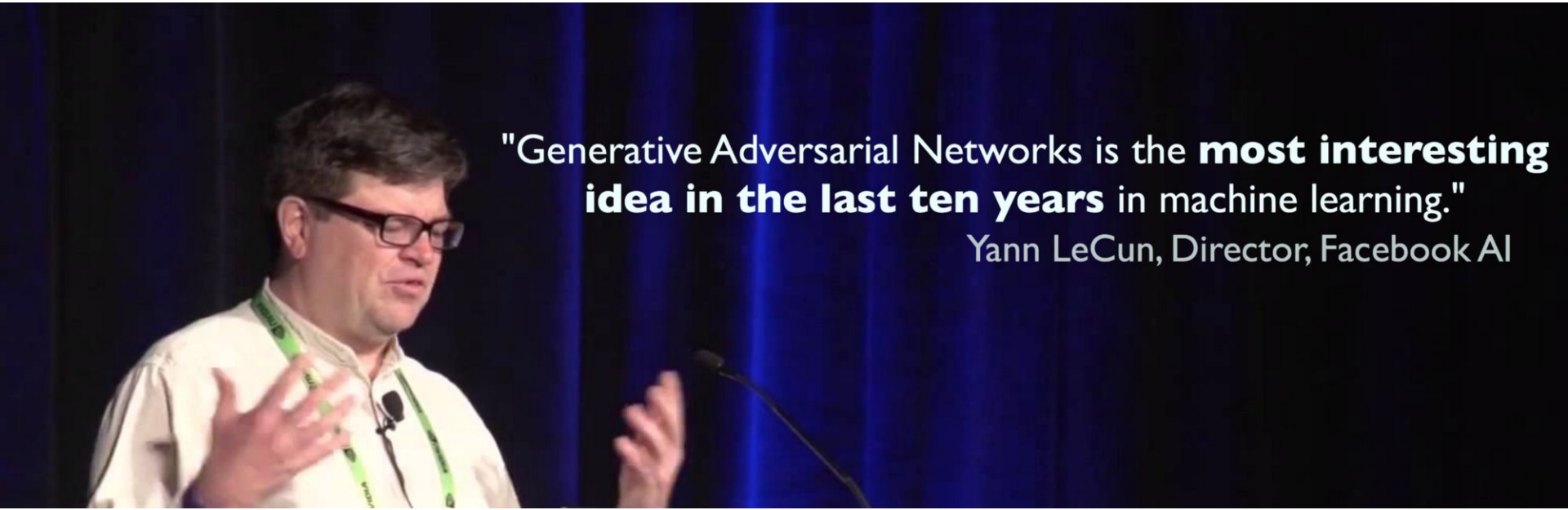
Generative Adversarial Networks (GAN)

University of Victoria - PHYS-555

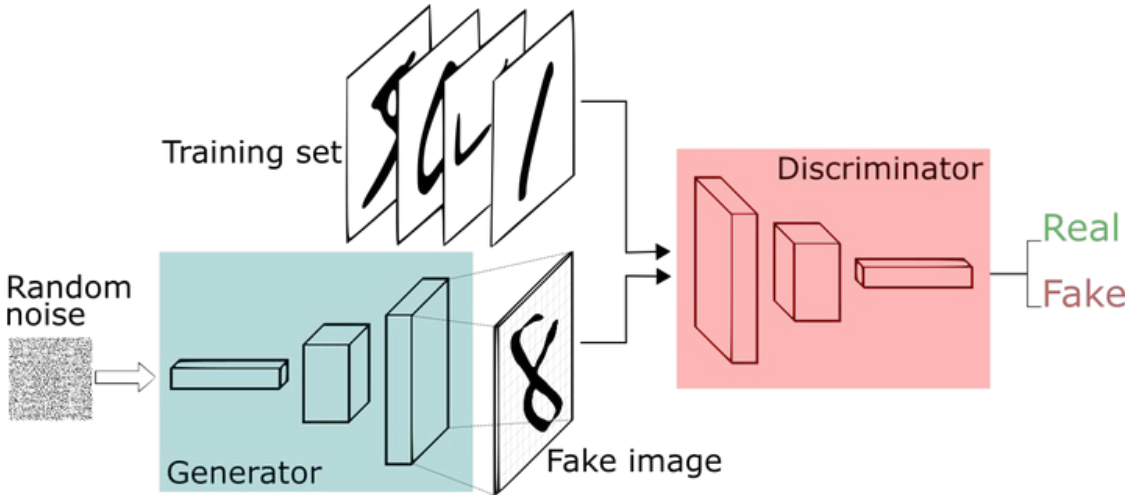
A long long time ago (2016)



GAN Architecture

Two components: the **generator** and the **discriminator:**

- The **generator** *G* will capture the data distribution *p*(**x**)
- The **discriminator** *D* gives the probability that a sample was from training data rather than *G*

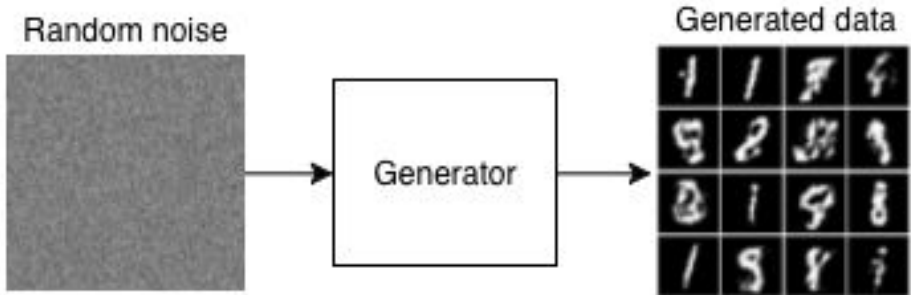


GAN Component: Generator

- Sample from a prior distribution

$$\mathbf{z} \sim \mathcal{N}(0, 1)$$

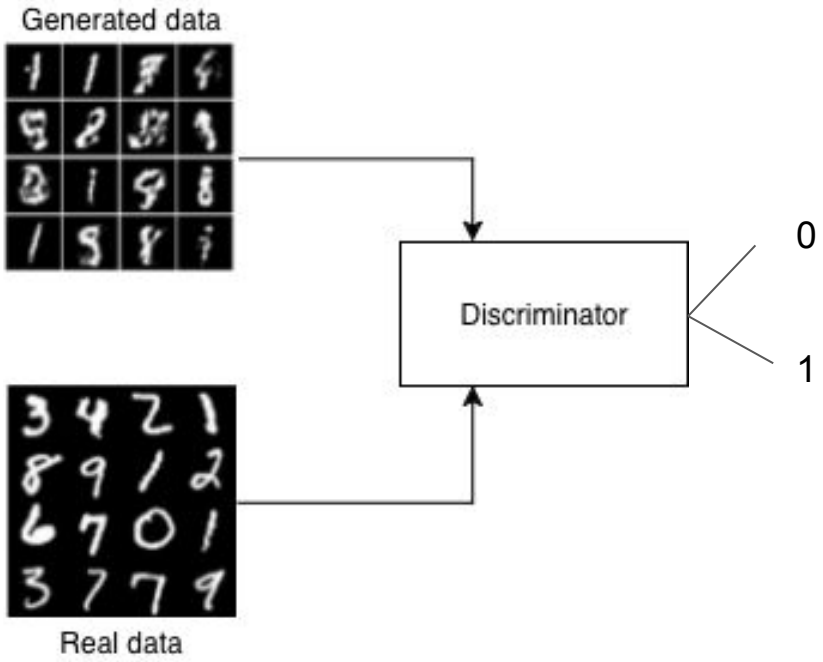
- Generate data with a NN. Original paper: generator was an MLP



$$\mathbf{x} = G(\mathbf{z})$$

GAN Component: Discriminator = Teacher = Critic

- Take input either from a training data sample or a generated sample
- Simplest one is just a NN classifier



GAN Training Game

Two-player zero-sum game between two networks:

- The generator network *G* which maps from a prior distribution of the latent space to the data space such as $\mathbf{z} \sim p(\mathbf{z}), \hat{\mathbf{x}} = G(\mathbf{z}; \theta)$.

- The discriminator network *D* is basically a classifier which learns to separate true samples **x** from generated ones *G*(**z**)

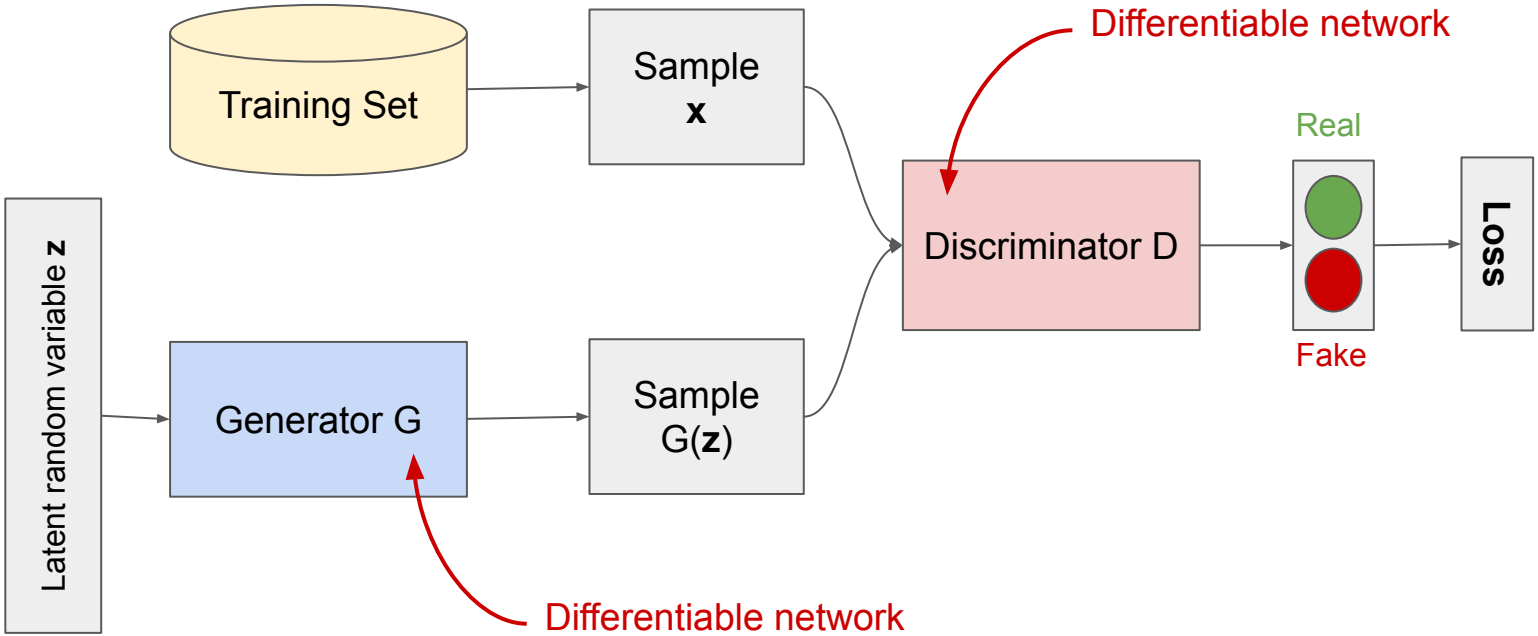
Use supervised learning to guide the unsupervised generator!

GAN Training Game

Two-player zero-sum gar

- The generator netwo istribution of the latent space to the data sp $(\mathbf{z}; \theta)$.
- The discriminator ne which learns to separate true samples **x** from

Use supervised learning erator!



$$\min_{\theta} \max_{\phi} \underbrace{\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \left[\log D(\mathbf{x};\phi)\right] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \left[\log(1 - D(G(\mathbf{z};\theta);\phi))\right]}_{V(\phi,\theta)}$$

GAN Learning Process

We attain the solution by alternating two SGDs on the value function *V*:

$$\theta \leftarrow \theta - \gamma \nabla_\theta V(\phi, \theta)$$

$$\phi \leftarrow \phi + \gamma \nabla_\phi V(\phi, \theta)$$

For one step on θ, we take N steps on ϕ (we want *D* to be optimal)

GAN Learning Process

We attain the solution by alternating two SGDs on the value function V:

$$\theta \leftarrow \theta - \gamma \nabla_\theta V(\phi, \theta)$$

backprop all the way!

$$\phi \leftarrow \phi + \gamma \nabla_\phi V(\phi, \theta)$$

For one step on θ, we take N steps on ϕ (we want D to be optimal)

GAN Training Algorithm

for number of training iterations **do**

  for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ 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

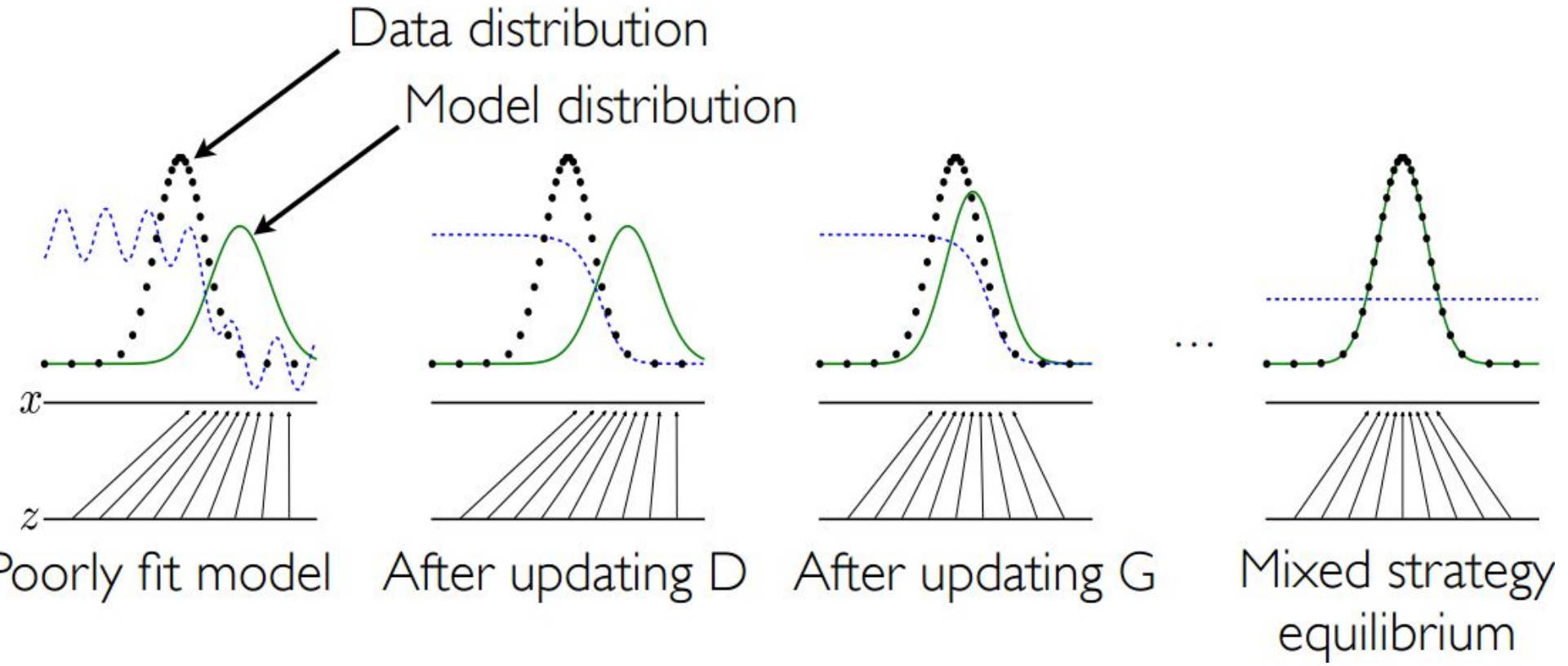
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

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

end for

Algorithm from the original 2014 paper.

Today lots of other tricks but main idea is the same.


Data distribution
Model distribution
x
z
...
Poorly fit model
After updating D
After updating G
Mixed strategy equilibrium

GAN Game Analysis

- For a fixed generator *G*, *V* is high if *D* is good at discriminating
- If *D* is the best discriminator given *G* with *V* high ⇒ *G* is not good
- But *G* is a good generator is *V* is low and *D* is a good opponent.

We want to reach the solution: $$\theta^* = \min_{\theta} \max_{\phi} V(\phi, \theta)$$

The optimization converges when the Nash equilibrium (game theory concept) is reached in the minimax zero-sum game.

Check the GAN Live Training Demo

GAN Garanty Proof 1

Tentative proof sketch that the GAN process *could* give an estimate of the true data distribution.

For a generator *G* fixed at θ, the classifier *D* is optimal if and only if:

$$D(\mathbf{x}; \phi_\theta^*) = \frac{p(\mathbf{x})}{q(\mathbf{x};\theta)+p(\mathbf{x})}.$$

where:

$$p(\mathbf{x}) = p_{\text{data}}(\mathbf{x}) \qquad q(\mathbf{x}, \theta) = p_{\text{model}}(\mathbf{x}; \theta)$$

GAN Garanty Proof 2

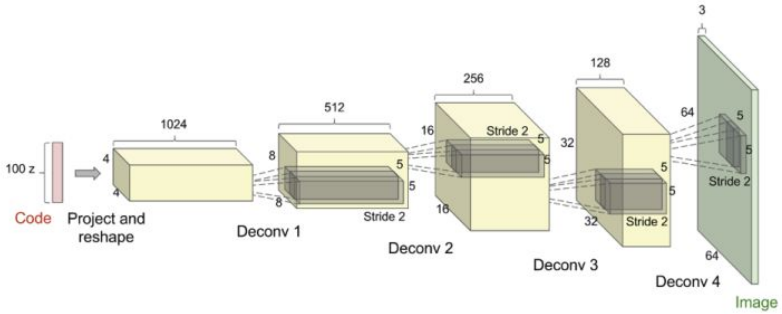
$$\min_\theta \max_\phi V(\phi, \theta) = \min_\theta V(\phi^*_\theta, \theta)$$
$$= \min_\theta \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \left[\log \frac{p(\mathbf{x})}{q(\mathbf{x};\theta)+p(\mathbf{x})}\right] + \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x};\theta)} \left[\log \frac{q(\mathbf{x};\theta)}{q(\mathbf{x};\theta)+p(\mathbf{x})}\right]$$
$$= \min_\theta \mathrm{KL}\left(p(\mathbf{x})||\frac{p(\mathbf{x})+q(\mathbf{x};\theta)}{2}\right)$$
$$+ \mathrm{KL}\left(q(\mathbf{x};\theta)||\frac{p(\mathbf{x})+q(\mathbf{x};\theta)}{2}\right) - \log 4$$
$$= \min_\theta 2\,\mathrm{JS}(p(\mathbf{x})||q(\mathbf{x};\theta)) - \log 4$$

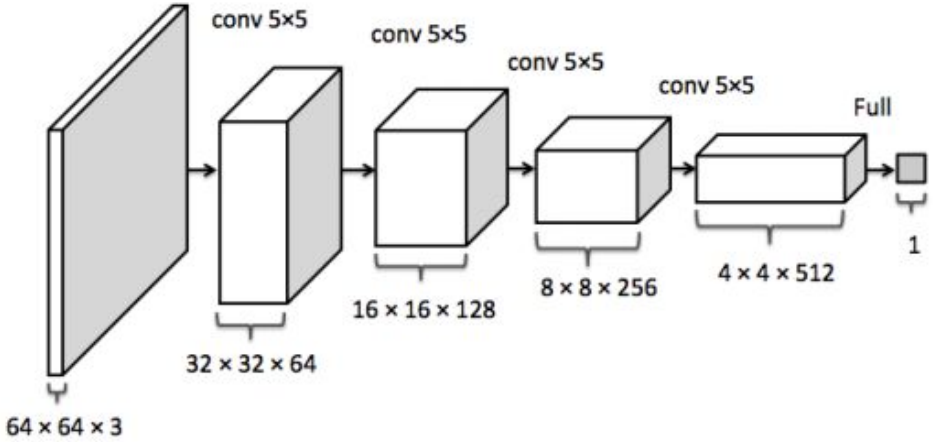
$\mathrm{JS}(p(\mathbf{x})||q(\mathbf{x};\theta))$is called the Jensen-Shannon divergence.

$$\theta^* = \min_\theta \max_\phi V(\phi, \theta)$$
$$= \min_\theta \mathrm{JS}(p(\mathbf{x})||q(\mathbf{x};\theta)).$$

The JS is minimum if and only if $p(\mathbf{x})=q(\mathbf{x}, \theta)$ for all **x** the minimax solution corresponds to a generative model that perfectly reproduces the data distribution.

Deep Convolutional GAN (DC-GAN)





Generator

Transposed CNN

Discriminator

CNN

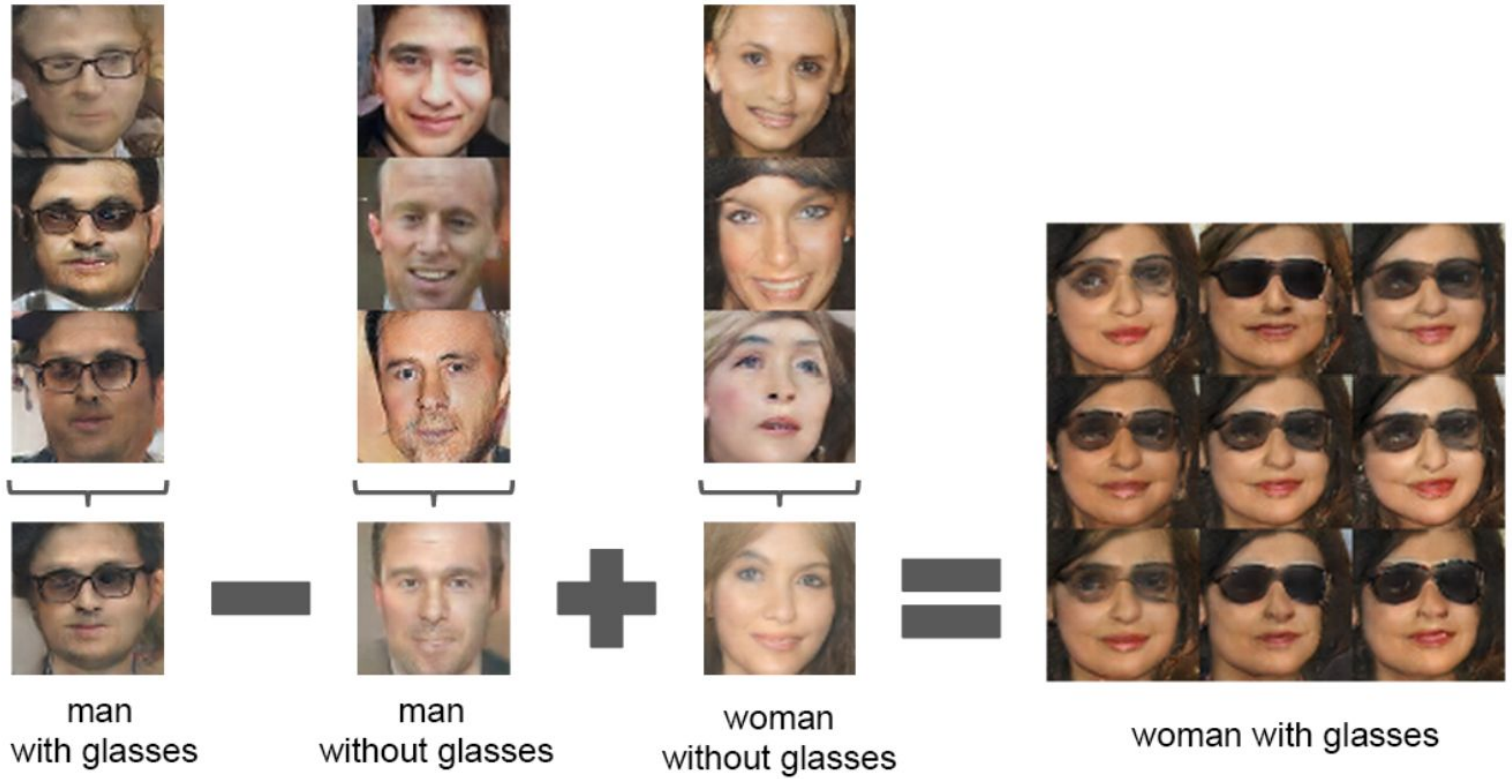
DC-GAN Tricks

- Use of batch normalizations
- Use of transposed strided convolutions
- Careful learning rates
- Several discriminator updates per generator update
- Latent space has linear properties

More tricks (very possibly outdated): https://github.com/soumith/ganhacks

(Radford et al, 2015)

DC-GAN Arithmetic



GAN Progress 2014-2021

2014: GAN

2015: DC-GAN

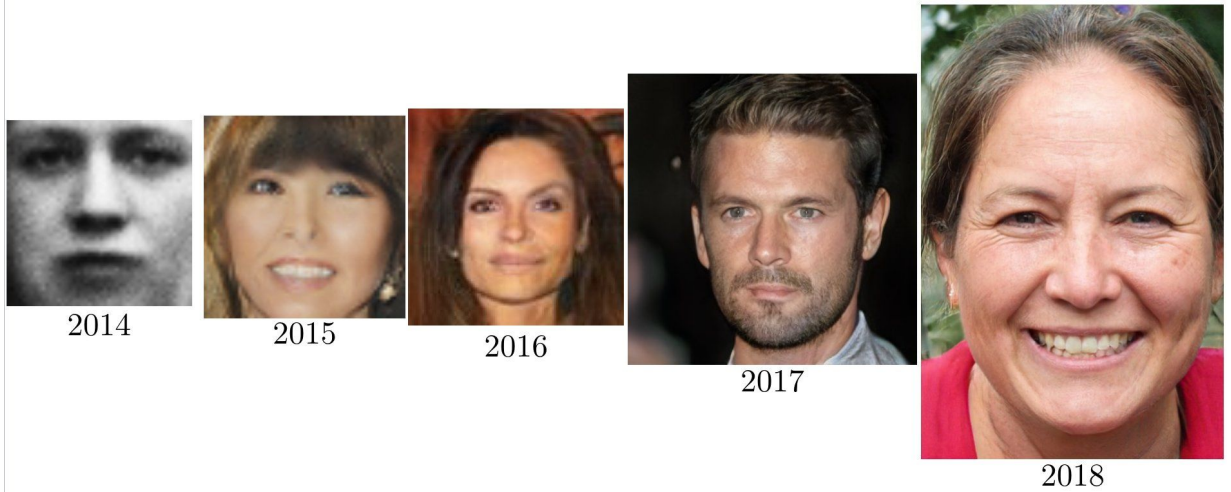
2016: Coupled GAN

2017: Progressive GAN

2018: StyleGAN

2019: StyleGAN2, BigBiGAN

2021: Alias-free GAN



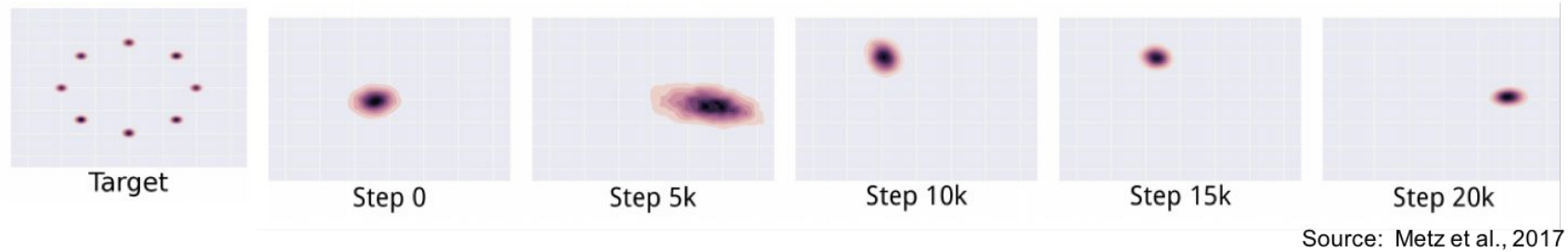

outperforming state-of-the-art diffusion models

GAN challenges

- Vanishing gradients: an over-powerful discriminator *D* leads to saturation of the value function *V*, and gradients go to zero
- Low dimensional support: the real data manifold does not have to be the generator manifold while still being indistinguishable in the data space
- Mode collapse: generator may only model a sub-sample of the data density
- Evaluation metrics are non trivial

Mode Collapse

Example: true distribution is a mixture of Gaussians



Source: Metz et al., 2017

Generator distribution keeps oscillating between different modes

- Discriminator feedback is insensitive to complete phase-space
  It will focus on point(s) of phase space the generator does not cover
- Discriminator will push generator to this mode → cycling behavior
- Would a softer distance metric address these issues?

Back to the Jensen-Shannon Divergence

Given 2 distributions ***p*** and ***q***:

$$\mathrm{JS}(p||q) = \mathrm{KL}(p||r) + \mathrm{KL}(q||r) \text{ with } r = (p+q)/2$$

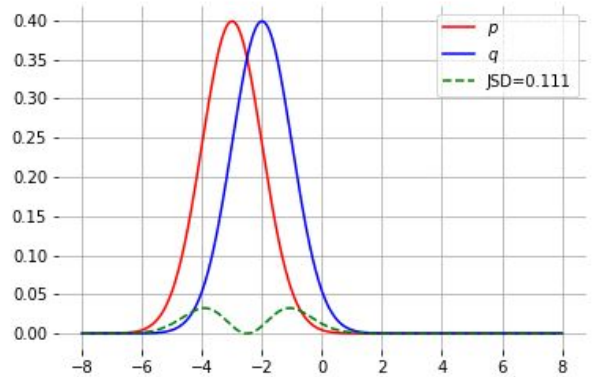
JS is symmetric and does not ignore zeros like KL

$$0 \leq \mathrm{JS}(p||q) \leq \log 2$$

when *p*=*q*

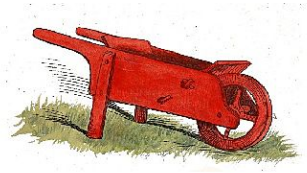
when *p* and *q* are disjoints

BUT: JS compares distributions "vertically" and not "horizontally"

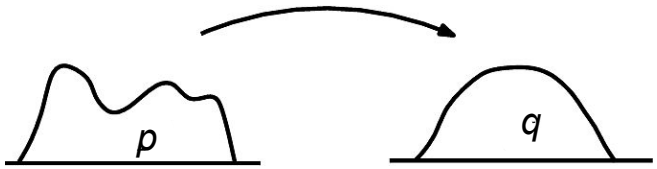


Earth Mover and Wasserstein Distance

The Earth Mover's (or Wasserstein-1) distance provides a measure of much "mass" needs to be moved to transform a distribution *p* to *q*



$$\mathrm{W}_1(p||q) = \inf_{\gamma \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\gamma}\left[||x-y||\right]$$



- $\Pi(p,q)$ is the set of all joint distributions $\gamma(x,y)$ with marginals *p* and *q*
- $\gamma(x,y)$ is the mass to be moved from *x* to *y* in order to transform *p* into *q*
- $||\cdot||$ is the L1 norm and ||*x*-*y*|| represents the cost of moving a unit of mass
- $\mathrm{W}_1(p||q) = 0$ if *p*=*q*

Wasserstein GAN: Theory

We can then try to produce a GAN with a Wasserstein-1 distance, solving:

$$\theta^* = \min_\theta \mathrm{W}_1(p(\mathbf{x})||q(\mathbf{x};\theta))$$

This does not fly very far given the definition with an inf. Instead, we can use the Kantorovich-Rubinstein duality that gives a supremum to our distance. One can show (search for proof) instead that we can rewrite our WGAN solution as:

$$\theta^* = \min_\theta \max_{\phi:||D(\cdot;\phi)||_L \leq 1} \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[D(\mathbf{x};\phi)] - \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x};\theta)}[D(\mathbf{x};\phi)]$$

real samples

generated samples

The classifier D is replaced by a "critic" D, that takes any floating point value.

Wasserstein GAN: Practice

- The new "critic" network *D* regularizes: the original WGAN clips the weight at each iteration.
- We can add a **gradient penalty** regularization term (WGAN-GP) which penalizes gradients being different from 1:

$$\mathcal{L}_{\text{GP}} = \lambda \mathbb{E}_{\mathbf{x} \sim r(\mathbf{x})} \left[\left(||\nabla_{\mathbf{x}} D(\mathbf{x})||_2 - 1 \right)^2 \right]$$

with ***r*** distribution defined as:

$$r(\mathbf{x}) : \epsilon \mathbf{x} + (1 - \epsilon) G(\mathbf{z})$$

WGAN and WGAN-GP seem to add stability and prevent mode collapse
See this blog from Lilian Weng`

Generative Models: Evaluation

Qualitative:

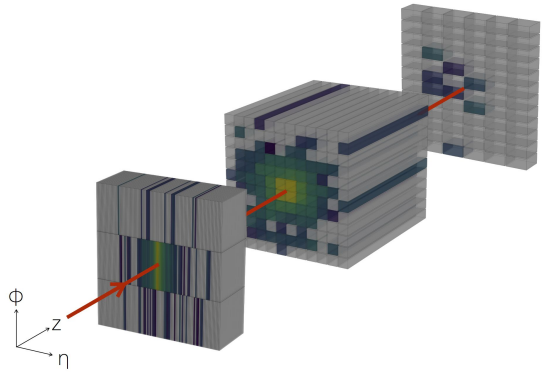
- Perceptual quality
- Diversity of generated samples
- Over-fitting

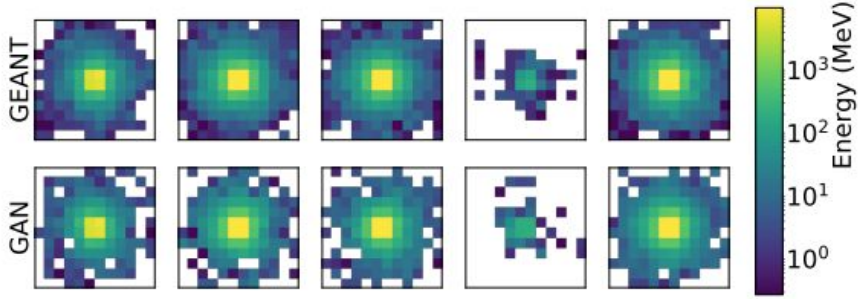
Quantitative

- log-likelihood (not for GAN)
- **Inception Score** and **Fréchet Inception Distance** (see this post)

CaloGAN

- Collisions at LHC
- CaloGAN emulates a 3 layers calorimeter at ATLAS-LHC
- Conditioned with particle energy as inputs
- Fast: generates 5M showers/min
- Custom NN design
- Paganini et al (2017)





WGAN in cosmology

N-Body simulation emulation with a Wasserstein GAN.

Rodriguez et al. (2018)

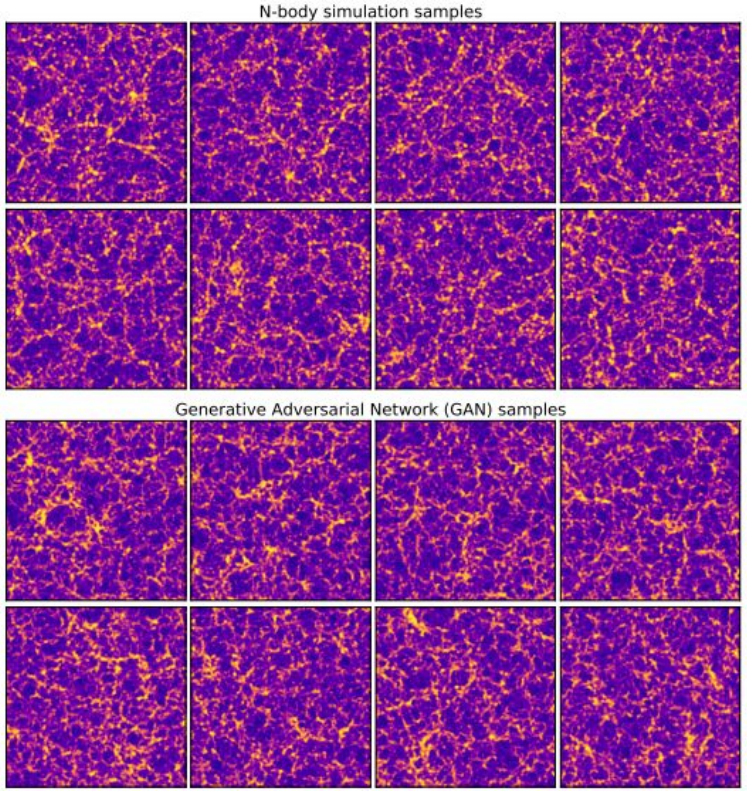
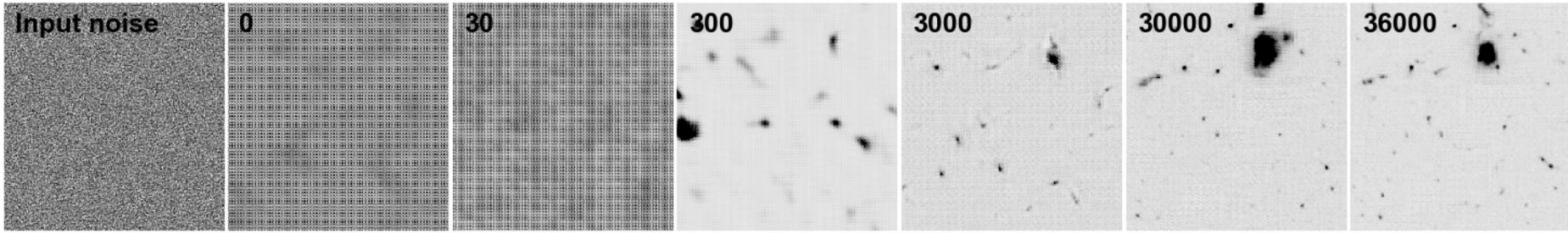


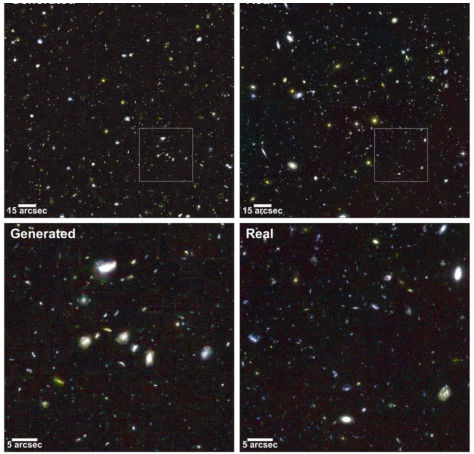
Figure 1: Samples from N-body simulation and from GAN for the box size of 500 Mpc. Note that the transformation in Equation 3.1 with $a = 20$ was applied to the images shown above for better clarity.

Generating full astronomical fields from Hubble data



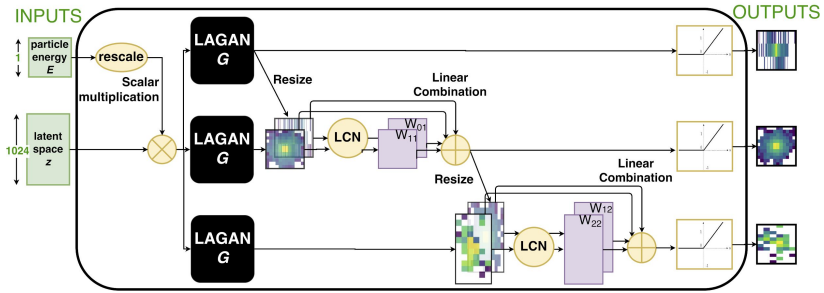
[Smith & Geach 2019]

Demo: http://star.herts.ac.uk/~jgeach/gdf

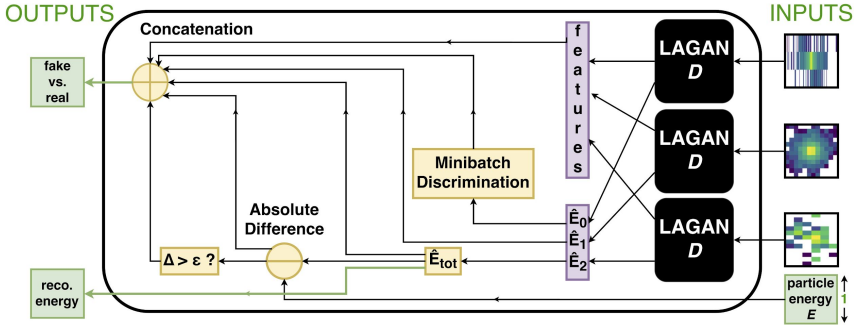


CaloGAN: a taylored GAN for ATLAS at LHC

Generator



Discriminator



Conditional Generative Models

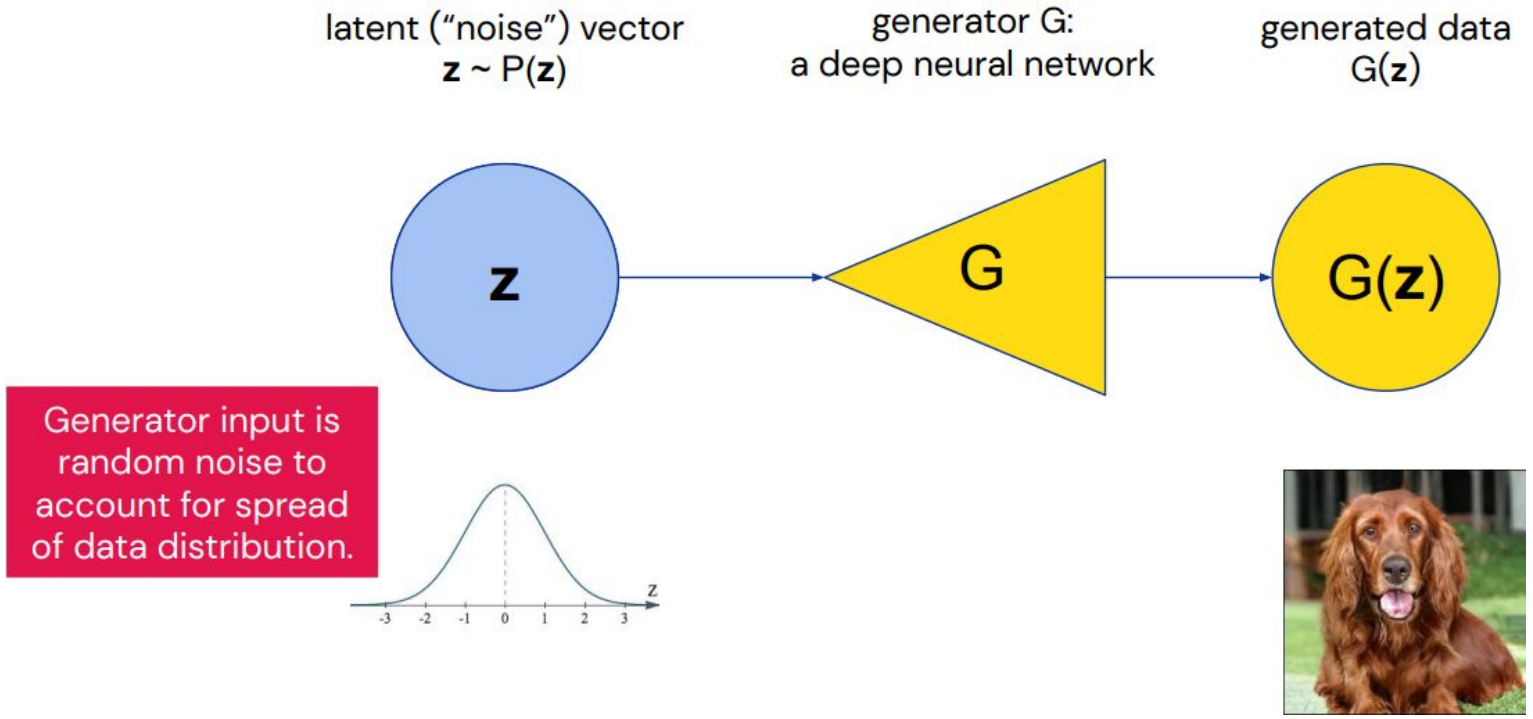
Unconditional

provides a sample from the data distribution p(**x**) without any control on what “kind” of sample

Conditional

can specify which “kind” of sample we want to generate

So far: Unsupervised GAN

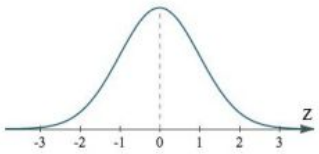


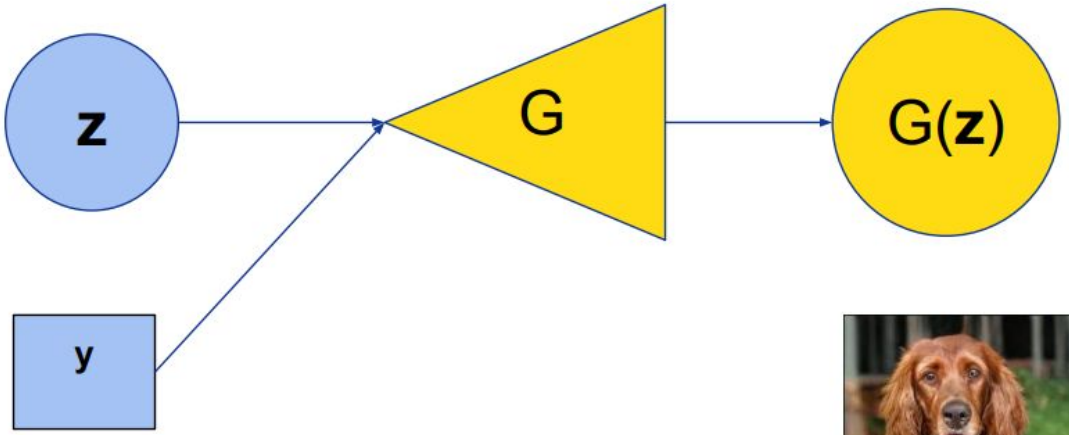
So far: Unsupervised GAN

latent ("noise") vector
z ~ P(**z**)

generator G:
a deep neural network

generated data
G(**z**)

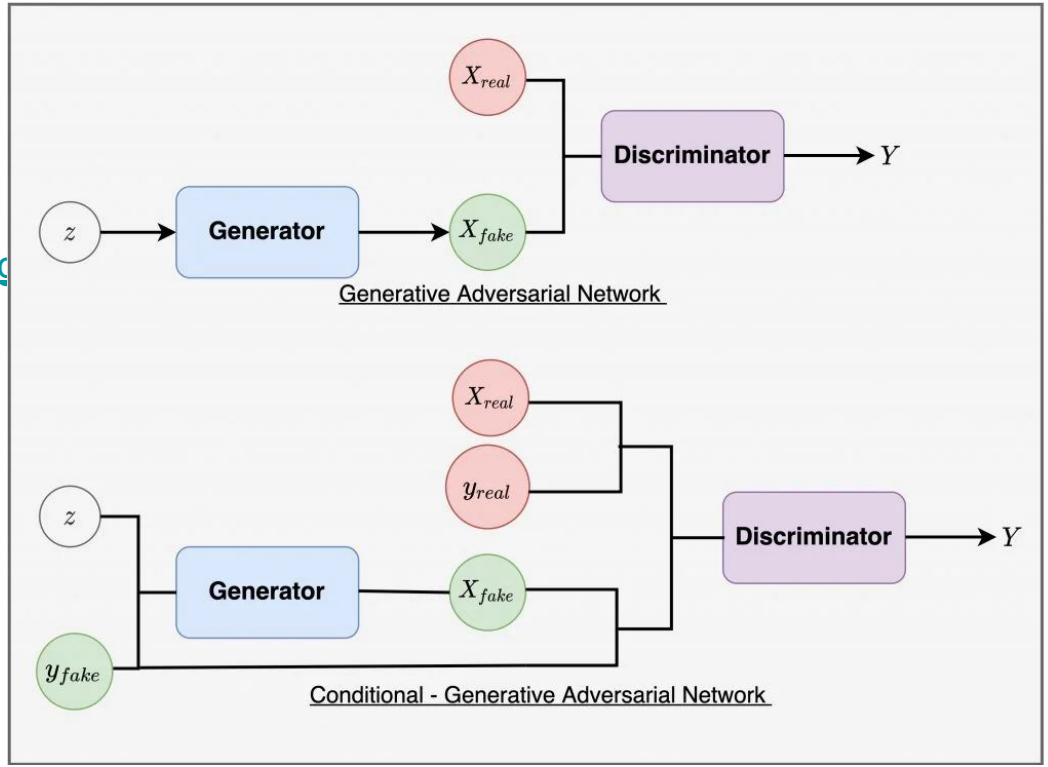




Add conditioning generation to specify information about generated sample.

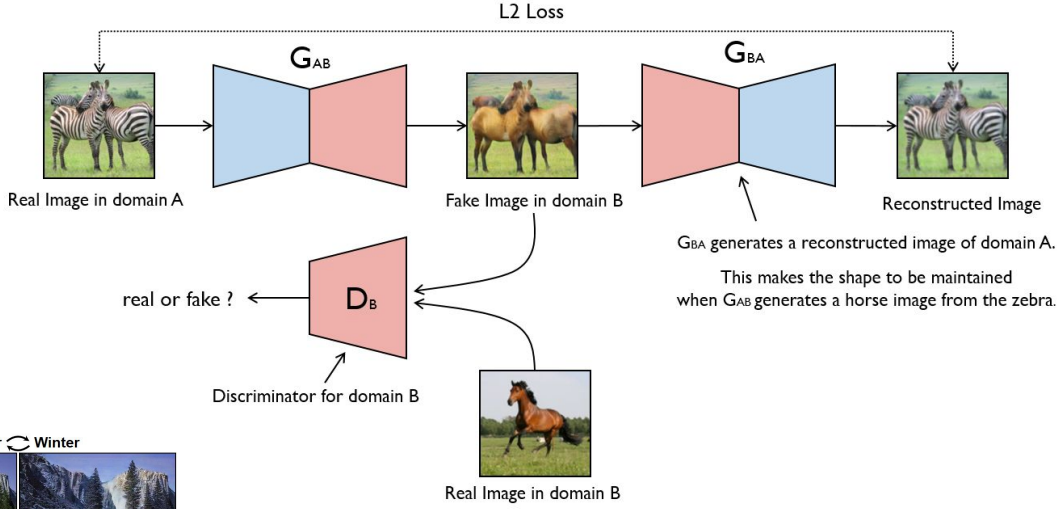
Conditional GAN

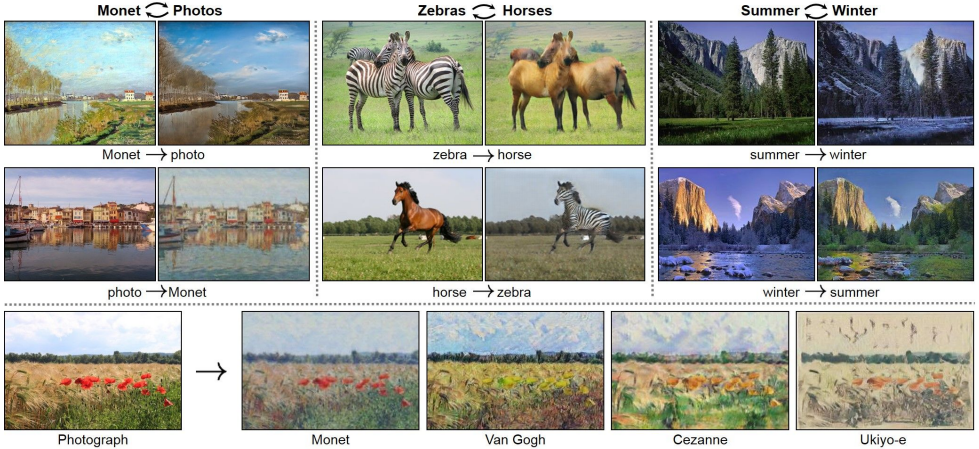
- Simplest conditional GAN on the right
- CGAN are the basis for an [Image to Imag translation] (supervised).



Cycle GAN

No alignment between paired data. Data-sets are treated separately. Zhu et al. (2017)





GAN Summary and Influence

GANs could theoretically reach generative modelling nirvana, but practically they are tricky to train. Mode collapse is not easy to solve without sacrificing the generation quality. Especially true for data which are more diverse than faces.

Adversarial training is also useful concept beyond GANs for:

- regularization with adversariality
- defending against malicious attacks (adversarial examples)
- representations blind to sensitive attributes
- domain adaptation