http://bit.ly/dlcv2018

#DLUPC

Day 4 Lecture 3

Generative Models

Kevin McGuinness
kevin.mcguinness@dcu.ie

Assistant Professor
School of Electronic Engineering
Dublin City University

DCU
Insight
Centre for Data Analytics

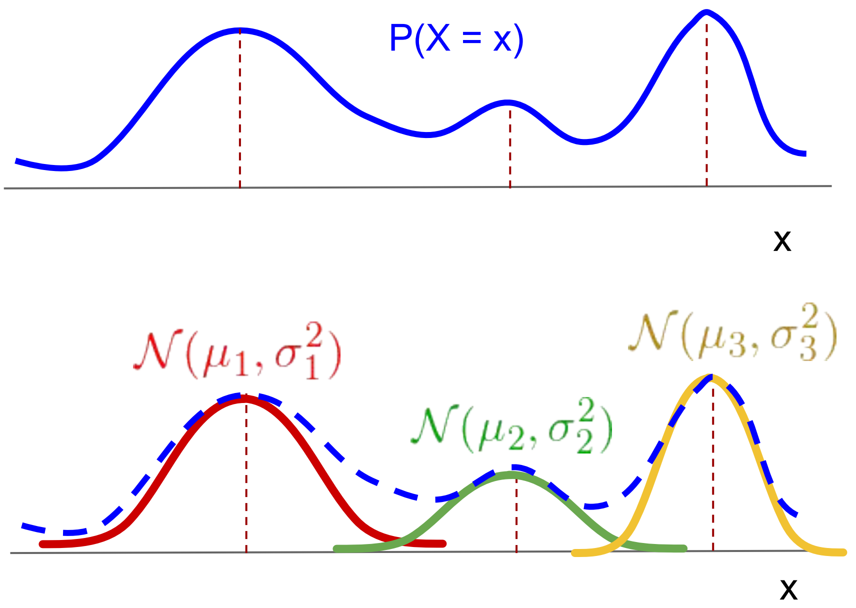
What is a generative model?

A model P(X; Θ) that we can draw samples from.

E.g. **A Gaussian Mixture Model**

- Fitting: EM algorithm
- Drawing samples:
    - Draw sample from categorical distribution to select Gaussian
    - Draw sample from Gaussian

GMMs are not generally complex enough to draw samples of images from.



$$P(X) = \lambda_1 \mathcal{N}(\mu_1, \sigma_1^2) + \lambda_2 \mathcal{N}(\mu_2, \sigma_2^2) + \dots$$

Why are generative models important?

- Model the probability density of images
- Understanding P(X) may help us understand P(Y | X)
- Generate novel content
- Generate training data for discriminative networks
- Artistic applications
- Image completion
- Monte-carlo estimators

Generative adversarial networks

Novel method of training deep generative models invented by Ian Goodfellow et al. in 2014

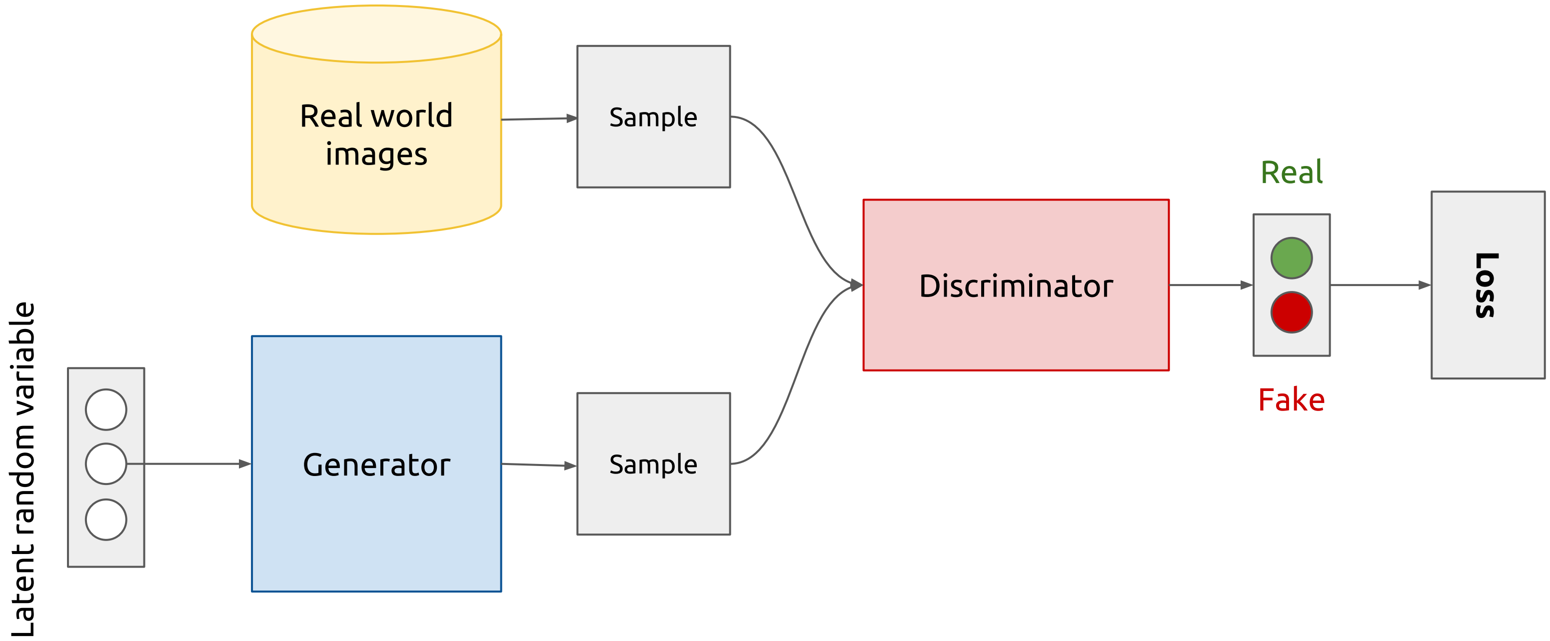
Idea: pit a generator and a discriminator against each other

- Generator tries to draw samples from P(X)
- Discriminator tries to tell if sample came from the generator or the real world

Both discriminator and generator are deep networks (differentiable functions)

Can train with backprop: train discriminator for a while, then train generator, then discriminator, ...

Generative adversarial networks (conceptual)

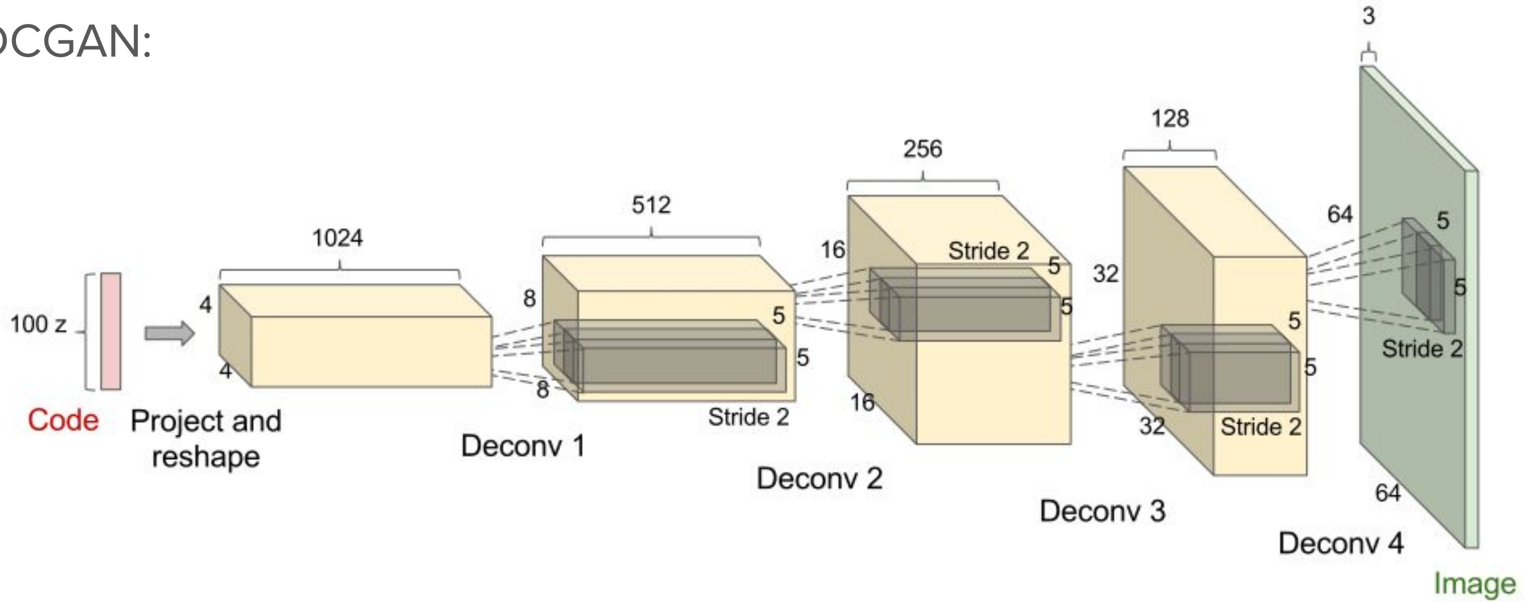


The generator

Deterministic mapping from a latent random vector to sample from q(x) ~ p(x)

Usually a deep neural network.

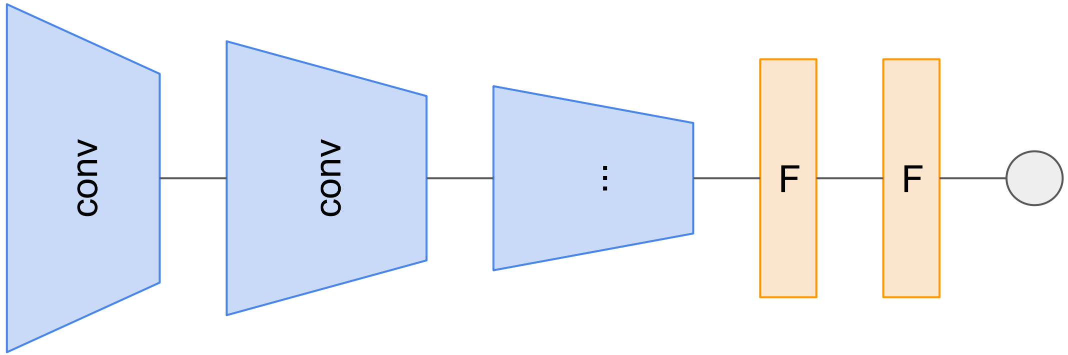
E.g. DCGAN:



The discriminator

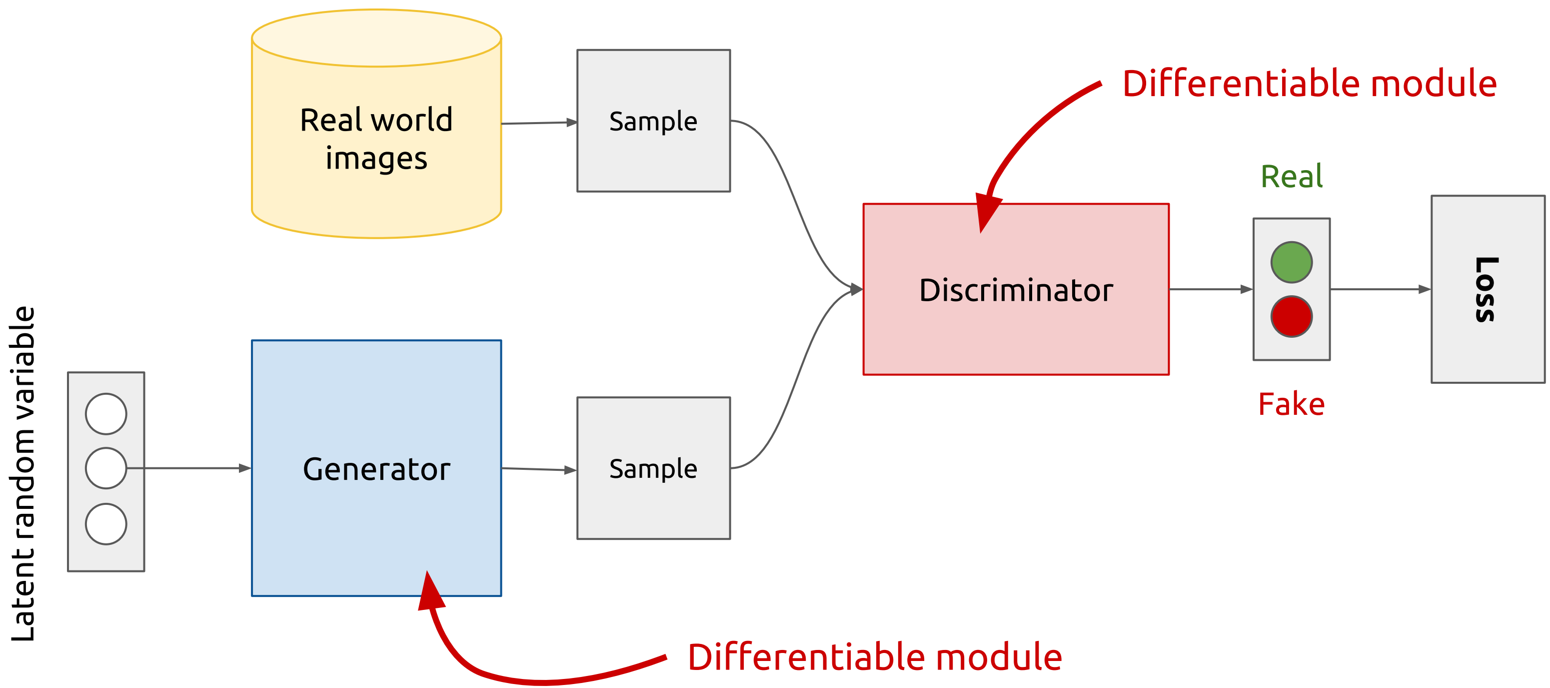
Parameterised function that tries to distinguish between samples from real images p(x) and generated ones q(x).

Usually a deep convolutional neural network.

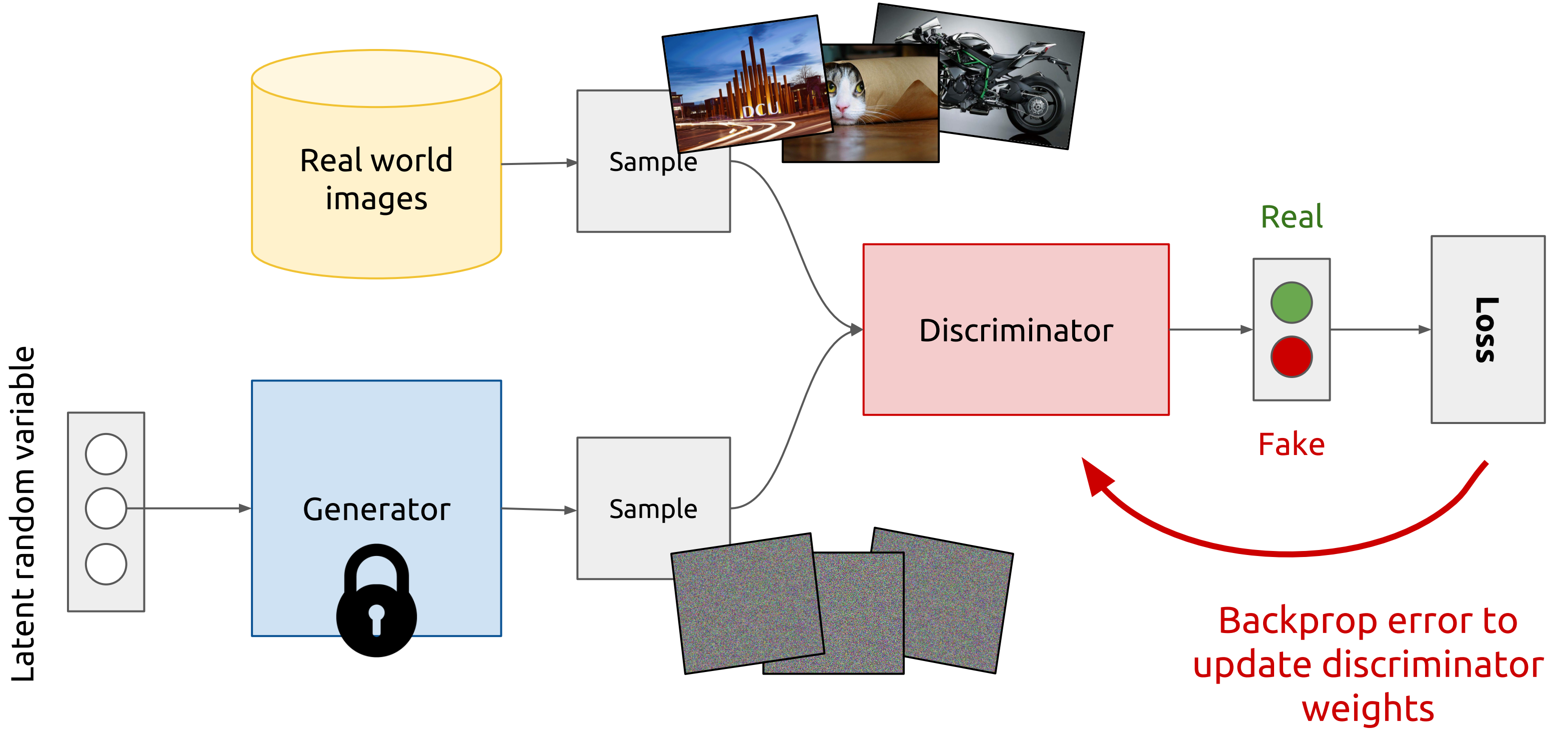


Training GANs

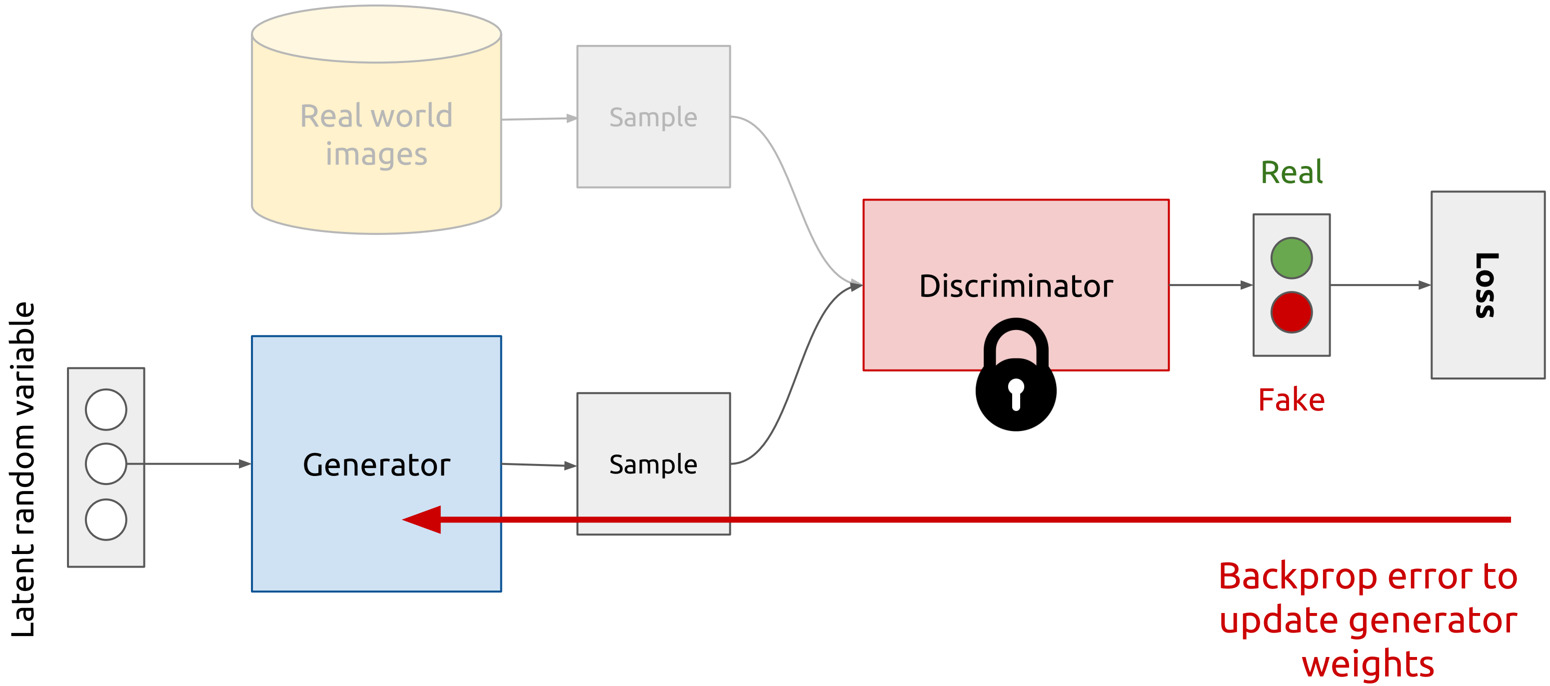
Alternate between training the discriminator and generator



1. Fix generator weights, draw samples from both real world and generated images
2. Train discriminator to distinguish between real world and generated images



1. Fix discriminator weights
2. Sample from generator
3. Backprop error through discriminator to update generator weights

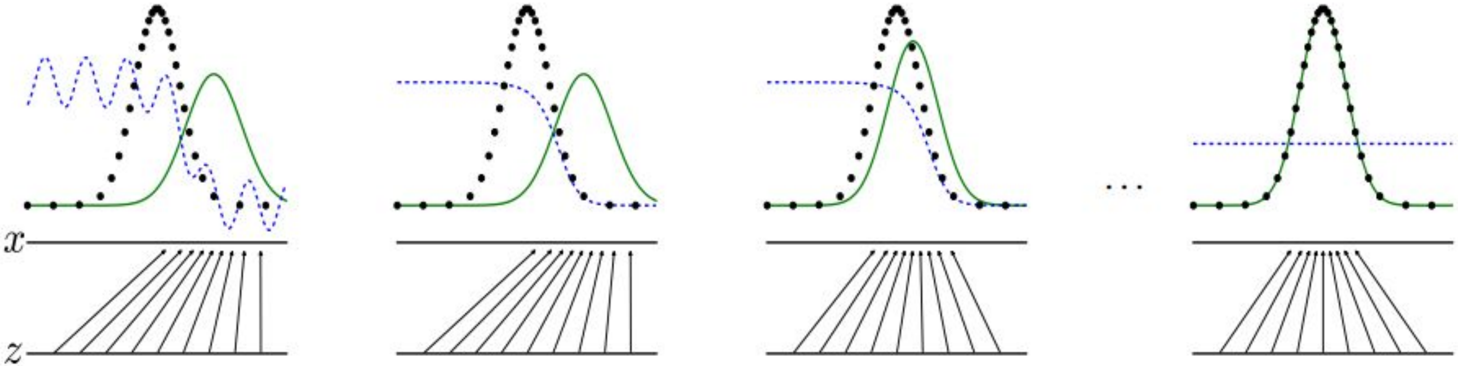


Training GANs

Iterate these two steps until convergence (which may not happen)

- Updating the discriminator should make it better at discriminating between real images and generated ones (**discriminator improves**)
- Updating the generator makes it better at fooling the current discriminator (**generator improves**)

Eventually (we hope) that the generator gets so good that it is impossible for the discriminator to tell the difference between real and generated images. Discriminator accuracy = 0.5



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**

Discriminator training

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

Generator training

- 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.

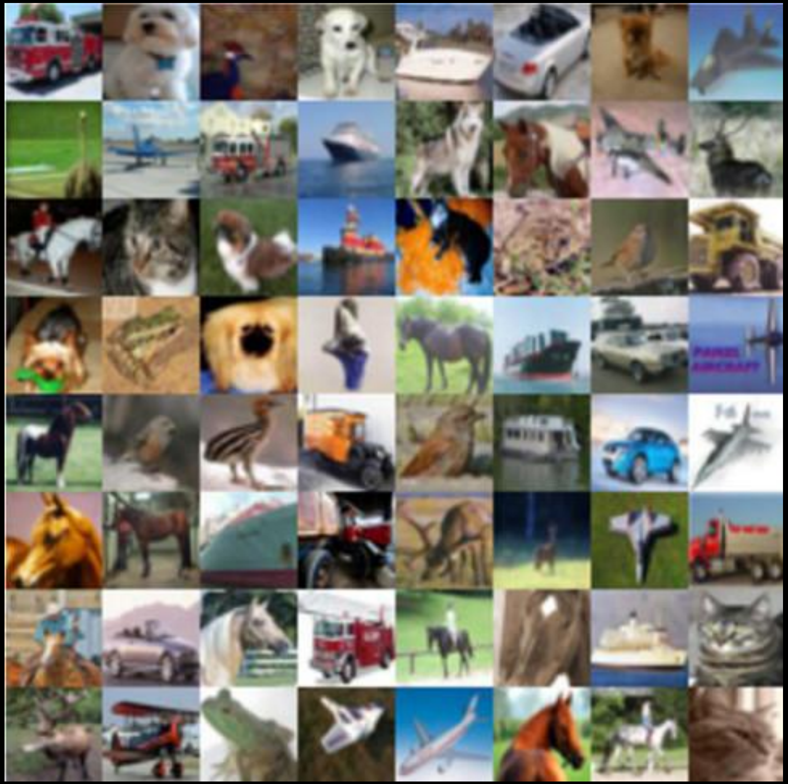
Some examples of generated images...

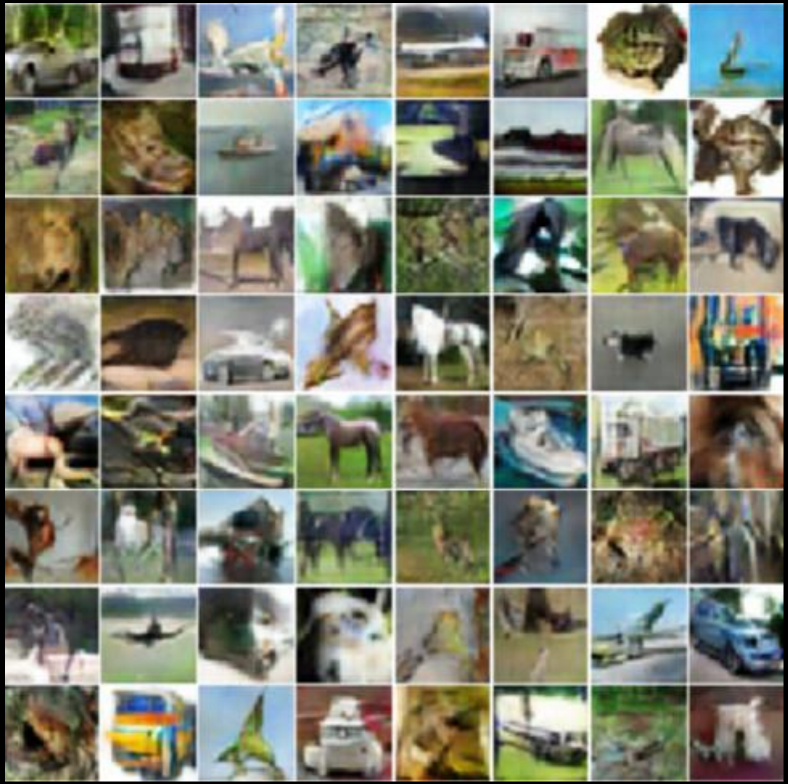
ImageNet

Source:
https://openai.com/blog/generative-models/

CIFAR-10

Source
https://openai.com/blog/generative-models/





Credit:
Alec Radford

Code on GitHub

Credit: Alec Radford   Code on GitHub

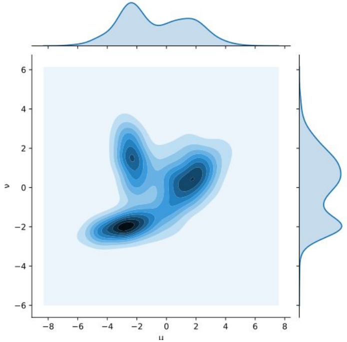
Issues

Known to be very difficult to train:

- Formulated as a “game” between two networks
- **Unstable** dynamics: hard to keep generator and discriminator in balance
- Optimization can **oscillate** between solutions
- **Mode collapse** in the generator

Difficult to evaluate results

Important variants



Wasserstein GAN (WGAN)

- MLE leads to a KL divergence loss.
- Numerical stability issues when estimated distribution and true distribution do not overlap significantly (loss blows up).
- WGAN idea is to use a coarse approximation of the Wasserstein distance (the Earth mover's distance).
- Weight clipping is needed to enforce Lipschitz constraint.

Overall effect is to make the GAN more stable. Discriminator can be trained more on each step without blowing up.

Can work well in practice, but clipping the weights to enforce Lipschitz slows training.

Important variants

Least squares GAN (LSGAN)

- Similar motivation to WGAN: want a loss that gives nice gradients and doesn't blow up.
- LSGAN Idea: just use squared error (L^2 distance)!
- Turns out this is the same as minimizing the Pearson χ^2 divergence.

(a) Church outdoor.

(b) Dining room.

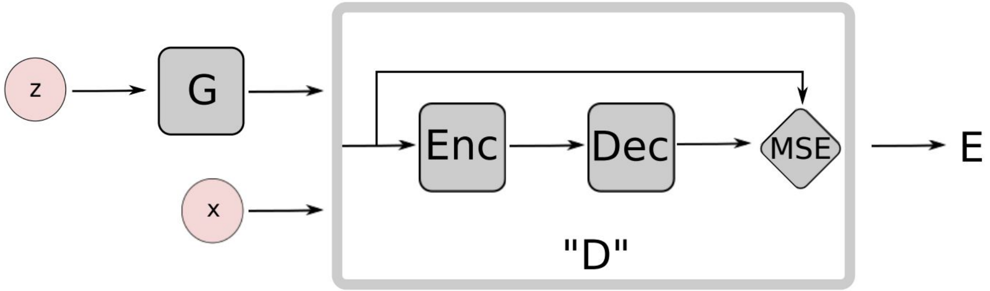
(c) Kitchen.

(d) Conference room.

Important variants

Energy-based GAN (EBGAN)

- Instead of using a binary classifier as the discriminator *D* use an energy-based model (an autoencoder)
- *D* models the image manifold since it is trained on real images
- Optimize to generate samples that have low energy
- Generator gets more *signal* from *D*



Important variants

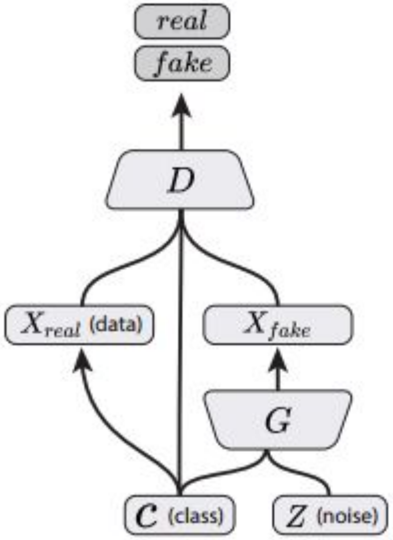
Boundary Equilibrium GAN (BEGAN)

- Combines ideas from WGAN and EBGAN
- BEGAN idea: matching the distributions of the reconstruction losses can be a suitable proxy for matching the data distributions.
- Use Wasserstein distance approximation to do this
- Includes mechanism for automatically maintaining equilibrium

Conditional GANs

GANs can be conditioned on other info: e.g. a label

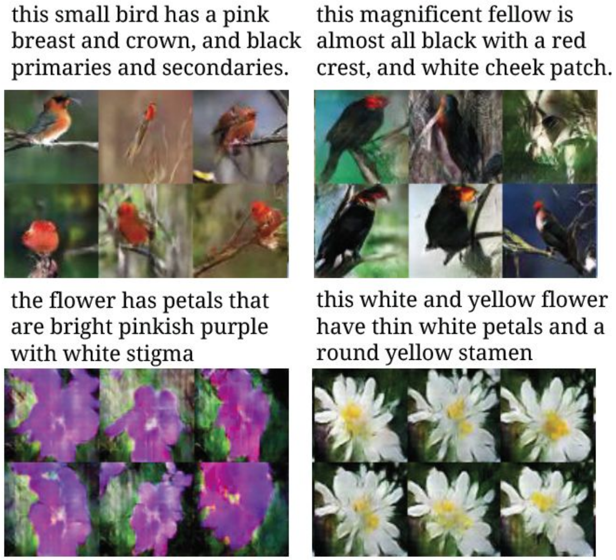
- z might capture random characteristics of the data, variabilities of possible futures,
- c would condition the deterministic parts (label)



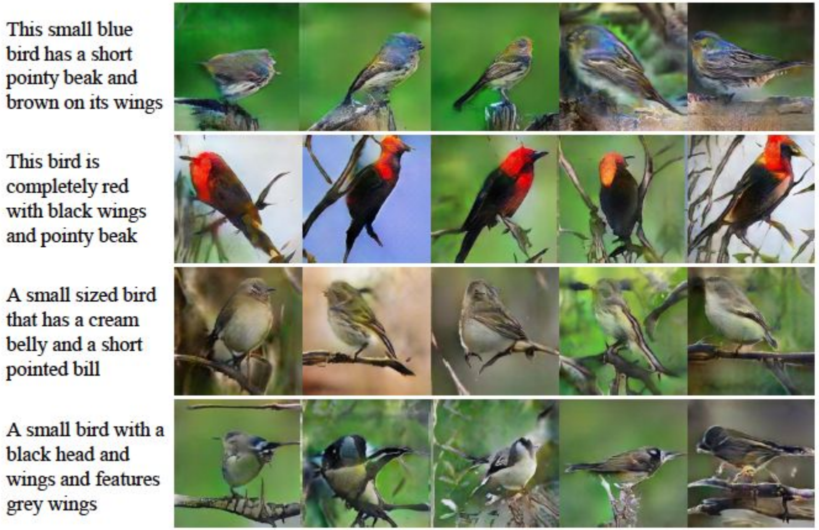
Conditional GAN
(Mirza & Osindero, 2014)

For details on ways to condition GANs: [Ways of Conditioning Generative Adversarial Networks (Wack et al.)]

Generating images/frames conditioned on captions



(Reed et al. 2016b)



(Zhang et al. 2016)

Predicting the future with adversarial training

Want to train a model to predict the pixels in frame (t+K) from pixels in frame t.

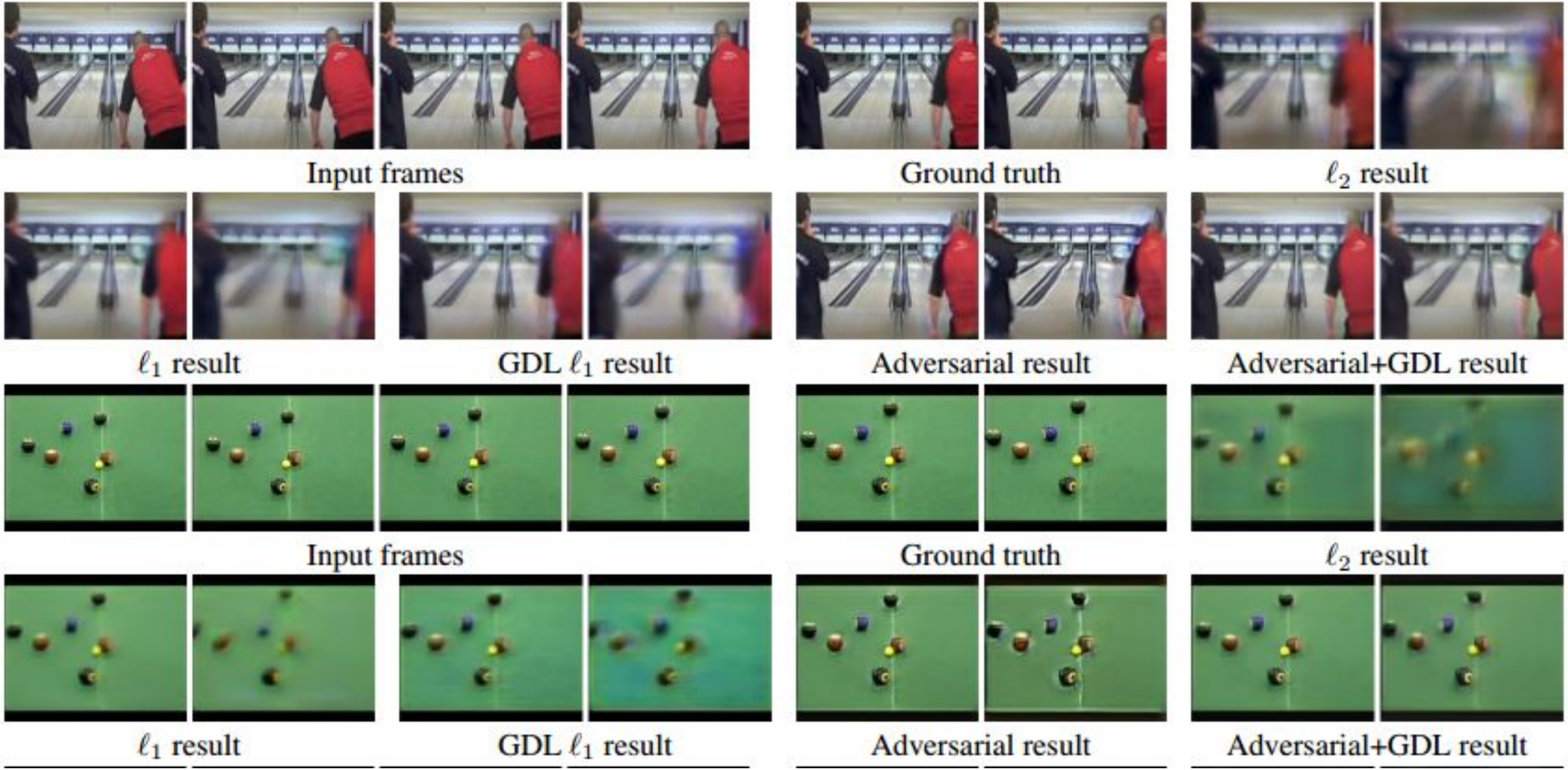
Many possible futures for same frame

Using supervised loss like MSE results in blurry solutions: loss if minimized if predictor averages over possibilities when predicting.

We really want a sample, not the mean

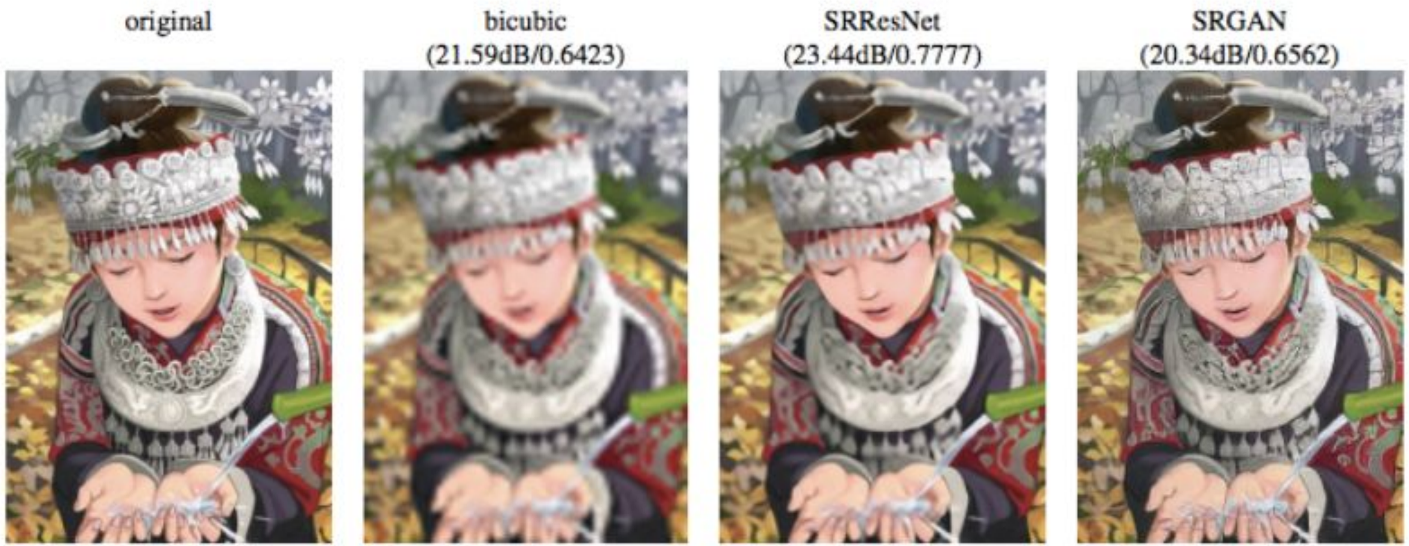
Adversarial training can solve this: easy for an adversary to detect blurry frames

Mathieu et al. **Deep multi-scale video prediction beyond mean square error**, ICLR 2016 (https://arxiv.org/abs/1511.05440)



Mathieu et al. **Deep multi-scale video prediction beyond mean square error**, ICLR 2016 (https://arxiv.org/abs/1511.05440)

Image super-resolution



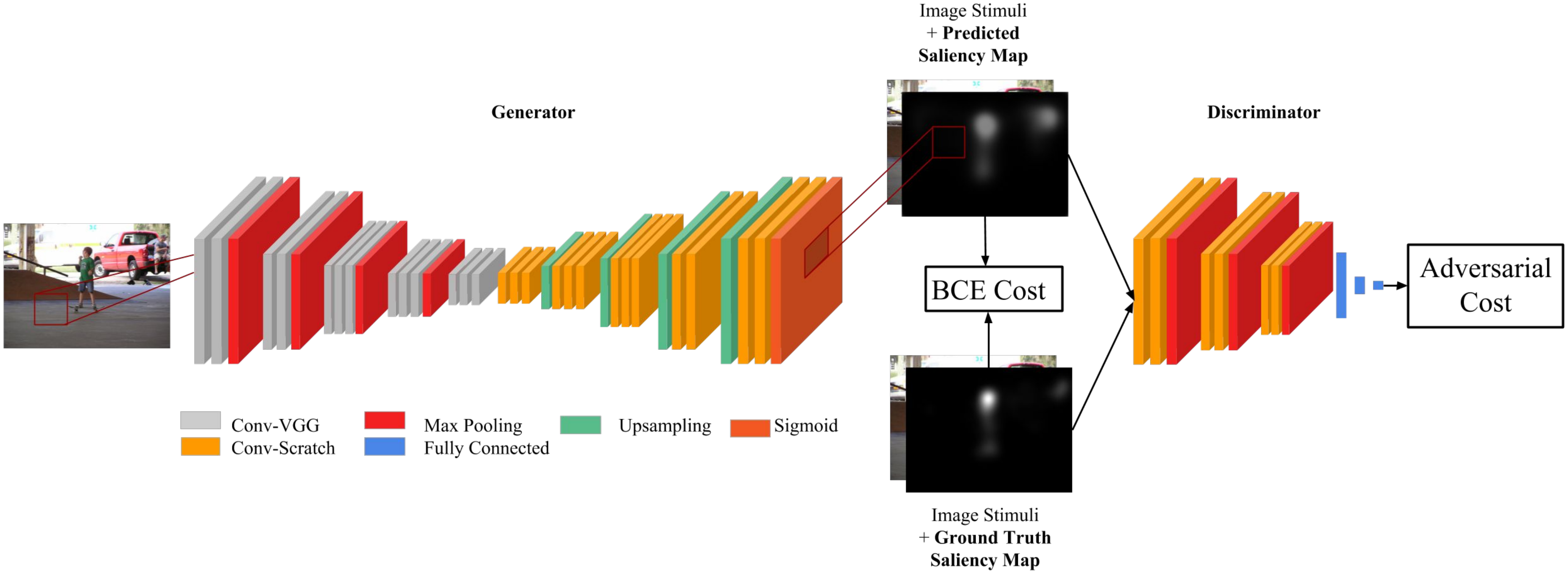
(Ledig et al. 2016)

Bicubic: not using data statistics. SRResNet: trained with MSE. SRGAN is able to understand that there are multiple correct answers, rather than averaging.

Saliency prediction

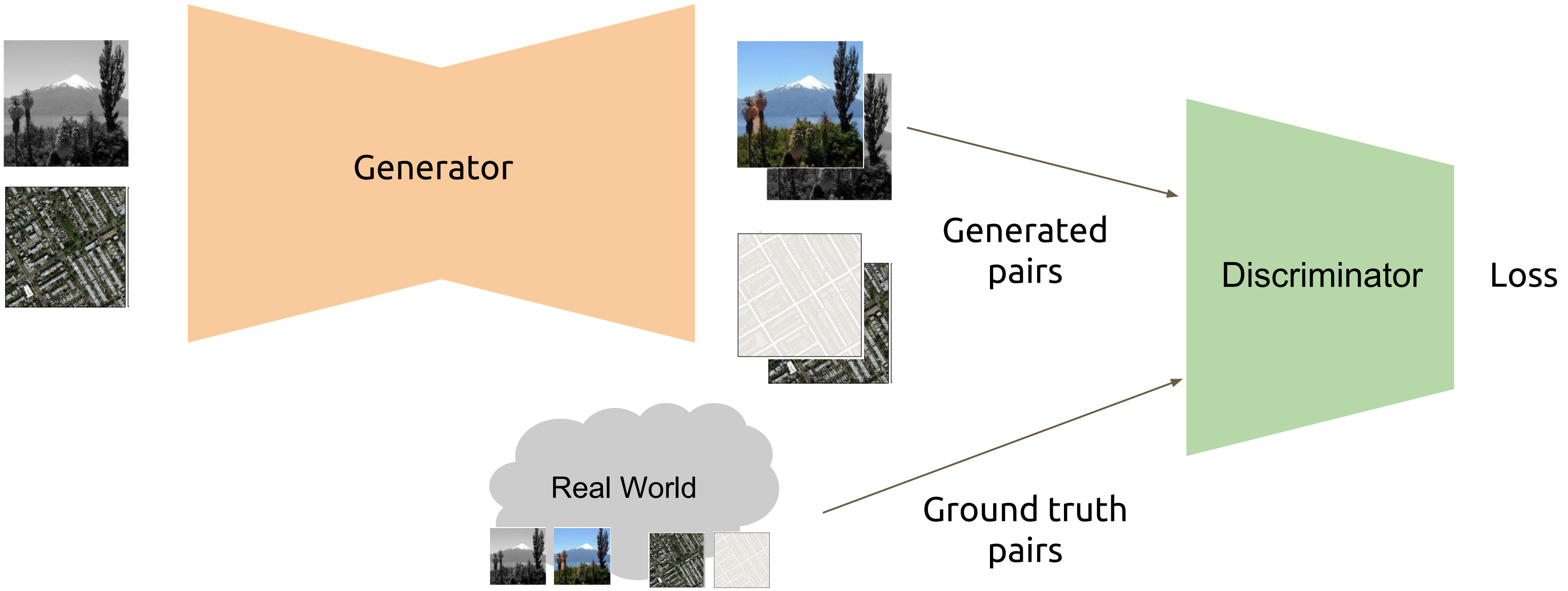
Dala loss Adversarial loss

$$\alpha \cdot \mathcal{L}_{BCE} - \boxed{\log D(I, \hat{S}),}$$



Junting Pan, Cristian Canton, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol and Xavier Giro-i-Nieto. "SalGAN: Visual Saliency Prediction with Generative Adversarial Networks." arXiv. 2017.

Image-to-Image translation



Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. "Image-to-image translation with conditional adversarial networks." arXiv:1611.07004 (2016).

Questions?