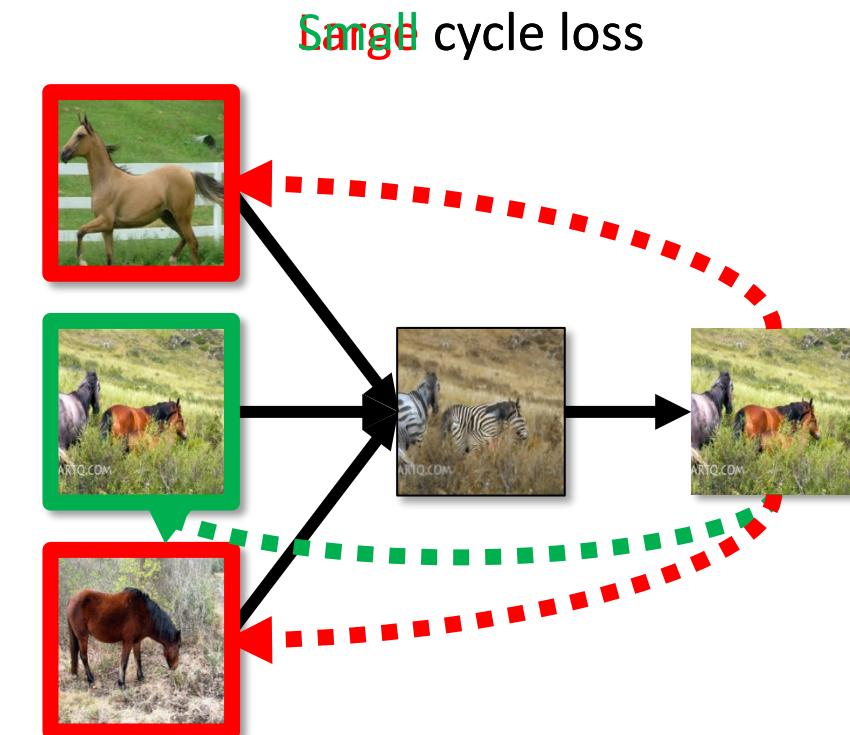
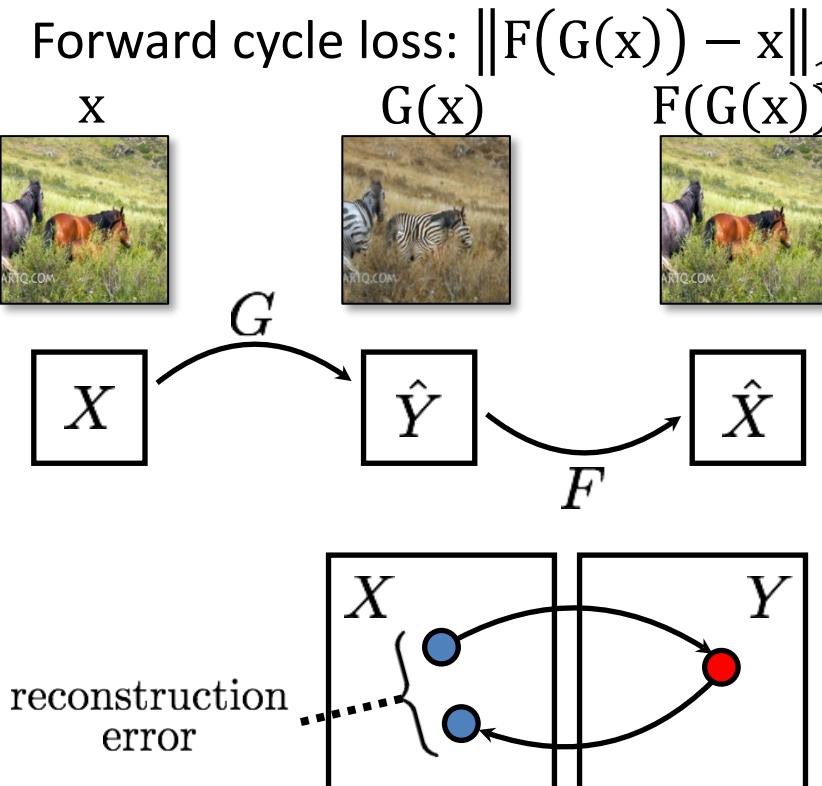
Cycle GAN



Cycle GAN



Cycle Consistency



Training Details: Objective

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = & \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log D_Y(y)] \\ & + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(1 - D_Y(G(x)))]\end{aligned}$$

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) = & \mathbb{E}_{x \sim p_{\text{data}}(x)} [\|F(G(x)) - x\|_1] \\ & + \mathbb{E}_{y \sim p_{\text{data}}(y)} [\|G(F(y)) - y\|_1].\end{aligned}$$

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = & \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ & + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ & + \lambda \mathcal{L}_{\text{cyc}}(G, F),\end{aligned}$$

$$G^*, F^* = \arg \min_{G, F} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y).$$





MI Project 2021-2022

- REVISITING ARTISTIC STYLE TRANSFER FOR DATA AUGMENTATION IN A REAL-CASE SCENARIO
Stefano D'Angelo, Frederic Precioso, Fabien Gandon, IEEE ICIP 2022.

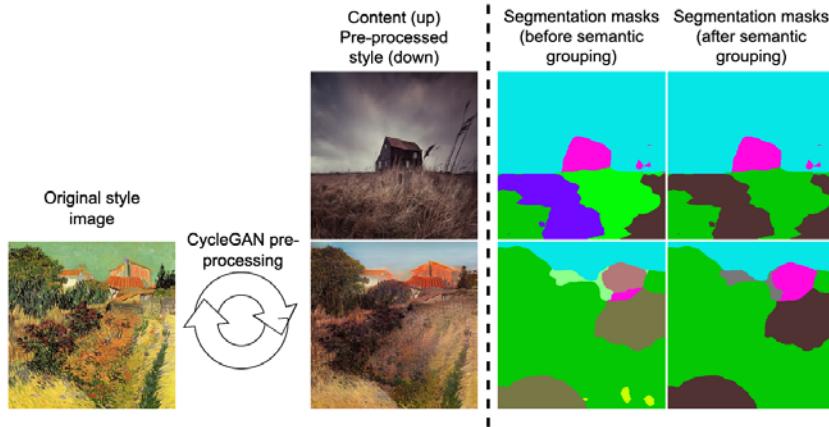


Fig. 1. Visual summary of the pre-processing phase.

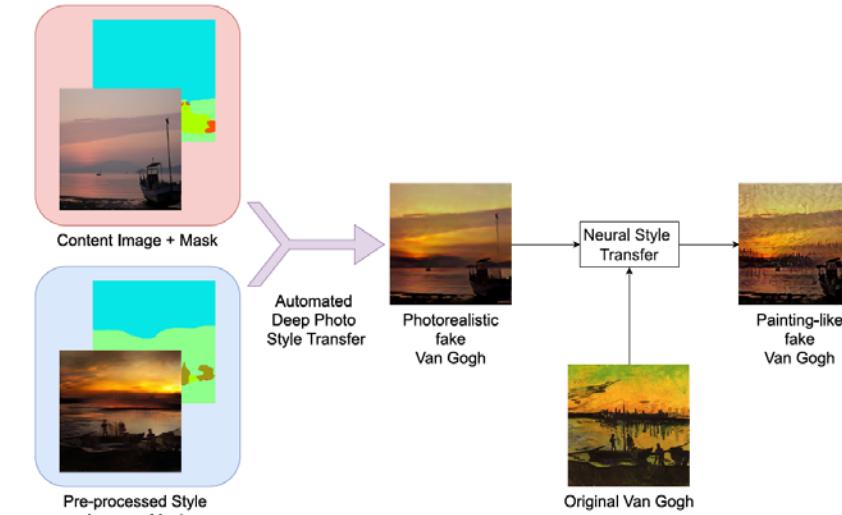


Fig. 2. Schema of the architecture.

MI Project 2021-2022

- REVISITING ARTISTIC STYLE TRANSFER FOR DATA AUGMENTATION IN A REAL-CASE SCENARIO
Stefano D'Angelo, Frederic Precioso, Fabien Gandon, IEEE ICIP 2022.



Fig. 3. Some of the results compared to three state-of-the-art models.

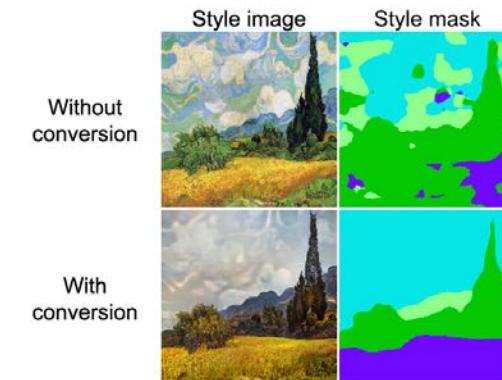


Fig. 4. Benefits of painting to photo conversion.

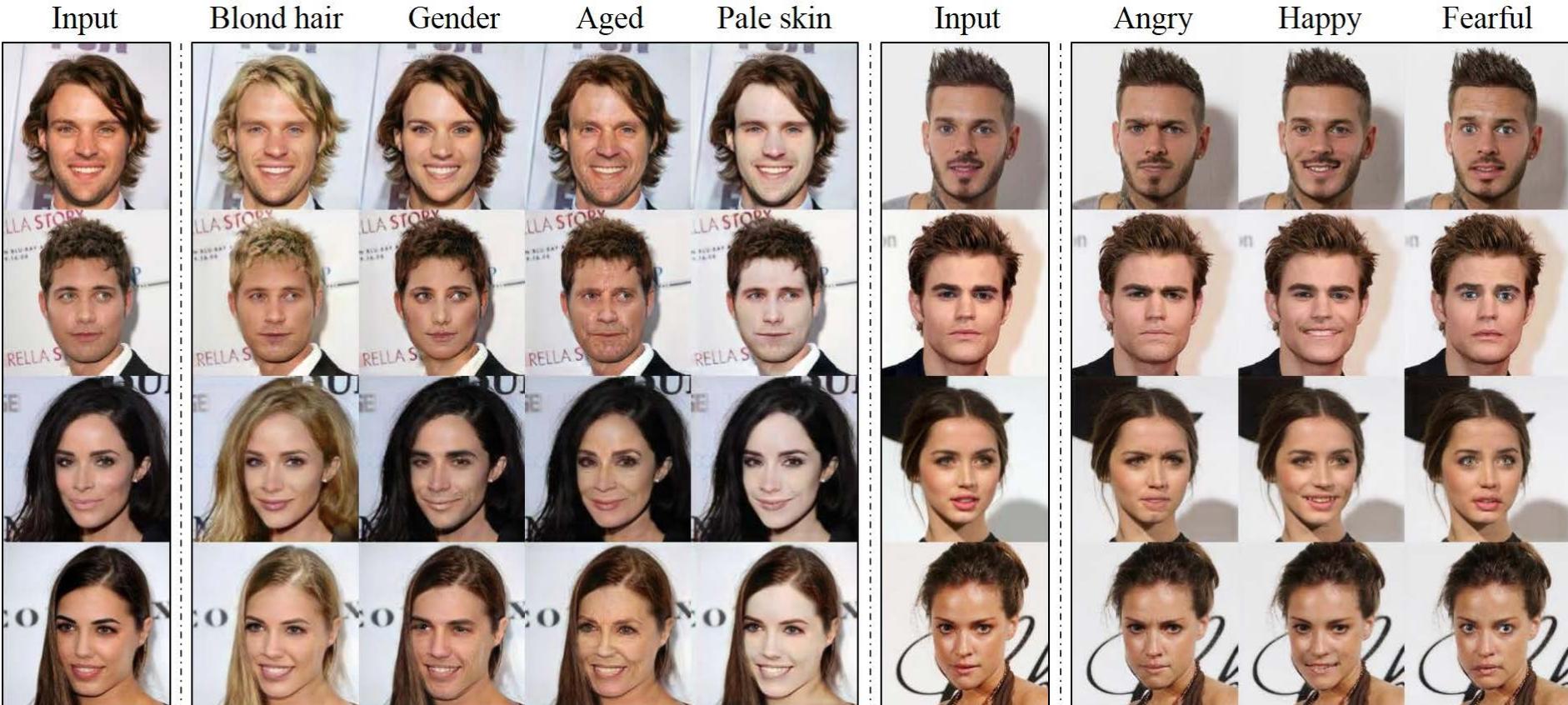
Celebrities Who Never Existed



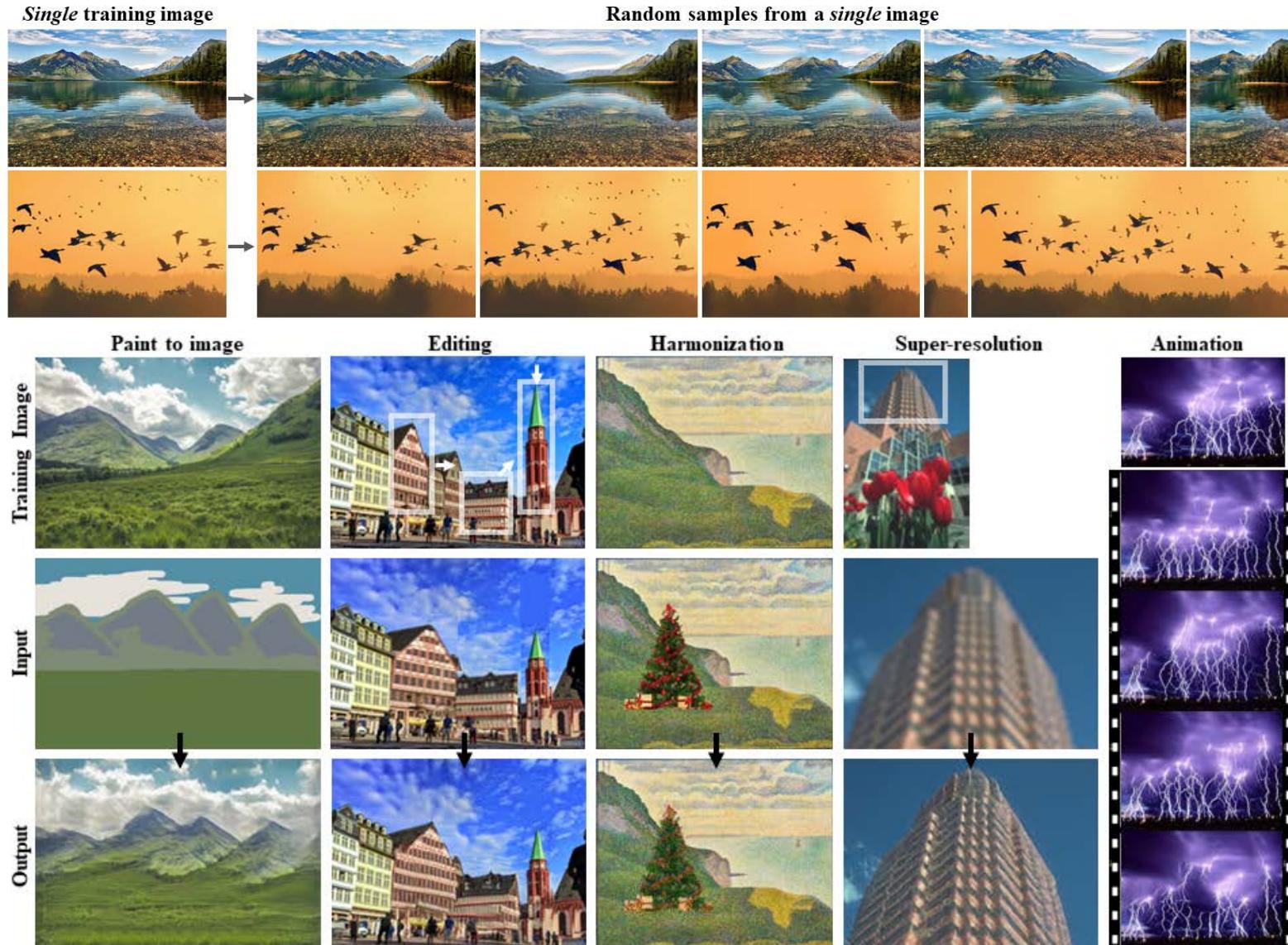
Creative Adversarial Networks



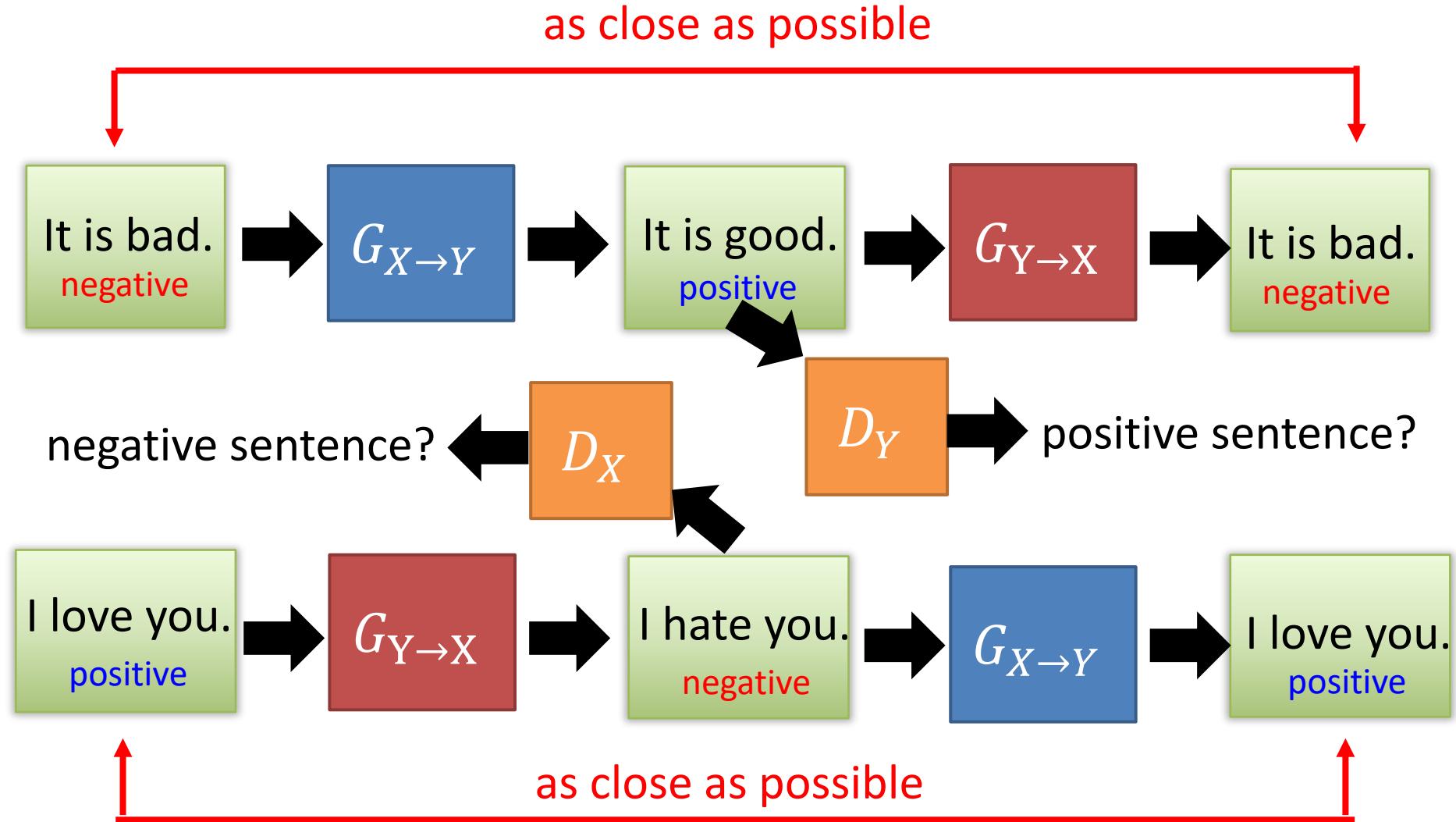
(Elgammal et al., 2017)



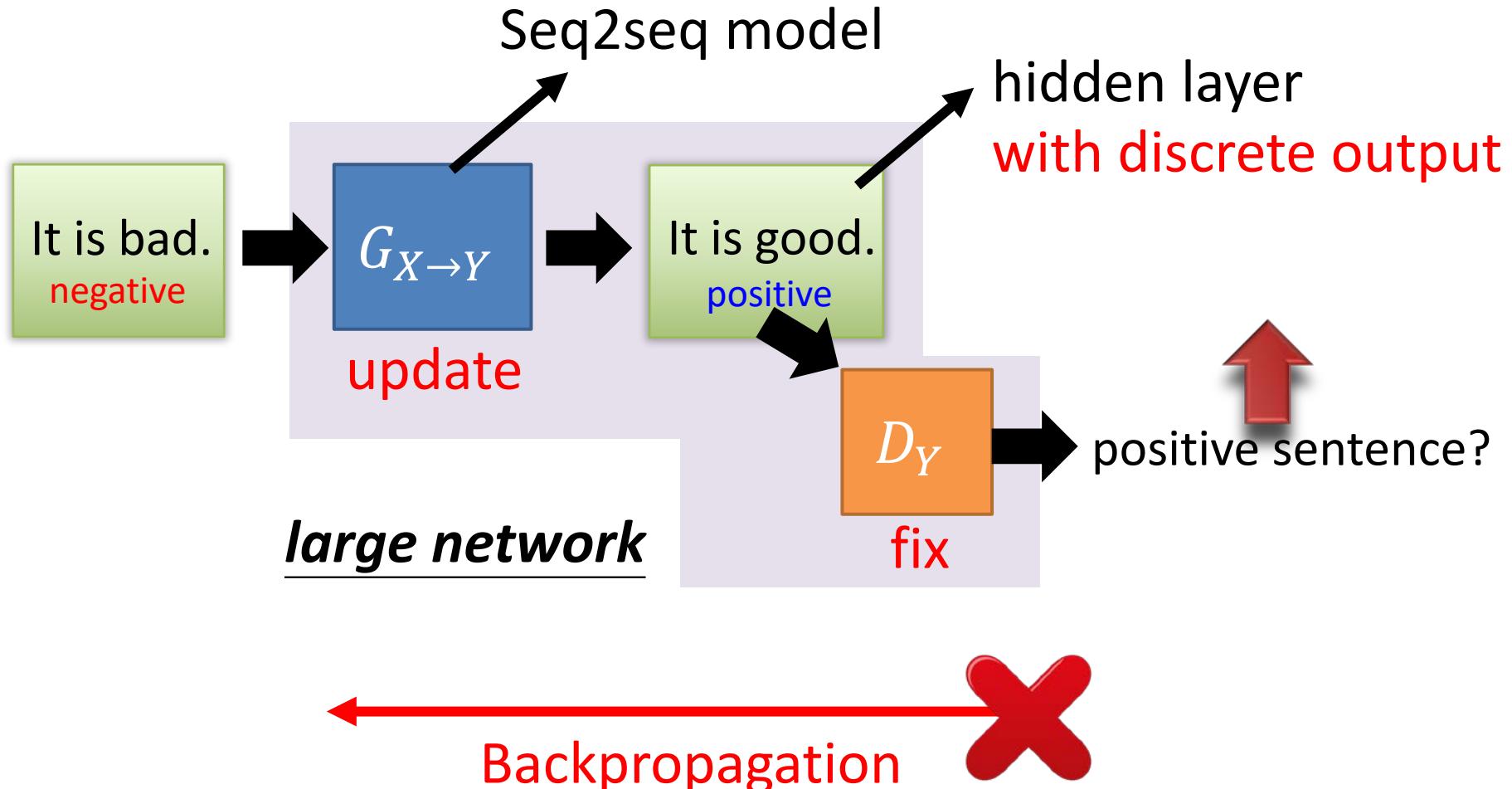
SinGAN



Cycle GAN



Discrete Issue



Three Categories of Solutions

Gumbel-softmax

- [Matt J. Kusner, et al, arXiv, 2016]

Continuous Input for Discriminator

- [Sai Rajeswar, et al., arXiv, 2017][Ofir Press, et al., ICML workshop, 2017][Zhen Xu, et al., EMNLP, 2017][Alex Lamb, et al., NIPS, 2016][Yizhe Zhang, et al., ICML, 2017]

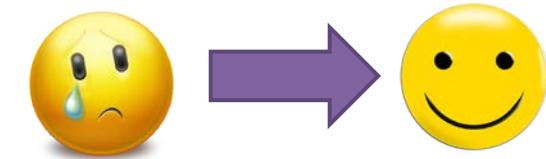
“Reinforcement Learning”

- [Yu, et al., AAAI, 2017][Li, et al., EMNLP, 2017][Tong Che, et al, arXiv, 2017][Jiaxian Guo, et al., AAAI, 2018][Kevin Lin, et al, NIPS, 2017][William Fedus, et al., ICLR, 2018]

Text rewriting

X Negative sentence to positive sentence:

- it's a crappy day -> it's a great day
i wish you could be here -> you could be here
it's not a good idea -> it's good idea
i miss you -> i love you
i don't love you -> i love you
i can't do that -> i can do that
i feel so sad -> i happy
it's a bad day -> it's a good day
it's a dummy day -> it's a great day
sorry for doing such a horrible thing -> thanks for doing a great thing
my doggy is sick -> my doggy is my doggy
my little doggy is sick -> my little doggy is my little doggy



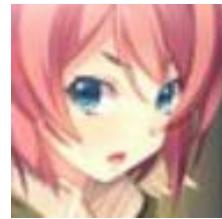


Three Categories of GANs

1. *Typical GAN*

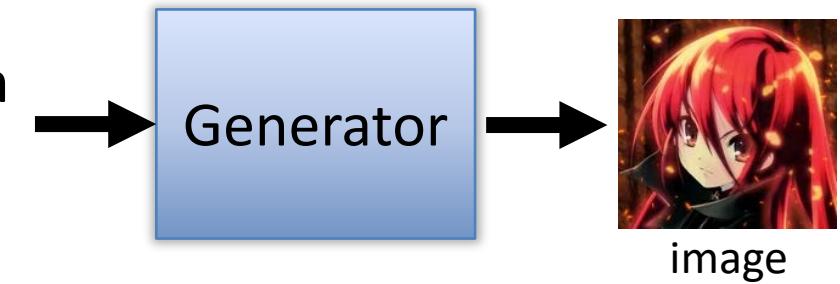


2. *Conditional GAN*



blue eyes,
red hair,
short hair
paired data

“Girl with
red hair”
text



3. *Unsupervised Conditional GAN*

domain x



domain y



x



Photo

Generator



y
Vincent van
Gogh's style

unpaired data



Ethics (Politics, Privacy)

Ethics (Politics, Privacy)

- Deep fakes
- Privacy
- Security and adversarial perturbations

Tips & Tricks

- Feature matching (facial for instance, see deepfake)



Deepfake

“Deepfakes”



<https://www.technologyreview.com/s/611726/the-defense-department-has-produced-the-first-tools-for-catching-deepfakes/>
<https://www.niemanlab.org/2018/11/how-the-wall-street-journal-is-preparing-its-journalists-to-detect-deepfakes/>

Full person!!!



Deepfake

<https://information.tv5monde.com/video/deepfake-de-tom-cruise-peut-encore-croire-ce-que-l-voit-vrai-dire>

https://www.youtube.com/watch?v=SDty803-hxg&ab_channel=Le HuffPost

<https://edition.cnn.com/videos/business/2021/03/02/tom-cruise-tiktok-deepfake-orig.cnn-business/video/playlists/business-misinformation/>

You can be anyone you want...



Detection methods

FaceForensics++: Learning to Detect Manipulated Facial Images

Andreas Rössler¹ Davide Cozzolino² Luisa Verdoliva² Christian Riess³
Justus Thies¹ Matthias Nießner¹

¹Technical University of Munich ²University Federico II of Naples ³University of Erlangen-Nuremberg

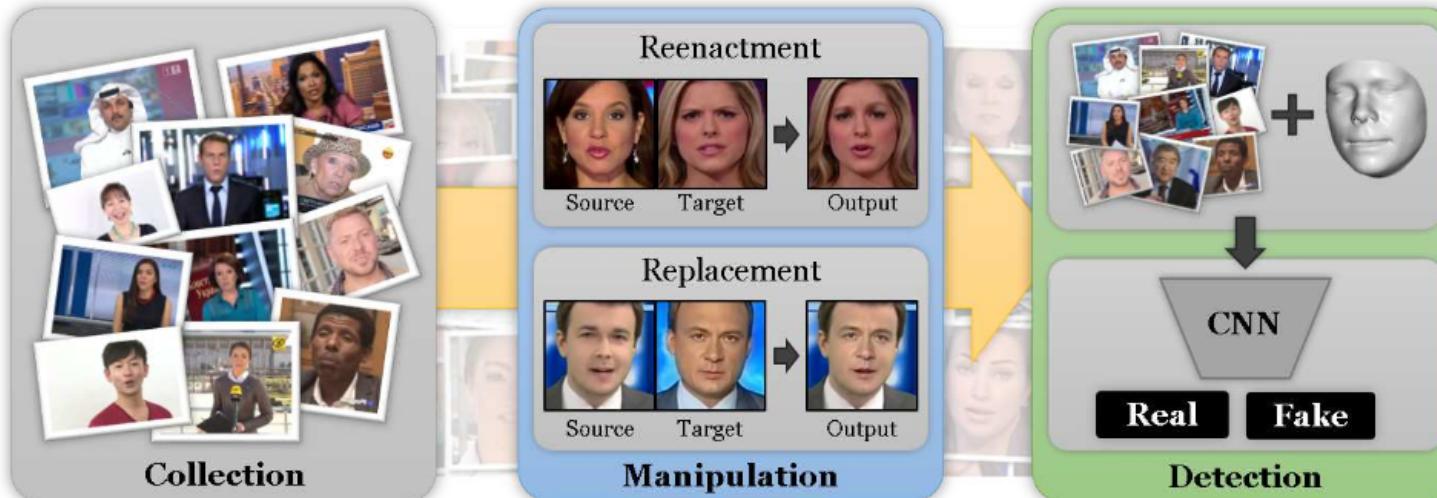


Figure 1: *FaceForensics++* is a dataset of facial forgeries that enables researchers to train deep-learning-based approaches in a supervised fashion. The dataset contains manipulations created with four state-of-the-art methods, namely, *Face2Face*, *FaceSwap*, *DeepFakes*, and *NeuralTextures*.

Detection methods

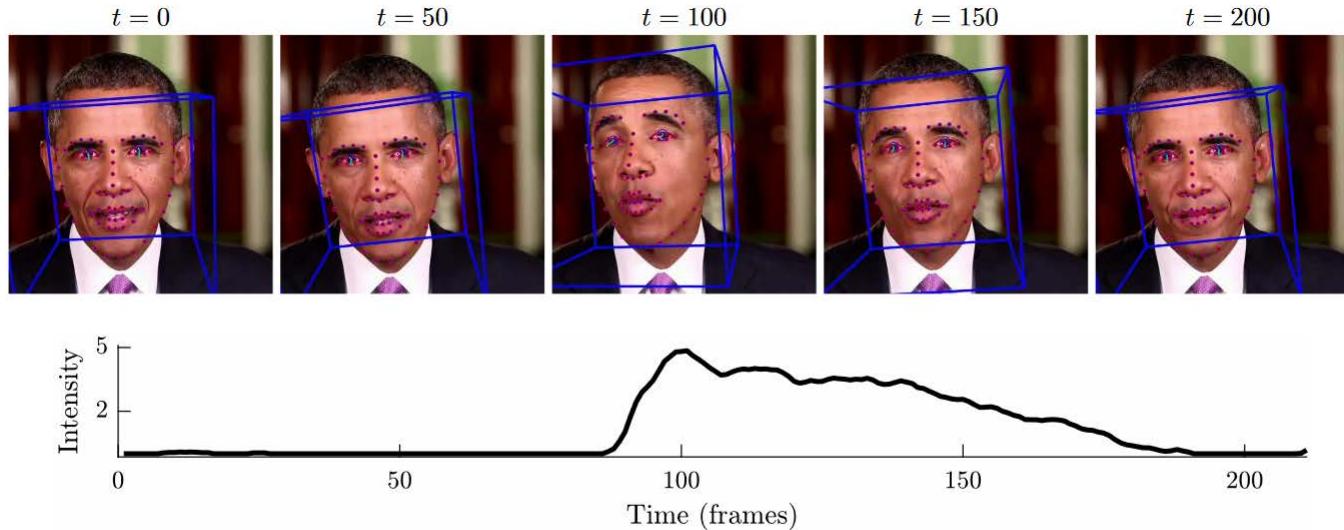


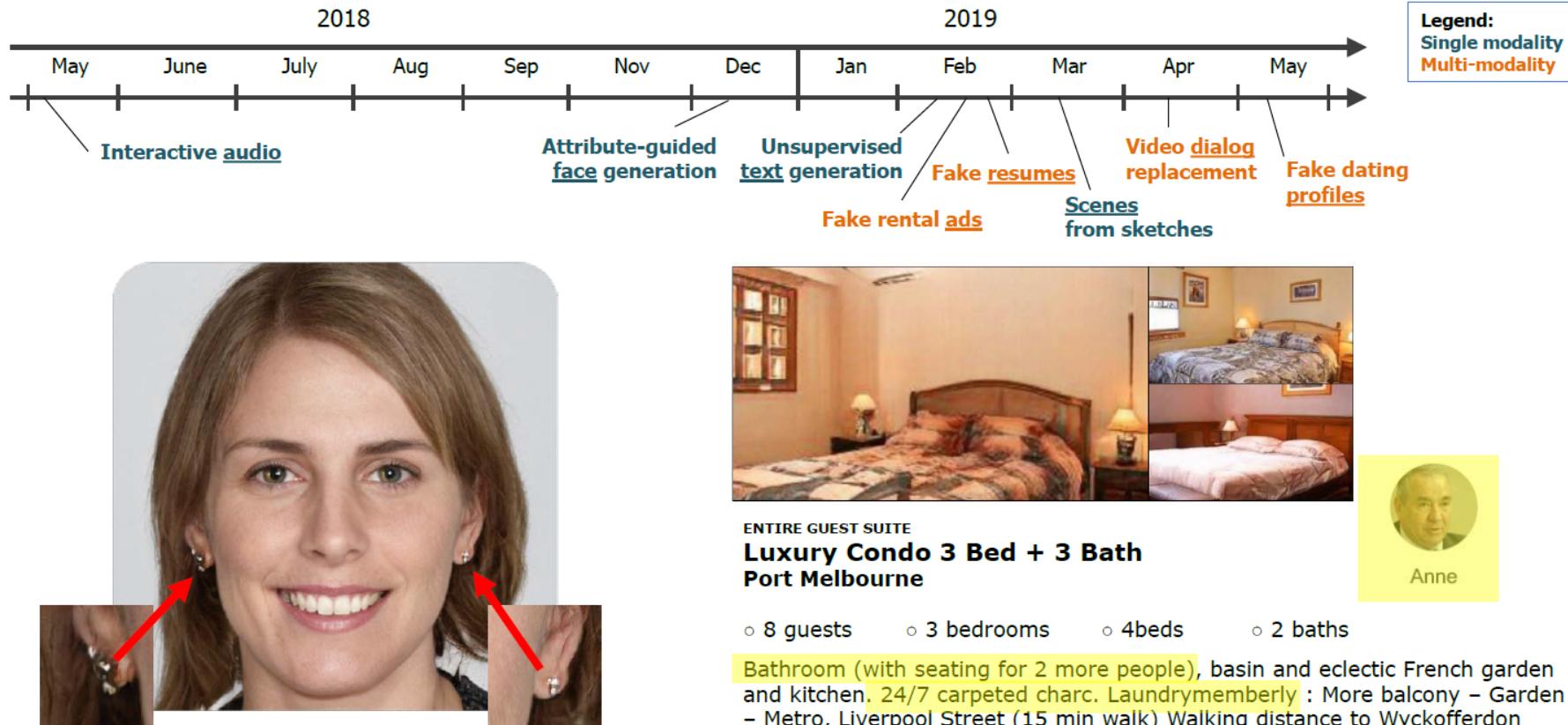
Figure 1. Shown above are five equally spaced frames from a 250-frame clip annotated with the results of OpenFace tracking. Shown below is the intensity of one action unit AU01 (eye brow lift) measured over this video clip.

“We describe a forensic technique that models facial expressions and movements that typify an individual’s speaking pattern. Although not visually apparent, these correlations are often violated by the nature of how deep-fake videos are created and can, therefore, be used for authentication.

Successes



Incredible Pace of Synthetic Media Generation

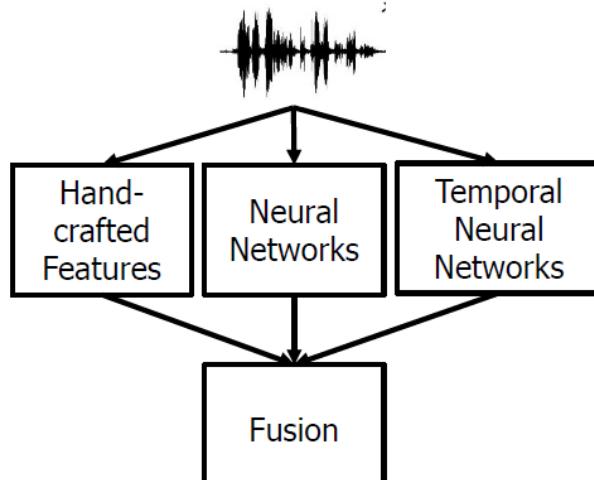


Successes



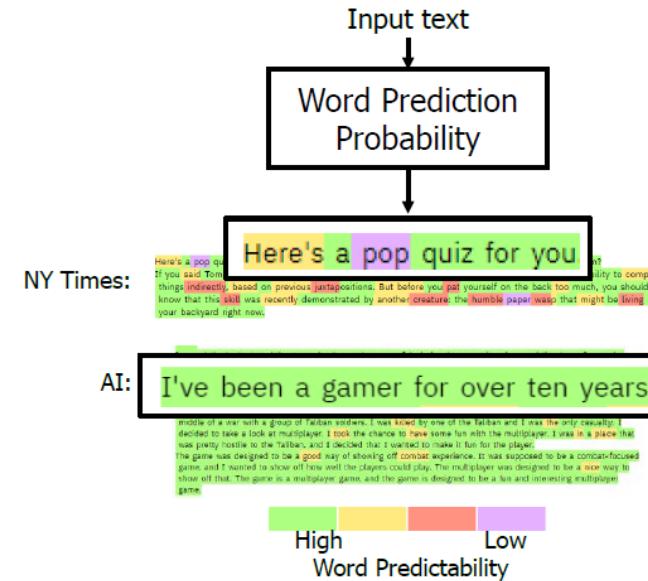
State of the Art Detection is Statistically Based, Narrow, or Both

Audio: ASVspoof



(Lavrentyeva et al. 2017)

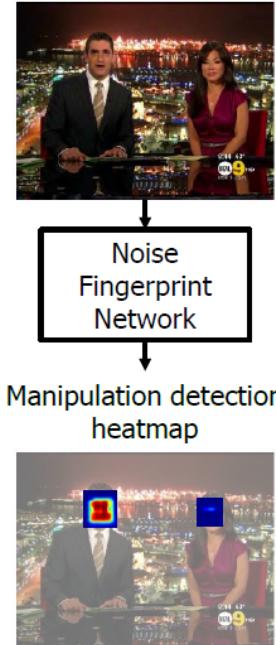
Text: GLTR



AI methods choose more predictable next-words than humans, statistically

(MIT-IBM Watson AI lab, HarvardNLP 2019)

Image/Video: DARPA MediFor



(MediFor: USC/ISI, Univ. Naples 2019)



Expected Threats



Expected Threats

Targeted Personal Attacks

Peele 2017

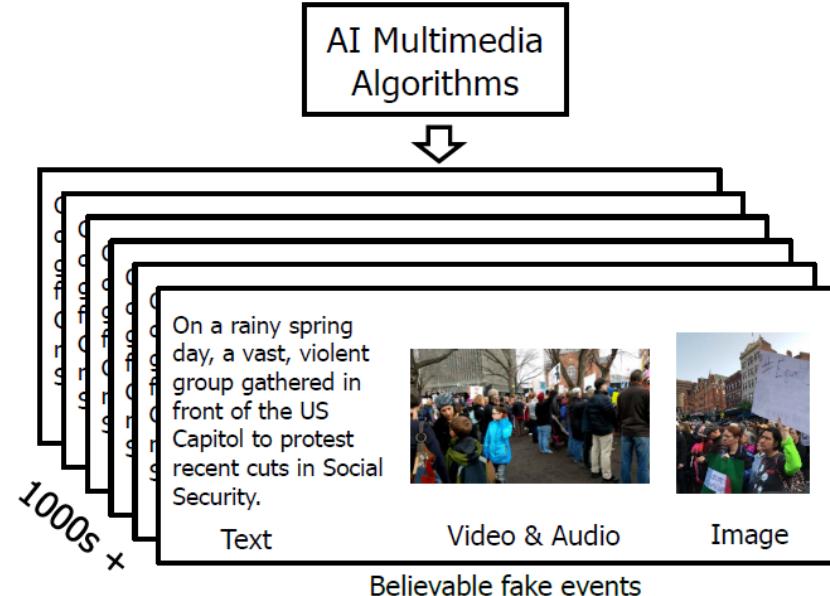


AI Multimedia
Algorithms



Highly realistic video

Generated Events at Scale



Ransomfake concept: Identity Attacks as a service (IAaaS)

Bricman 2019

AI Multimedia
Algorithms

Forged
Evidence

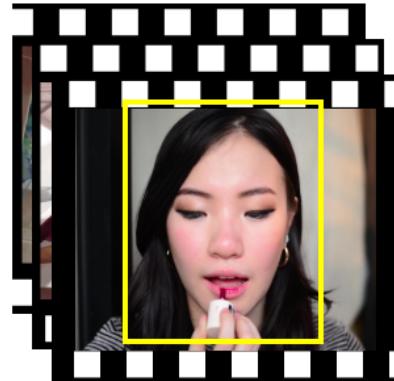
Identity
Attacks

Examples of possible fakes:

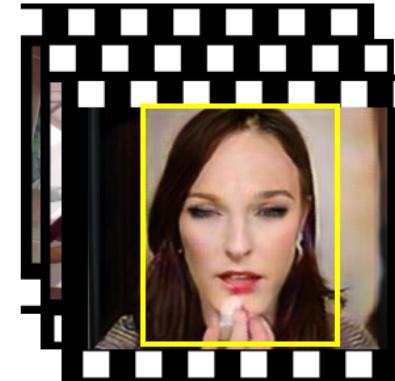
- Substance abuse
- Foreign contacts
- Compromising events
- Social media postings
- Financial inconsistencies
- Forging identity

Undermines key individuals and organizations

GANs for Privacy (Action Detection)



Identity: Jessica
Action: Applying Make-up on Lips



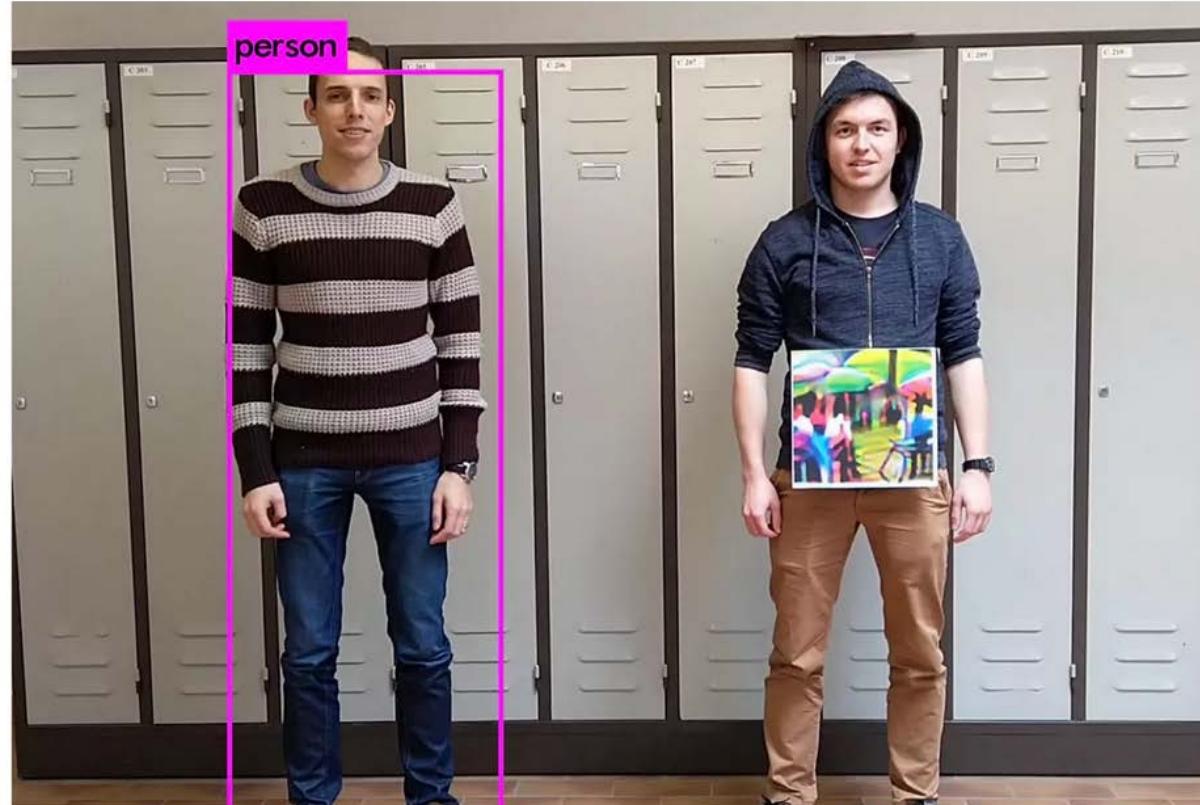
Identity: ???
Action: Applying Make-up on Lips



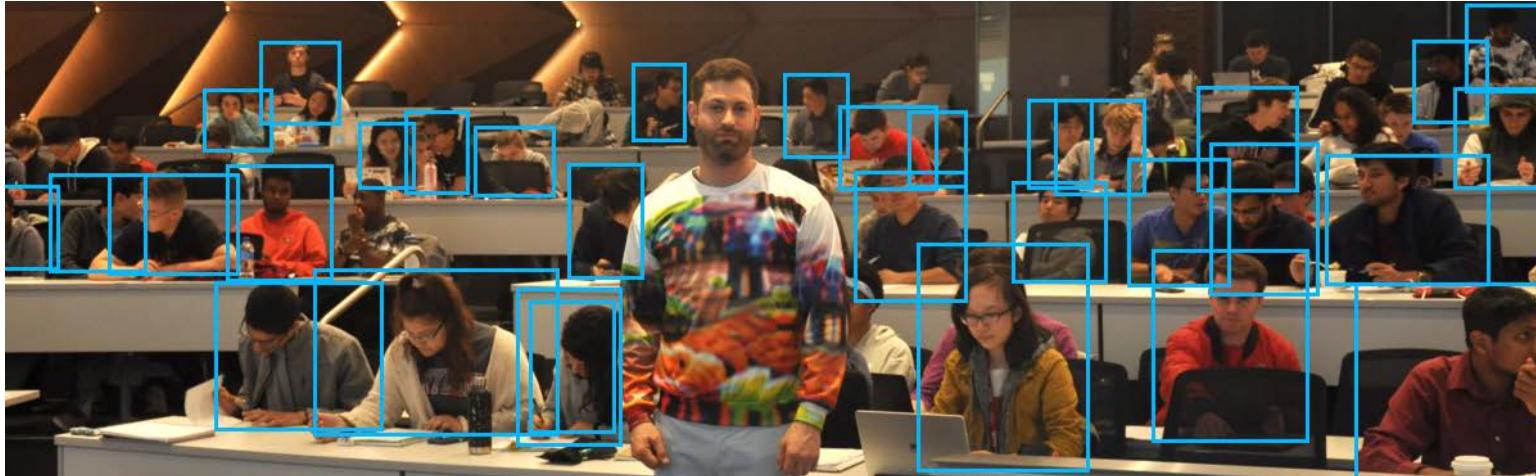
Adversarial Attacks



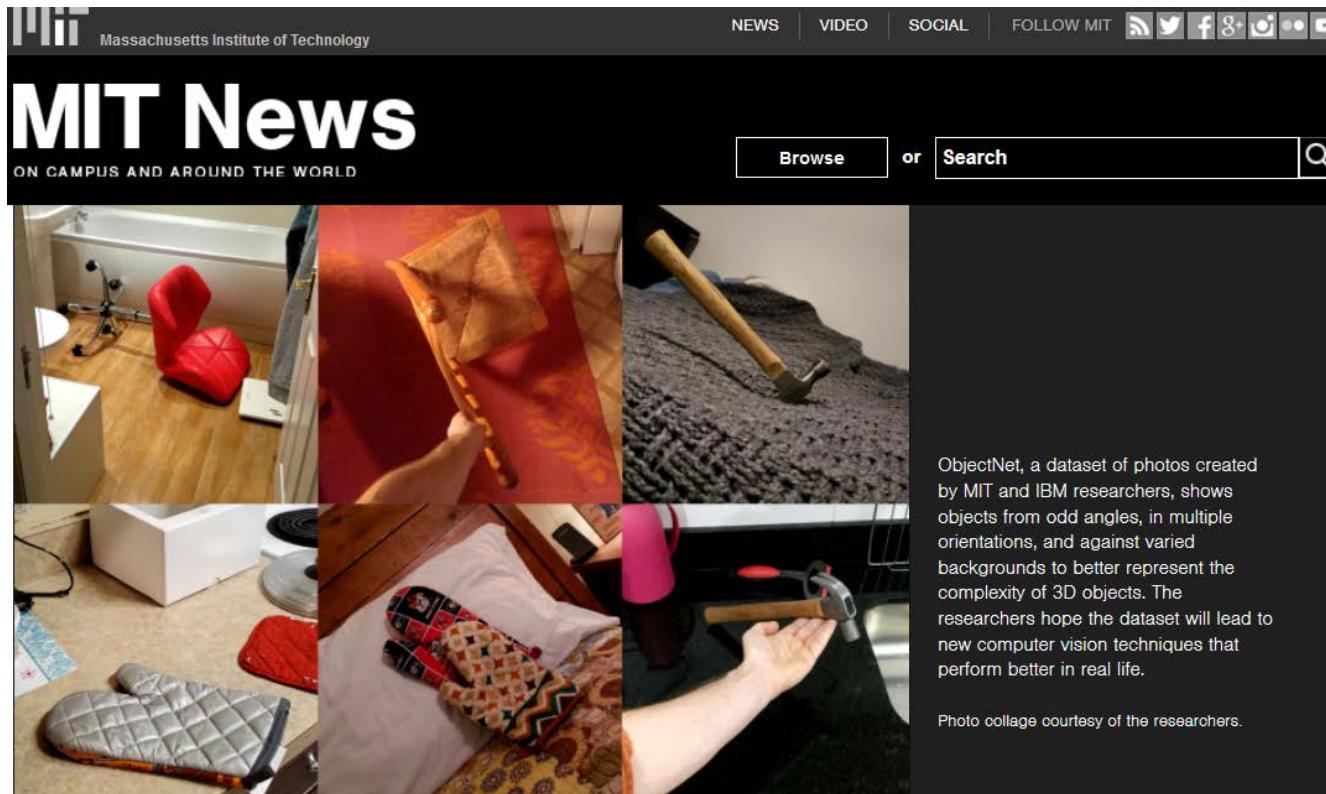
Adversarial Attacks



Adversarial Attacks

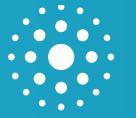


Adversarial Attacks



This object-recognition dataset stumped the world's best computer vision models

Objects are posed in varied positions and shot at odd angles to spur new AI techniques.



UNIVERSITÉ
CÔTE D'AZUR