Deep Learning
Summer semester '24

Autoencoders & Generative Adversarial Networks

).

So far........

- Insights into Deep Learning
- Optimization and Training
- Convolutional Neural networks
  I. Convolutions
  II. Padding
  III. Stride
  IV. Transposed Convolutions
  V. Pooling Upsampling

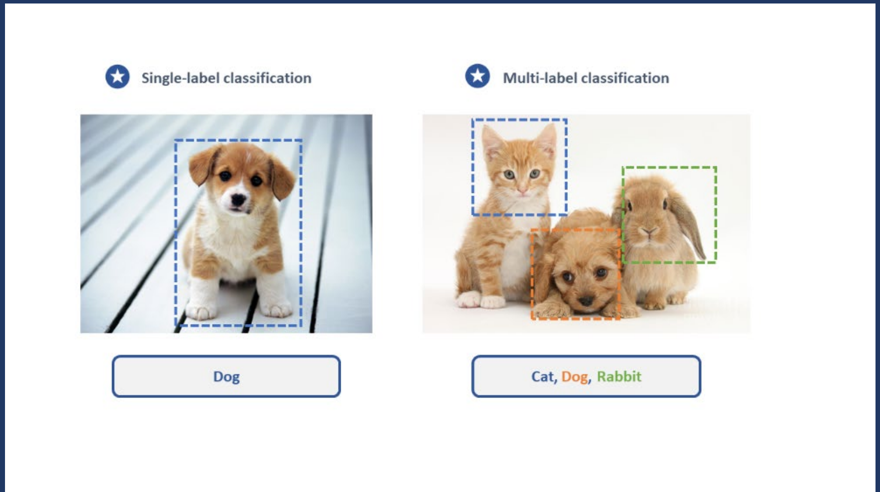
Content

- Insights into Generative Models
- Autoencoders
  - Variational Autoencoders
- Generative Adversarial Networks

Supervised vs Unsupervised Learning

Supervised Learning

- **Data**: (x,y)
- x is the data, y is the label
- **Goal**: Learn the mapping function X →Y
- **Examples**: Classification, Object Detection

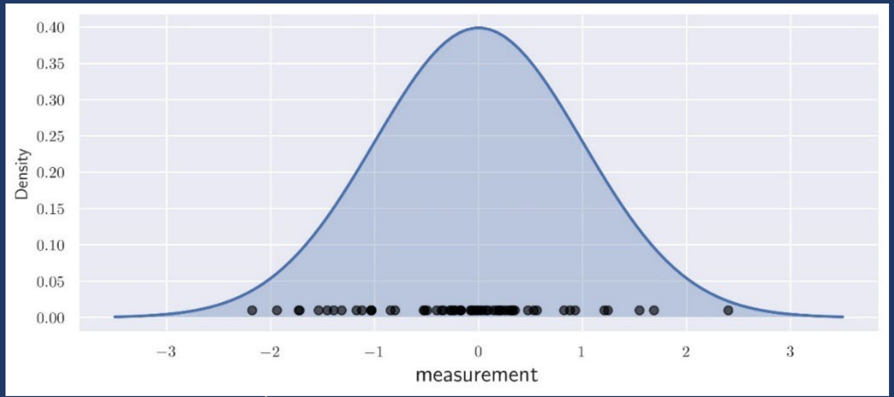


Pic source: https://ambolt.io/en/image-classification-and-object-detection/

Unsupervised Learning

Core problem in unsupervised learning

- **Data**: x
- Just data, without any labels!
- **Goal**: To learn some underlying structure of data
- **Examples**: Clustering (K-Means), dimensionality reduction (PCA), density estimation (understand distribution of data),etc.



Pic source: https://towardsdatascience.com/kernel-density-estimation-explained-step-by-step-7cc5b5bc4517

Generative vs. Discriminative Models

Our Target

Generative

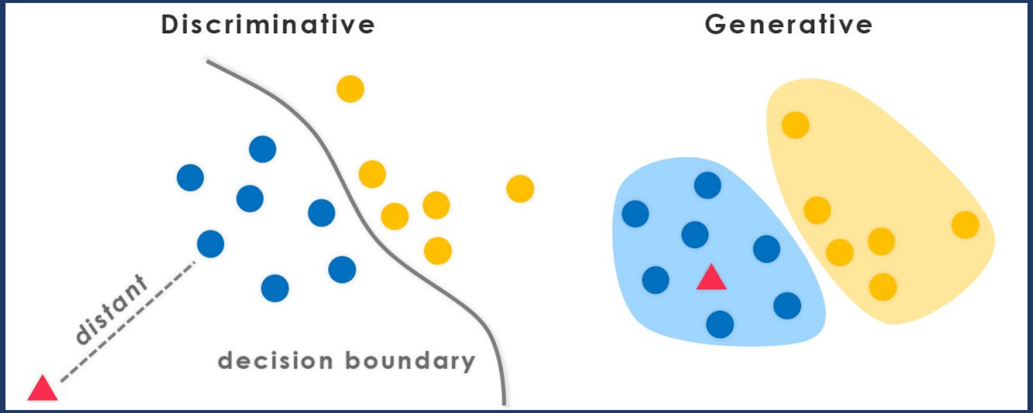
- Generate data from the given data samples
- Learn a model of the joint probability P(y, x)
- Use Bayes' Rule to calculate P(x|y)
- Build a model of each class; given example, return the model most likely to have generated that example by learning the probability distribution of data p(x)
- Examples: Naïve Bayes, Gaussian Discriminant Analysis, Autoencoders, Diffusion Models.

Recall **Bayes' Rule:**

$$\underbrace{\boxed{P(x\mid y)}}_{\text{Conditional Generative Model}} = \frac{\overbrace{\boxed{P(y\mid x)}}^{\text{Discriminative Model}}}{\underbrace{\boxed{P(y)}}_{\text{Prior over labels}}}\overbrace{\boxed{P(x)}}^{\text{(Unconditional) Generative Model}}$$

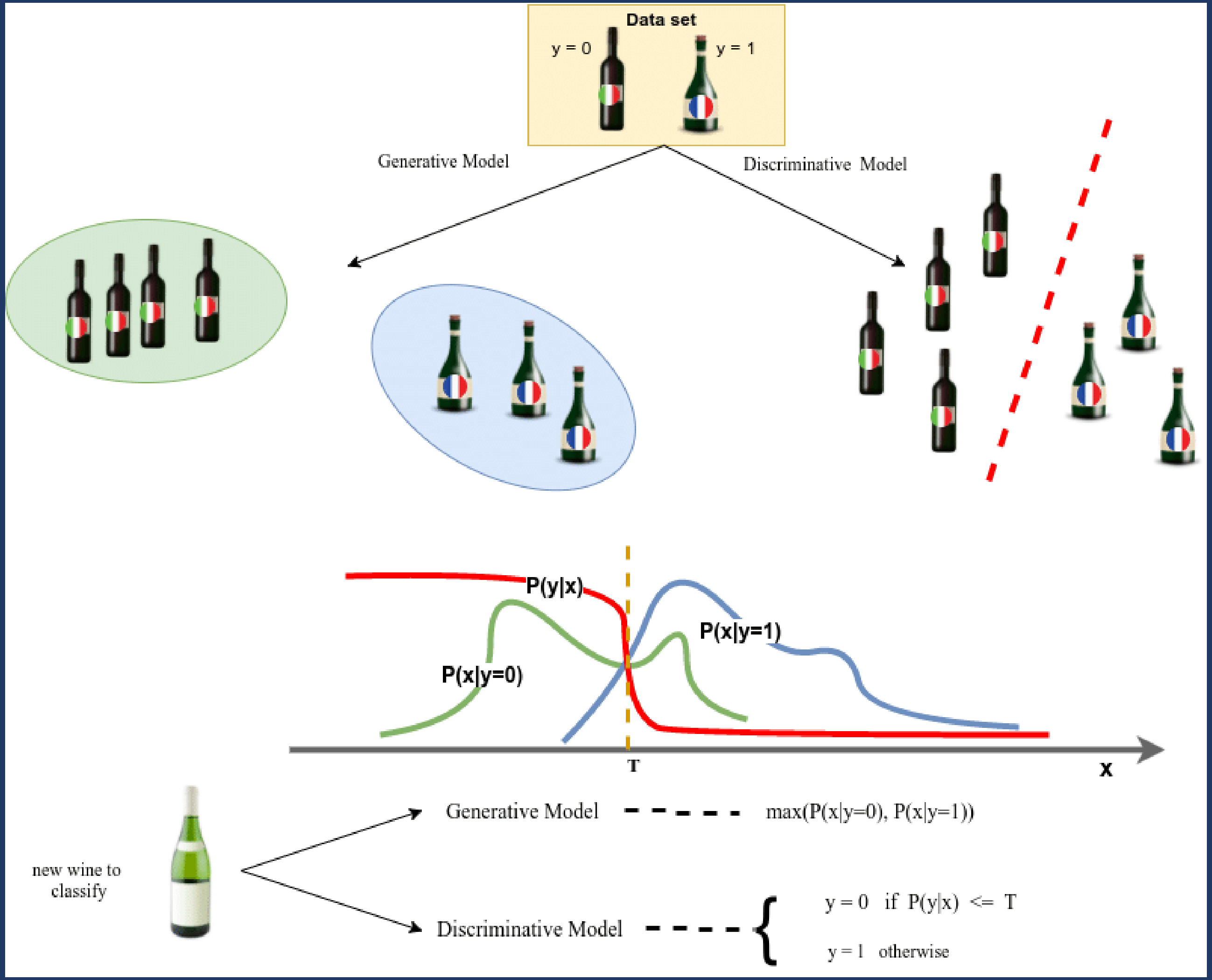
Discriminative

- Classify data by finding the decision boundary
- Model posterior probability P(y|x) directly
- Find the exact function that minimizes classification errors on the training data without learning the probability distribution of data p(x)
- Examples: Logistic regression, Support Vector Machines (SVMs), Decision Trees



Pic source: https://www.baeldung.com/cs/ml-generative-vs-discriminative

Generative vs. Discriminative Models

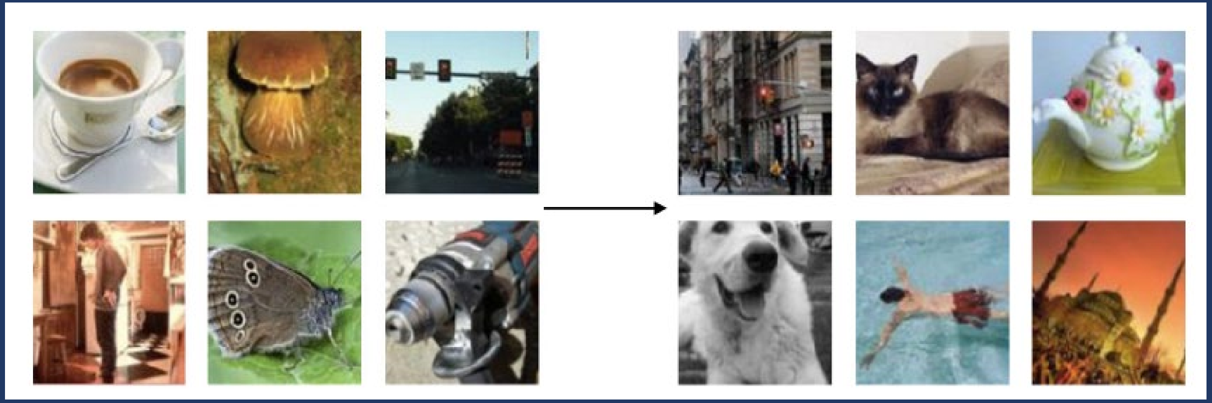


- Given an image X, the discriminative models predict the label Y and can't model P(X).
- They can't sample from P(X) and can't generate new images.
- Generative models can model P(X) and generate new images.

Pic source: https://www.baeldung.com/cs/ml-generative-vs-discriminative

Generative Models

Given training data, generate new samples from same distribution



Training data~$p_{data}(x)$ Generated samples~$p_{model}(x)$

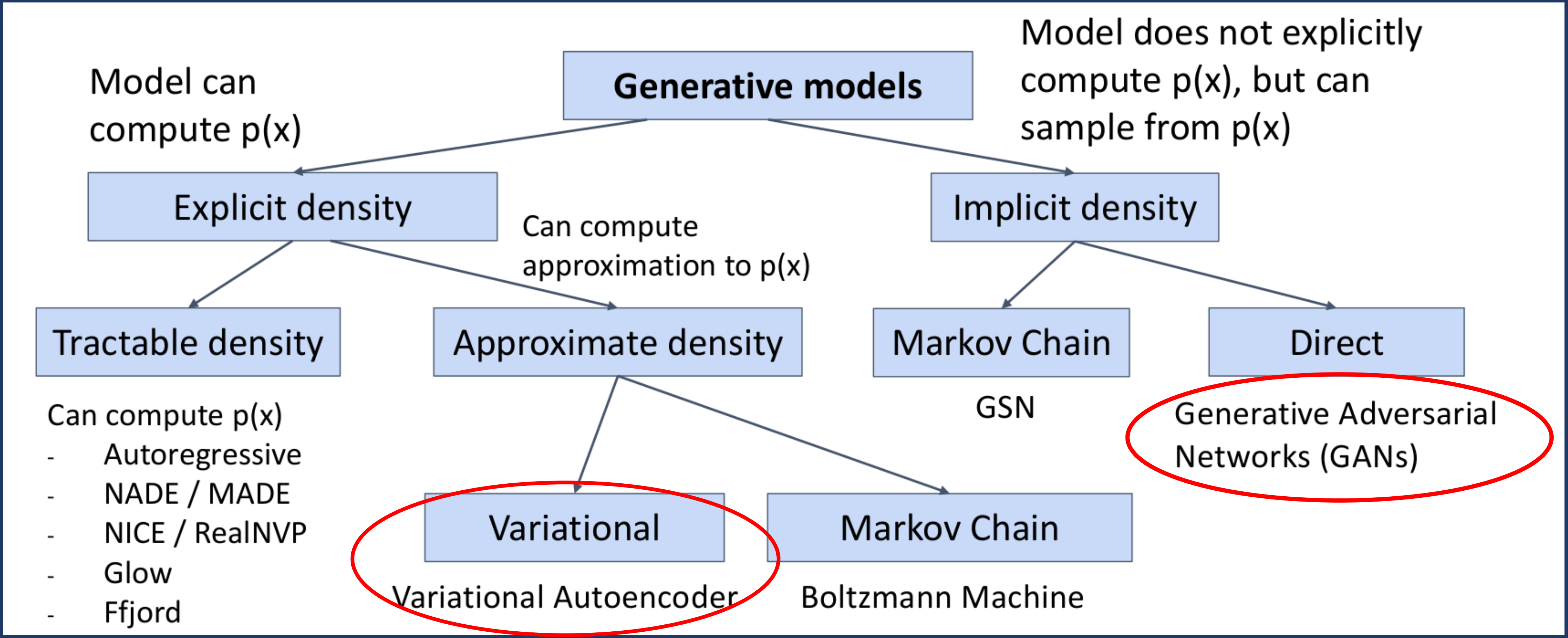
Want to: learn $p_{model}(x)$ similar to $p_{data}(x)$

- Addreses density estimation that is a core problem in unsupervised learning

- Implicit Density Estimation: learn model that can sample from $p_{model}(x)$ without explicitly defining it.
- Explicit Density Estimation: explicitly define and solve for $p_{model}(x)$

source: Slide Adapted from Stanford Lecture on Generative Models

Taxonomy of generative models

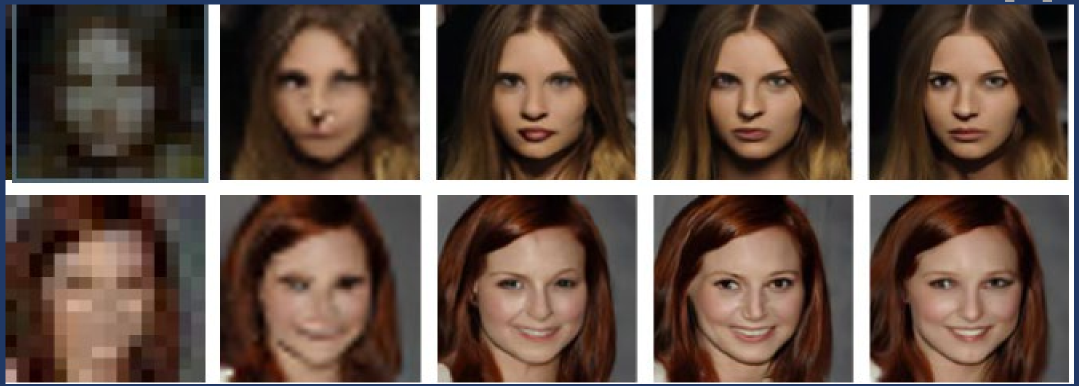


Pic source: Ian Goodfellow, Tutorial on Generative Adversarial /networks, 2017

Magic of Generative models

Image and Video Generation: They generate realistic sampled for content creation, virtual reality, artwork, super-resolution, colorization.

[1]



[2]

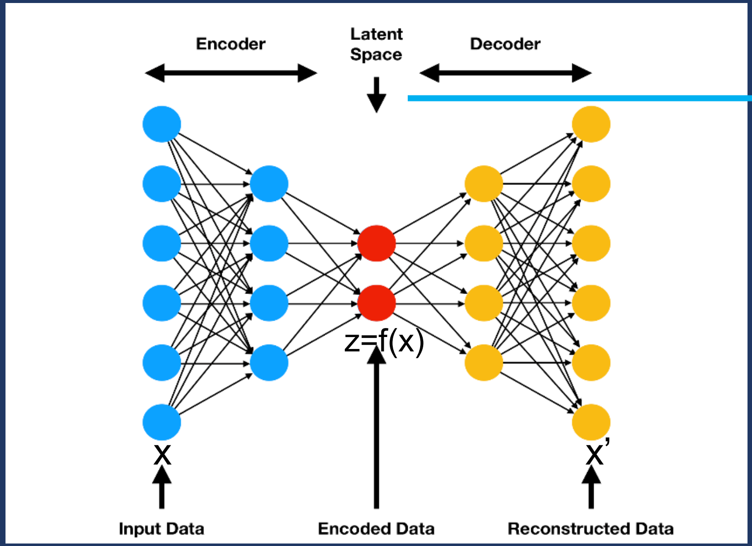
Text to Image Generation: They can generate human like text, making them useful for chatbots, language translation, and content generation.
Model physical world for simulation and planning (robotics and reinforcement learning applications)
Music Generation: They can compose original music, allowing for creation of new melodies and harmonies.
Many more….

[1] Gao, Sicheng, et al. "Implicit diffusion models for continuous super-resolution"CVPR. 2023.
[2] Zhu Junchen, et al. "Moviefactory: Automatic movie creation from text using large generative models for language and images." *Proceedings of the 31st ACM*. 2023

Content

- Insights into Generative Models
- Autoencoders
  - Variational Autoencoders
- Generative Adversarial Networks

Autoencoders



Latent space has dimension smaller than x to capture the important features.

Still we are unable to generate new images as we don't know about the space of z .

How to make AEs a generative model?

- An autoencoder is a feed-forward neural net whose job it is to take an input and reconstruct **x'**.
- **Encoder**: z = f(x)
- **Decoder:** x'=g(y)
- Basically, what is happening here, we train for x'=x.
- AEs tries to learn an approximation of the identity by "Autoencoding"- encoding itself.

Pic Source: https://www.compthree.com/blog/autoencoder/

Content

- Insights into Generative Models
- Autoencoders
  - Variational Autoencoders
- Generative Adversarial Networks

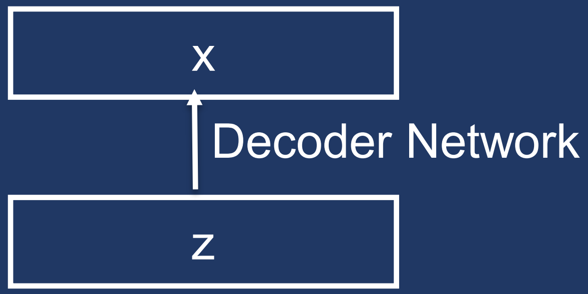
Variational Autoencoders

- Traditional AEs compute a deterministic feature vector describing the attributes of input in the latent space.
- However, Variational Autoencoders an unsupervised approach uses a variational outlook to learn the latent representation.
- **Probalisitic spin** on data will let us sample from the model to generate data.
- Thus allowing to model uncertainty in the input data

Variational Autoencoders: Statistical Motivation

- **Assumption:** Latent/hidden variable (z) that generates an observation x.
- Training a variational autoencoder: determining the distribution of z.

Sample from true conditional $p(x|z)$ (More complex)



Sample from true prior $z \sim p(z)$ (Simple Gaussian)

To learn the model parameters we need to maximize the likelihood of training data or the intractable density function,

$$p(x) = \int p(z)p(x|z)$$

$$p(z|x) = \int p(x|z)p(z)/p(x)$$

- Computing arbitrary p(x|z) for every z is usually intractable.
- Solution: In addition to the decoder network that is modelling p(x|z), also define an additional encoder tractable distribution q(z|x) that approximates the true distribution p(z|x).

*Autoencoding Variational Bayes (Kingma and Welling, 2013)

Variational Autoencoders: Statistical Motivation

Tractable lower bound that we can compute the gradient of

$$\log p(x) = E_{z \sim q(z|x)} \, [\log p(x)]$$

Using Baye's rule and multiplying with constants we get,

$$= \boxed{E_z[\log p(x|z)] - D_{KL}(q(z|x)||p(z))} + D_{KL}(Q(z|x)||p(z|x))$$

This data likelihood needs to be maximized

Decoder network gives pθ(x|z), can compute estimate of this term through Sampling and reconstruct the input data

This KL term (between Gaussians for encoder and z prior) has nice closed-form Solution and thus the encoder make posterior distribution close to prior.

p(z|x) intractable can't compute this KL term :(But the KL divergence always >= 0.

*Autoencoding Variational Bayes (Kingma and Welling, 2013)

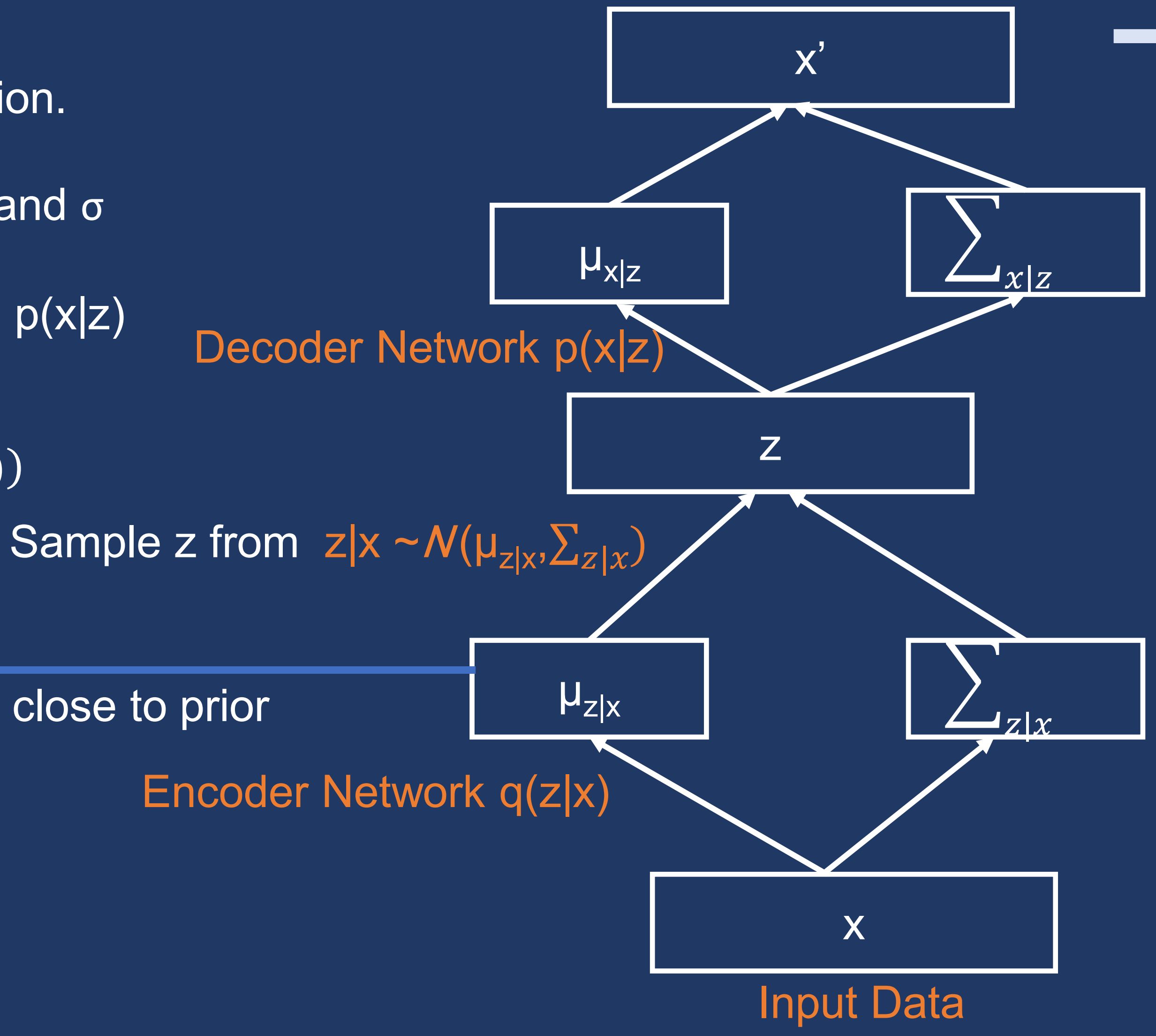
Variational Autoencoders: Statistical Motivation

- P(z) often assumed to be a Gaussian distribution.
- Determining q(z|x) boils down to estimating μ and σ
- Use neural networks for computing q (z|x) and p(x|z)

Loss function = $\boxed{E_z[\log p(x|z)]} - D_{KL}(q(z|x)||p(z))$

Maximizing the likelihood lower bound

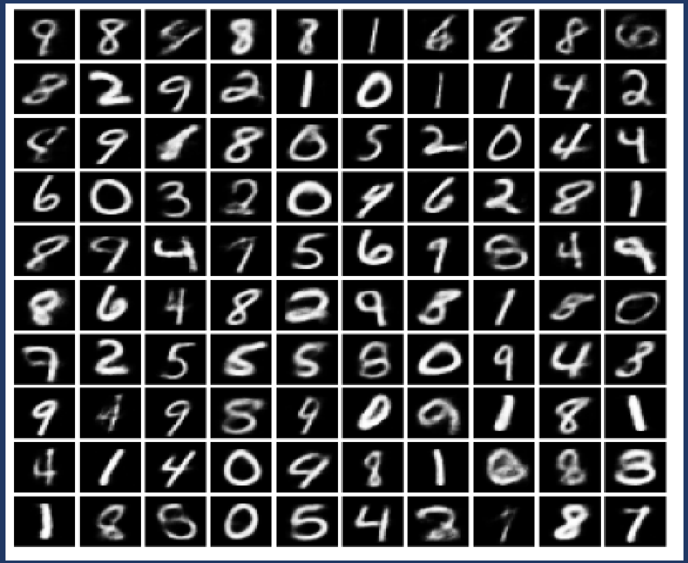
Make posterior distribution close to prior



*Autoencoding Variational Bayes (Kingma and Welling, 2013)

Variational autoencoders as generative models

- New data can be generated by sampling from the distributions in the latent space i.e. reconstructed by the decoder.
- Can encode different levels of variations.
- Example: smoothly varying of head pose and smile.



Samples from a VAE trained on MNIST

Samples from a VAE trained on a faces dataset

*Autoencoding Variational Bayes (Kingma and Welling, 2013)

17

Variational Autoencoders: Summary

- Probabilistic spin to traditional autoencoders => allows generating data
- Defines an intractable density => derive and optimize a (variational) lower bound

Pros:

- \- Principled approach to generative models
- \- Interpretable latent space.
- \- Allows inference of $q(z|x)$, can be useful feature representation for other tasks

Cons:

- Samples blurrier and lower quality compared to state-of-the-art (GANs)

Active areas of research:

- \- More flexible approximations, e.g. richer approximate posterior instead of diagonal
- Gaussian, e.g., Gaussian Mixture Models (GMMs), Categorical Distributions.
- \- Learning disentangled representations.

Content

- ➢ Insights into Generative Models
- ➢ Autoencoders
  - Variational Autoencoders
- ➢ Generative Adversarial Networks

Generative Adversarial Networks

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

Generative

Generative models which generate data similar to the training data. E.g. Variational Autoencoders (VAE)

Adversarial

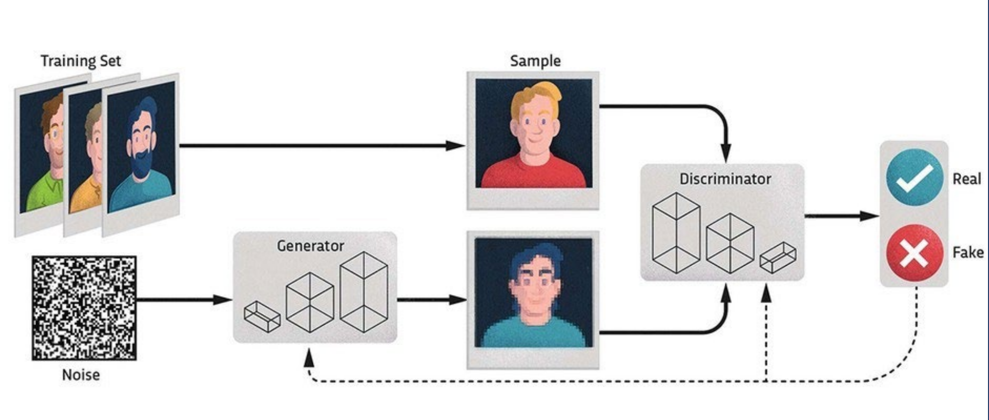
GANs are made up of two competing networks (adversaries) that are trying to beat each other.

Networks

Neural Networks

Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

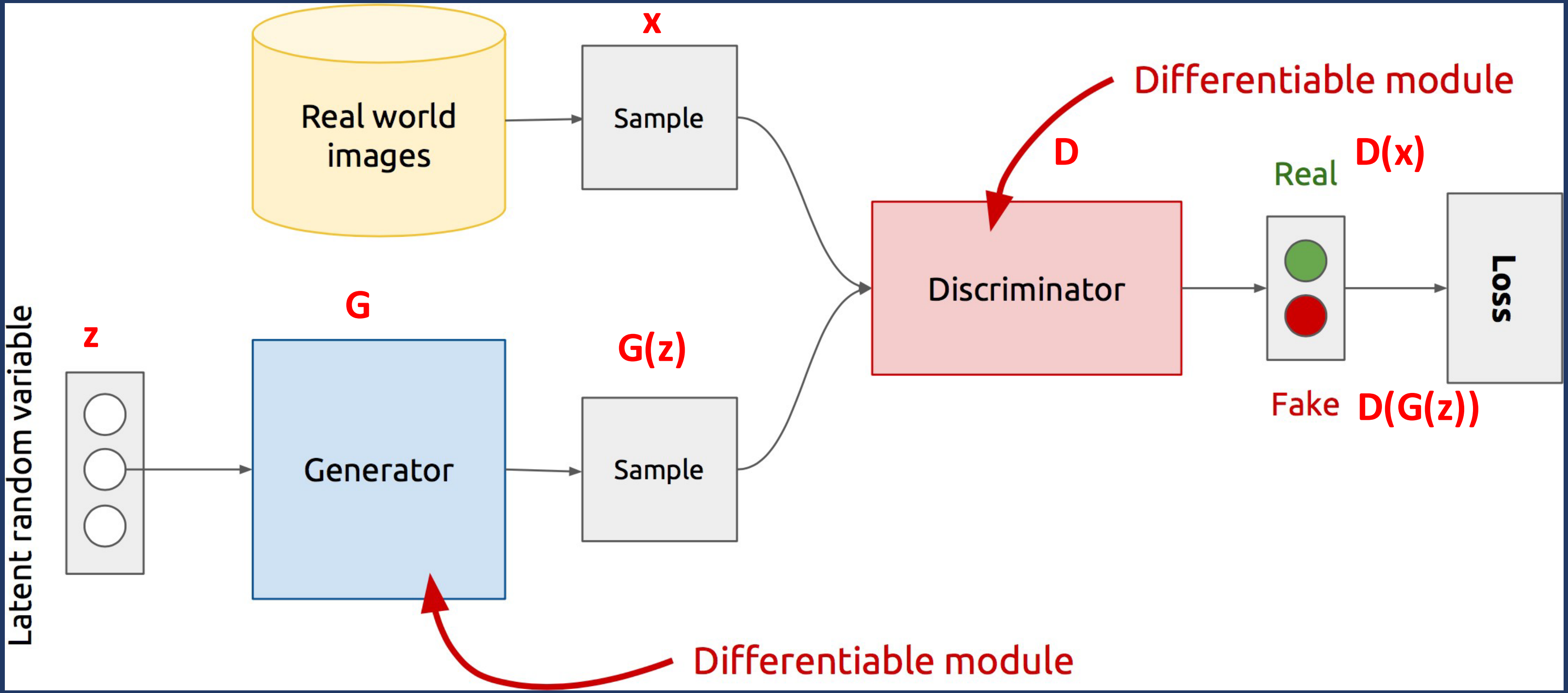
Solution: Sample from a simple distribution we can easily sample from, e.g. random noise.



Pic Source: https://blog.stackademic.com/generative-adversarial-networks-gans-and-their-applications-8df022b39939

thispersondoesnotexist.com

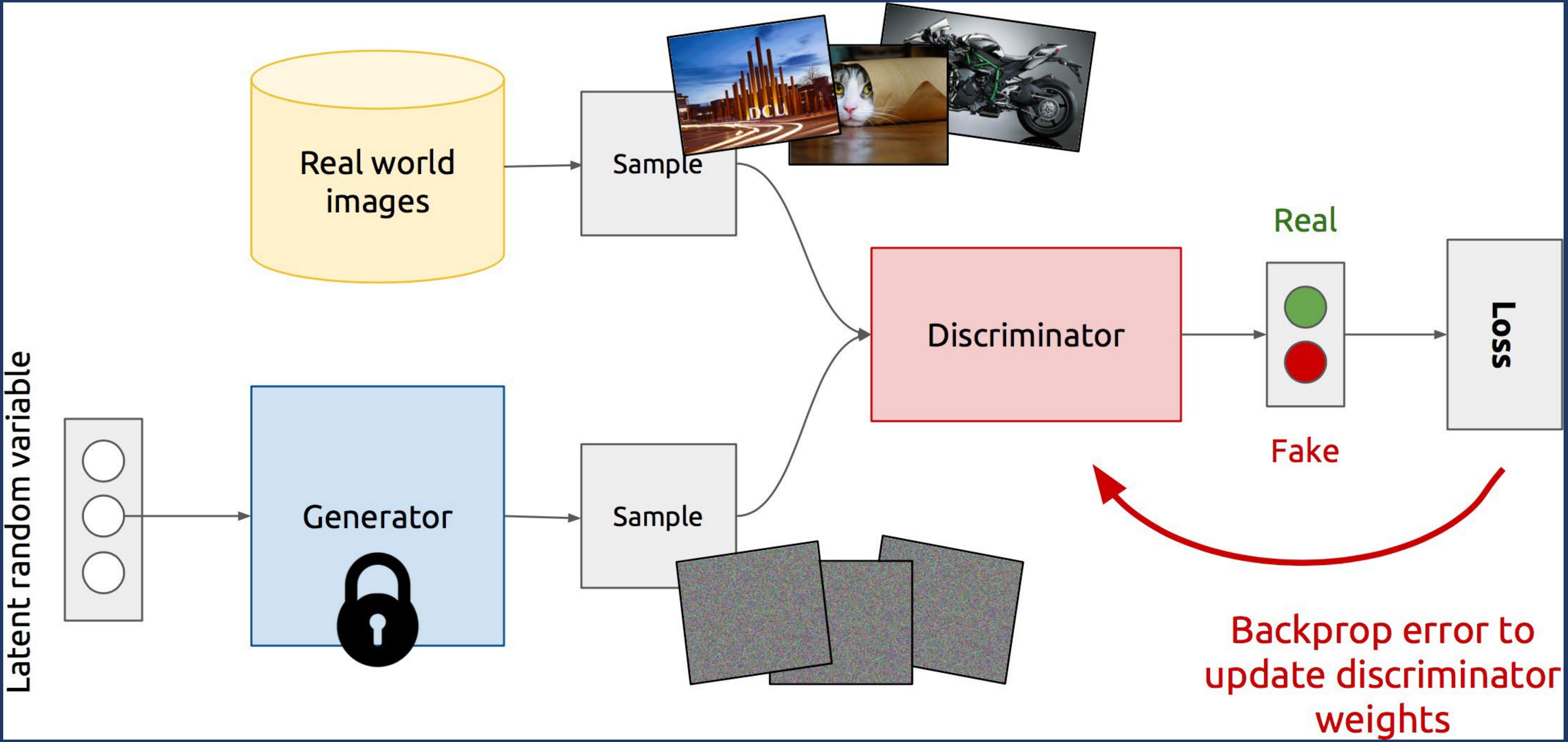
GANs Architecture



- **Z** is some random noise (Gaussian/Uniform).
- **Z** can be thought as the latent representation of the image.

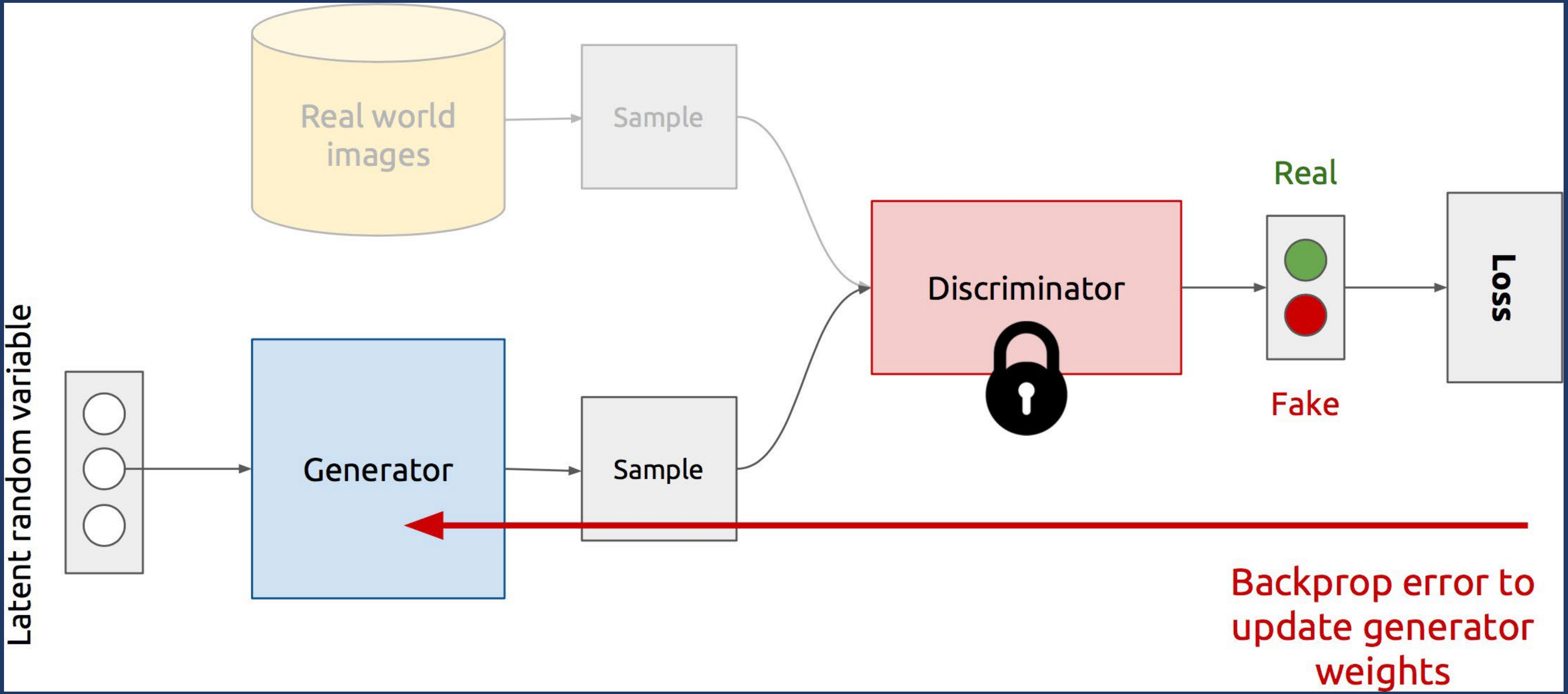
https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Training Discriminator



https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Training Generator



https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Training GANS: Two-Player Game

- Discriminator Network: trying to distinguish between real and fake images
- Generator Network: trying to fool the discriminator by generating real looking images
- Minmax objective function:

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

- Discriminator (θd) wants to **maximize objective** such that D(x) is close to 1 (real) and
- D(G(z)) is close to 0 (fake)
- Generator (θg) wants to **minimize objective** such that D(G(z)) is close to 1

Training GANS: Two-Player Game

Minimax objective function:

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

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

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

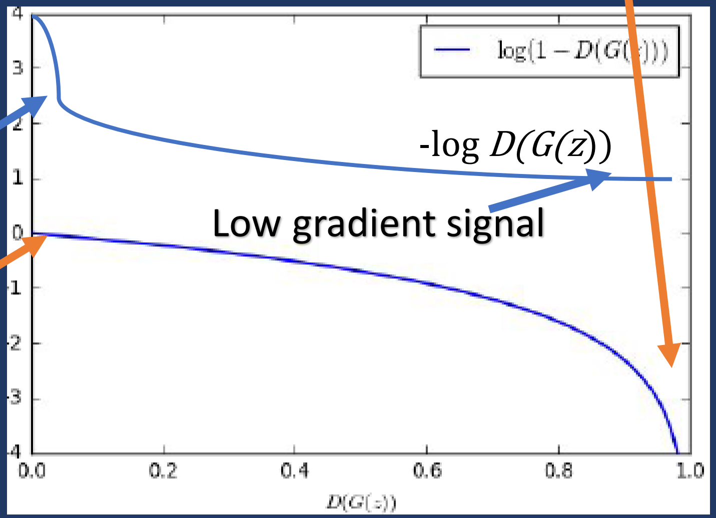
2. Gradient ascent on generator, higher gradient signals for bad samples work better

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

Gradient signal dominated by region where sample is really good

High gradient signal

But gradient in this region is flat



GAN Training Algorithm

Discriminator updates

Generator updates

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 $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ 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:

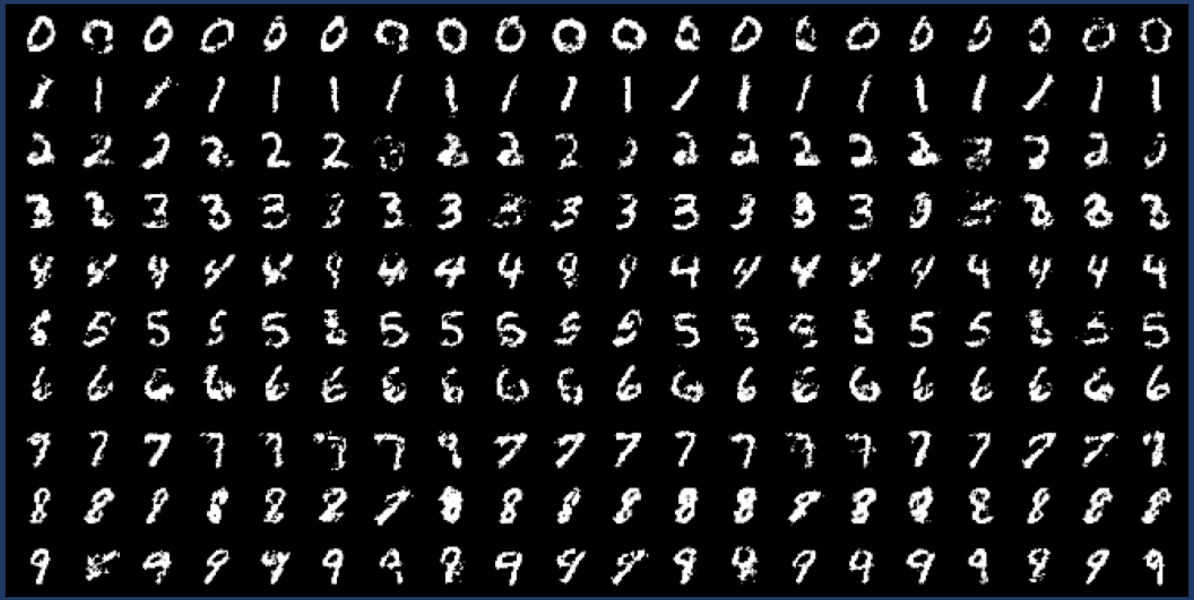
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

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

Conditional GANs

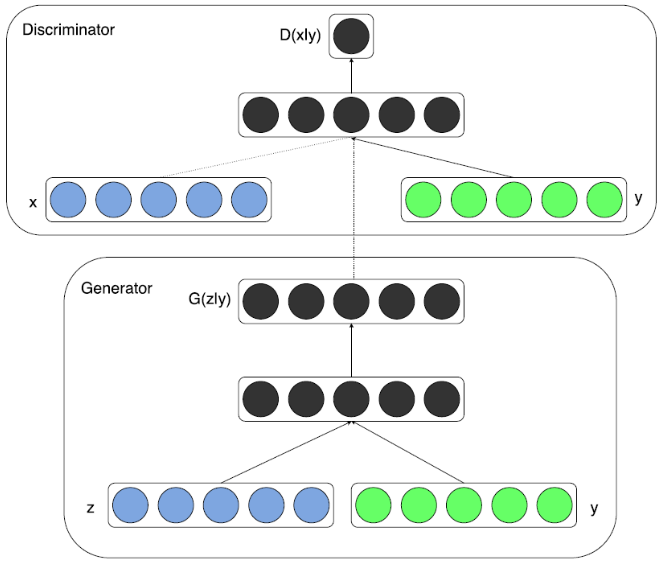
- Problem: Generator creates a "fake" generic image : is not specific for a certain condition/characteristic
- Example: text to image generation – image should depend on the text
- Idea: Provide additional vector y to networks to encode conditioning.



Generated samples conditioned on one label

Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: CoRR abs/1411.1784 (2014). arXiv: 1411.1784.

Conditional GANs



- Generator G receives the latent vector **z** and a conditioning vector **y**
- Discriminator D receives **x** and also **y**
- The objective function of a two-player minimax game changes to:

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

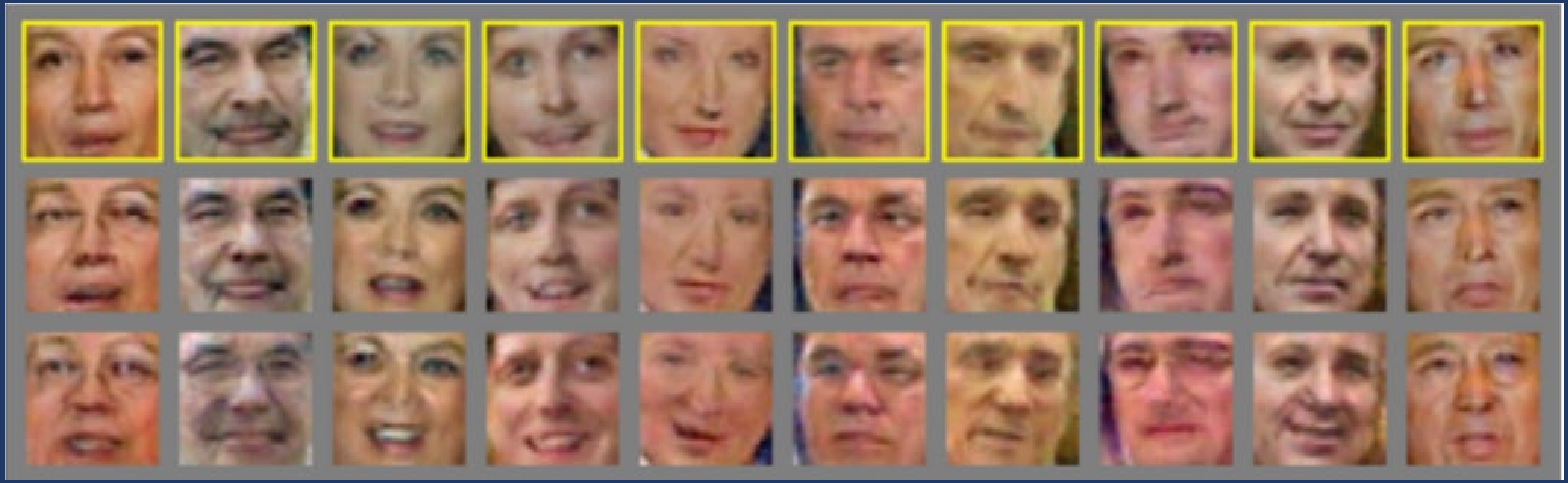
Mehdi Mirza and Simon Osindero. “Conditional Generative Adversarial Nets”. In: CoRR abs/1411.1784 (2014). arXiv: 1411.1784.

Example: Conditional GANs for Face Generation

- Add conditional feature (e.g., smiling, gender, old age, ...)
- Generator/Discriminator learn to operate in modes:
- Generator learns to generate a face with a certain attribute
- Discriminator learns to decide whether the face contains attribute

Old Age

Old Age
+
Smiling

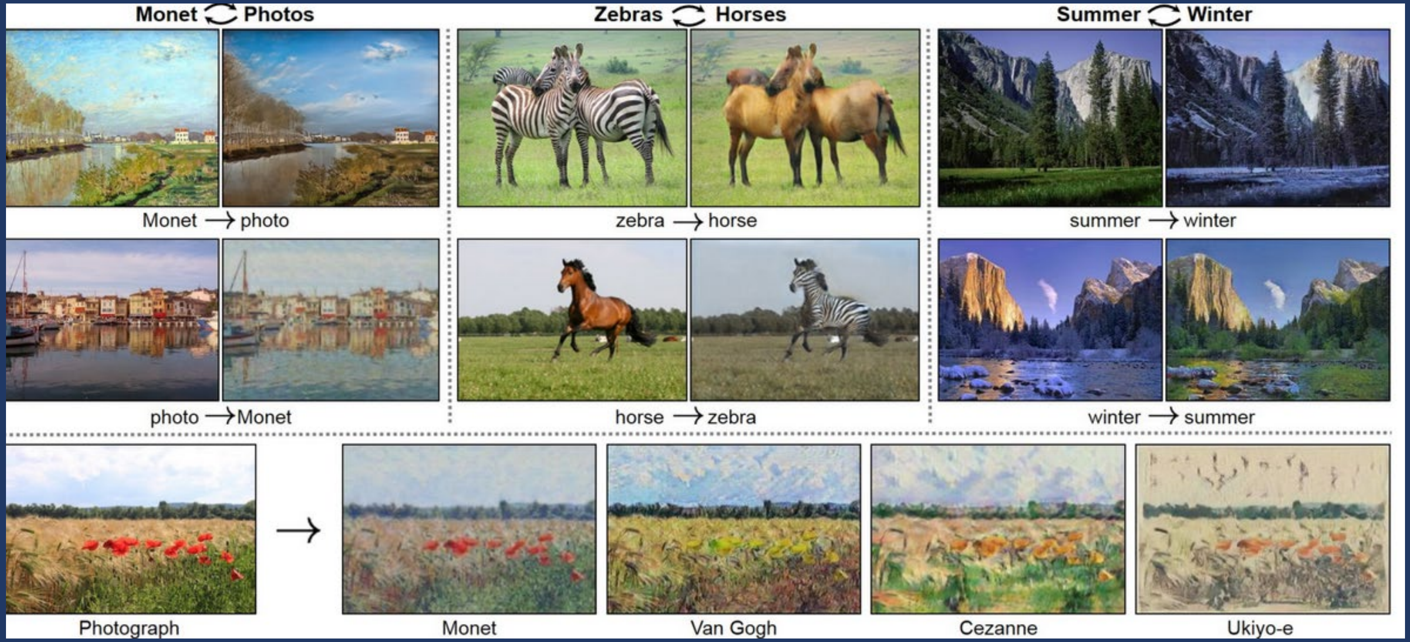


Mehdi Mirza and Simon Osindero. “Conditional Generative Adversarial Nets”. In: CoRR abs/1411.1784 (2014). arXiv: 1411.1784.

Cycle Consistent GANs

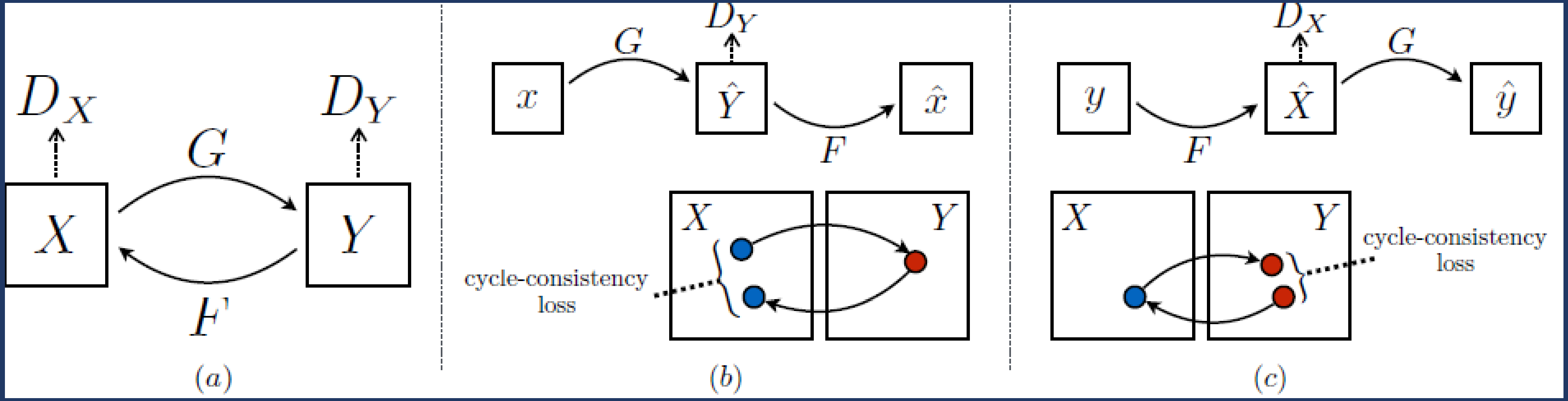
- Image to Image GAN should generate plausible results w.r.t. input
- Paired data difficult/impossible to obtain
- Cycle consistency loss: Couple GAN with trainable inverse mapping F such that

$$F(G(\mathbf{x})) \approx \mathbf{x} \text{ and } G(F(\mathbf{y})) \approx \mathbf{y}$$



Jun-Yan Zhu et al. “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”. In: CoRR abs/1703.10593 (2017). arXiv: 1703.10593.

Cycle Consistent GANs



- Two discriminators D_Y and D_X
- Cycle consistency loss for two generators G, F

$$L_{cyc}(G, F) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\left[\|F(G(\mathbf{x})) - \mathbf{x}\|_1\right] + \mathbb{E}_{\mathbf{y}\sim p_{data}(\mathbf{y})}\left[\|G(F(\mathbf{y})) - \mathbf{y}\|_1\right]$$

- Total Loss

:

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F)$$

Problems with GANs

- **Probability Distribution is Implicit**
  - Not straightforward to compute P(X).
  - Thus **Vanilla GANs** are only good for Sampling/Generation.

- **Training is Hard**
  - Non-Convergence
  - Mode-Collapse

Goodfellow, Ian. **"NIPS 2016 Tutorial: Generative Adversarial Networks."** arXiv preprint arXiv:1701.00160 (2016).

Training Problems

- **Non-Convergence**
- Mode-Collapse

Training Problems

- **Deep Learning models (in general) involve a single player**
  - The player tries to maximize its reward (minimize its loss).
  - Use SGD (with Backpropagation) to find the optimal parameters.
  - SGD has convergence guarantees (under certain conditions).
  - **Problem:** With non-convexity, we might converge to local optima.

$$\min_G L(G)$$

- **GANs instead involve two (or more) players**
  - Discriminator is trying to maximize its reward.
  - Generator is trying to minimize Discriminator's reward.

$$\min_G \max_D V(D, G)$$

  - SGD was not designed to find the Nash equilibrium of a game.
  - **Problem:** We might not converge to the Nash equilibrium at all.

Salimans, Tim, et al. "Improved techniques for training gans." *Advances in Neural Information Processing Systems*. 2016.

Non-Convergence

$$\min_x \max_y V(x, y)$$

Let $V(x, y) = xy$

• State 1:	$x > 0$	$y > 0$	$V > 0$	Increase y	Decrease x
• State 2:	$x < 0$	$y > 0$	$V < 0$	Decrease y	Decrease x
• State 3:	$x < 0$	$y < 0$	$V > 0$	Decrease y	Increase x
• State 4 :	$x > 0$	$y < 0$	$V < 0$	Increase y	Increase x
• State 5:	$x > 0$	$y > 0$	$V > 0$ == State 1	Increase y	Decrease x

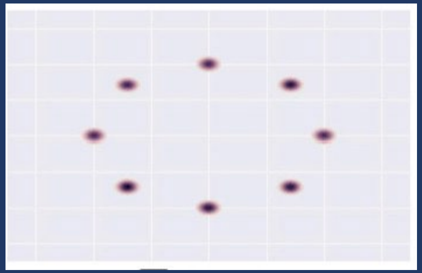
Problems with GANs

- Non-Convergence
- **Mode-Collapse**

Mode-Collapse

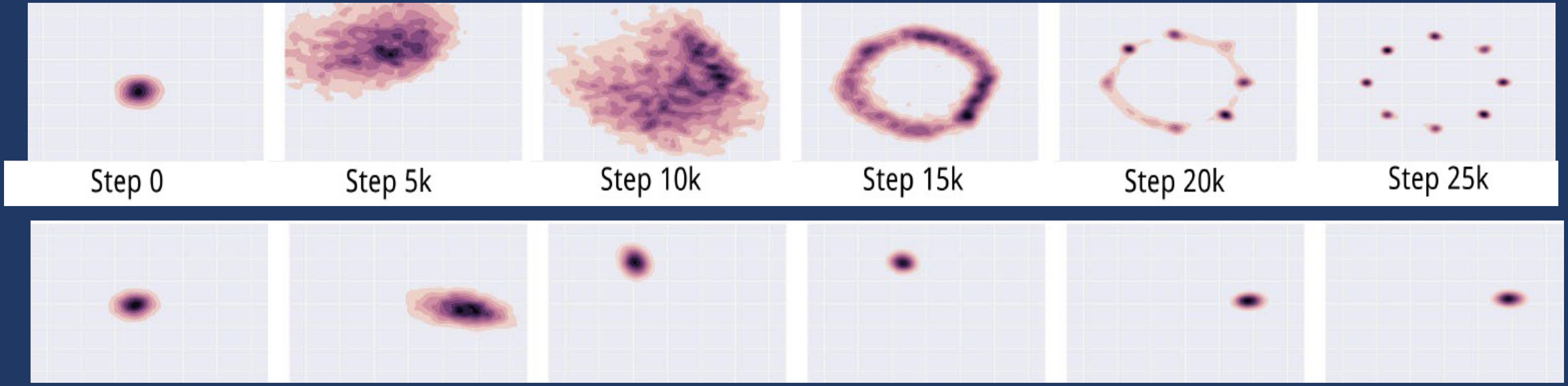
- **Generator fails to output diverse samples**

Target



Expected

Output



Metz, Luke, et al. **"Unrolled Generative Adversarial Networks."** arXiv preprint arXiv:1611.02163 (2016)

Some Solutions

- **Mini-Batch GANs**
- **Supervision with labels**

- **Some recent attempts :-**
  - Unrolled GANs
  - W-GANs

Basic (Heuristic) Solutions

- **Mini-Batch GANs**
- Supervision with labels

How to reward sample diversity?

- **At Mode Collapse,**
  - Generator produces good samples, but a very few of them.
  - Thus, Discriminator can't tag them as fake.

- **To address this problem,**
  - Let the Discriminator know about this edge-case.

- **More formally,**
  - Let the Discriminator look at the entire batch instead of single examples
  - If there is lack of diversity, it will mark the examples as fake

- **Thus,**
  - Generator will be forced to produce diverse samples.

Salimans, Tim, et al. "Improved techniques for training gans." *Advances in Neural Information Processing Systems*. 2016.

Mini-Batch GANs

- **Extract features that capture diversity in the mini-batch**
  - For e.g. L2 norm of the difference between all pairs from the batch
- **Feed those features to the discriminator along with the image**
- **Feature values will differ b/w diverse and non-diverse batches**
  - Thus, Discriminator will rely on those features for classification
- **This in turn,**
  - Will force the Generator to match those feature values with the real data
  - Will generate diverse batches

Salimans, Tim, et al. "Improved techniques for training gans." *Advances in Neural Information Processing Systems*. 2016.

Basic (Heuristic) Solutions

- Mini-Batch GANs
- **Supervision with labels**

Supervision with Labels

- Label information of the real data might help

- Empirically generates much better samples

Salimans, Tim, et al. "Improved techniques for training gans." *Advances in Neural Information Processing Systems*. 2016.

Image-to-Image Translation

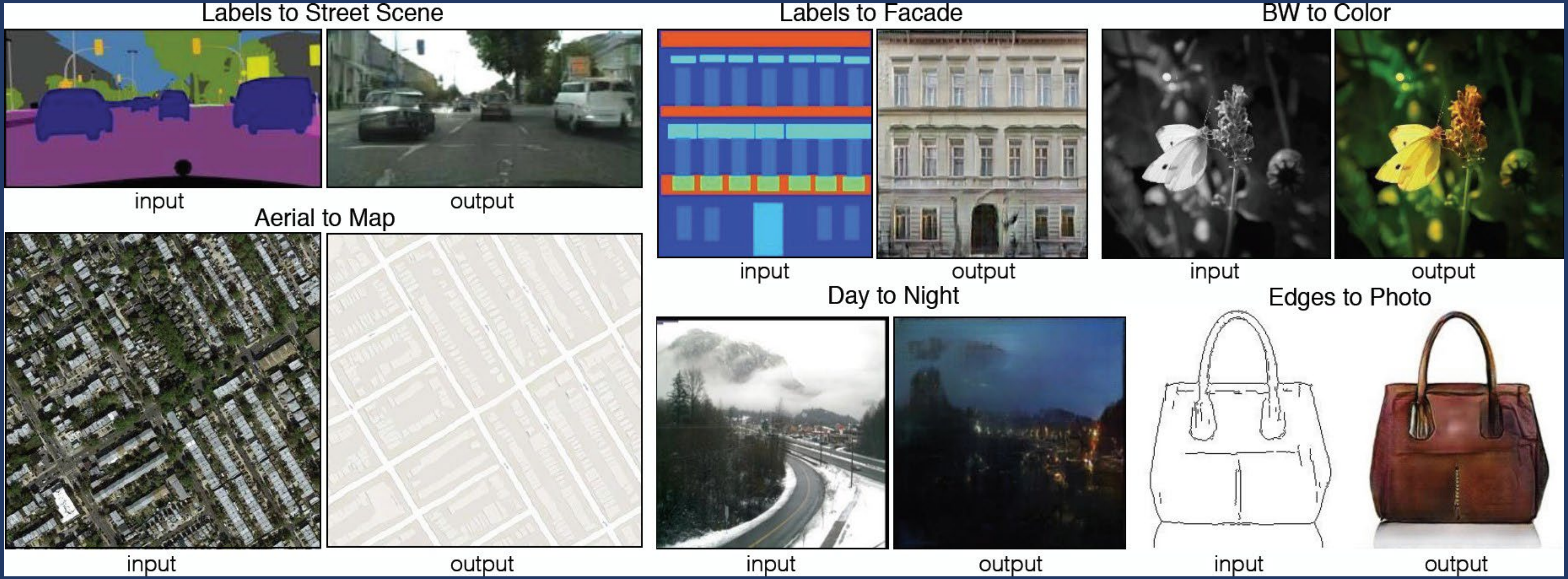


Figure 1 in the original paper.

[Link to an interactive demo of this paper]

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. "**Image-to-image translation with conditional adversarial networks**". arXiv preprint arXiv:1611.07004. (2016).

Text-to-Image Synthesis

Motivation

Given a text description, generate images closely associated.

Uses a conditional GAN with the generator and discriminator being condition on "dense" text embedding.

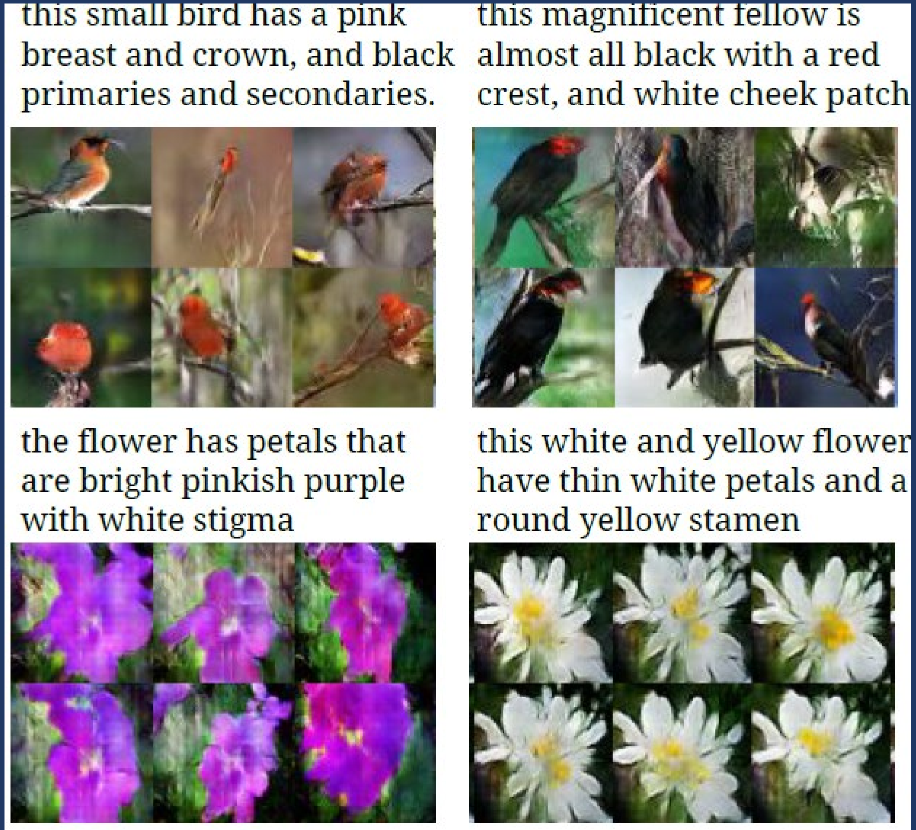
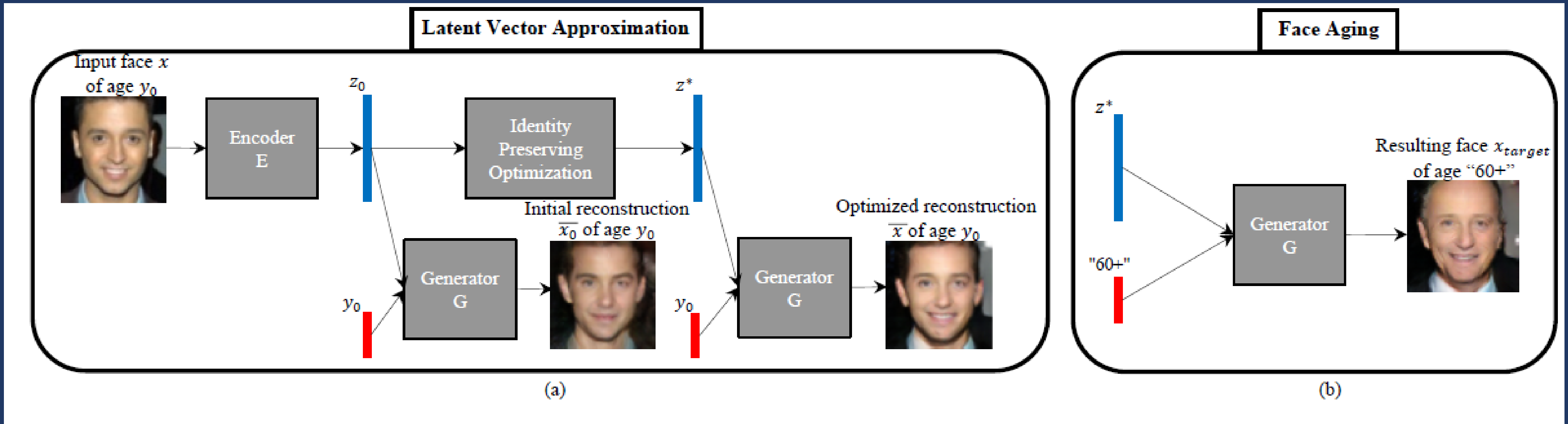


Figure 1 in the original paper.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. "**Generative adversarial text to image synthesis**". ICML (2016).

Face Aging with Conditional GANs

- Differentiating Feature: Uses an *Identity Preservation Optimization* using an auxiliary network to get a better approximation of the latent code (z*) for an input image.
- Latent code is then conditioned on a discrete (one-hot) embedding of age categories.



Face Aging with Conditional GANs

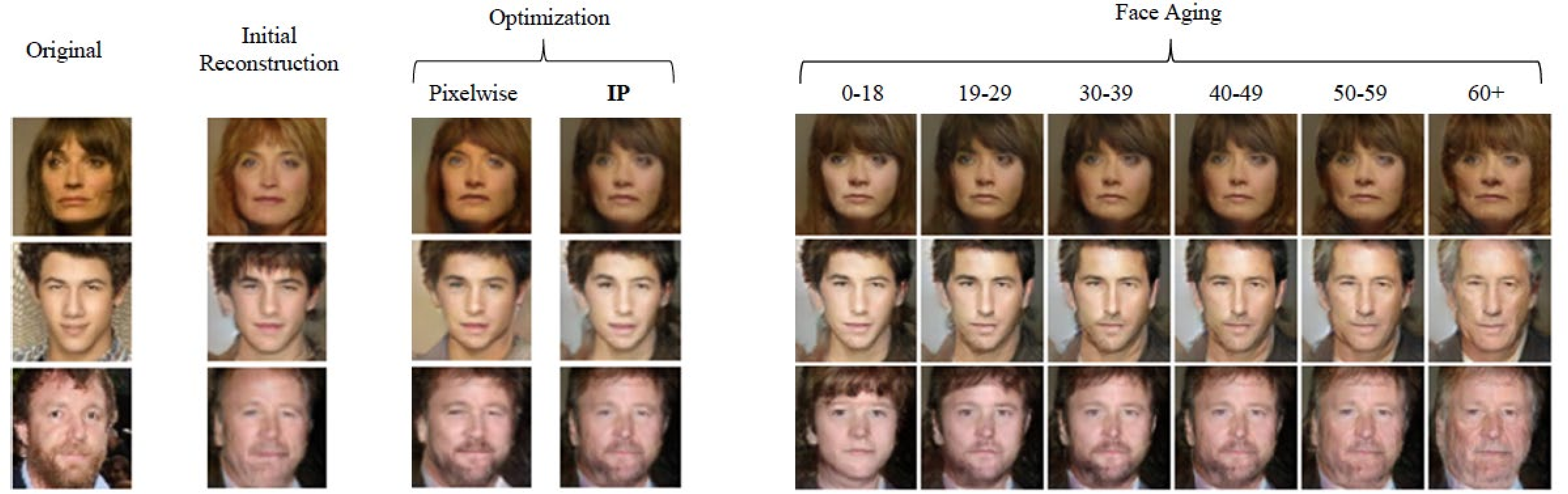
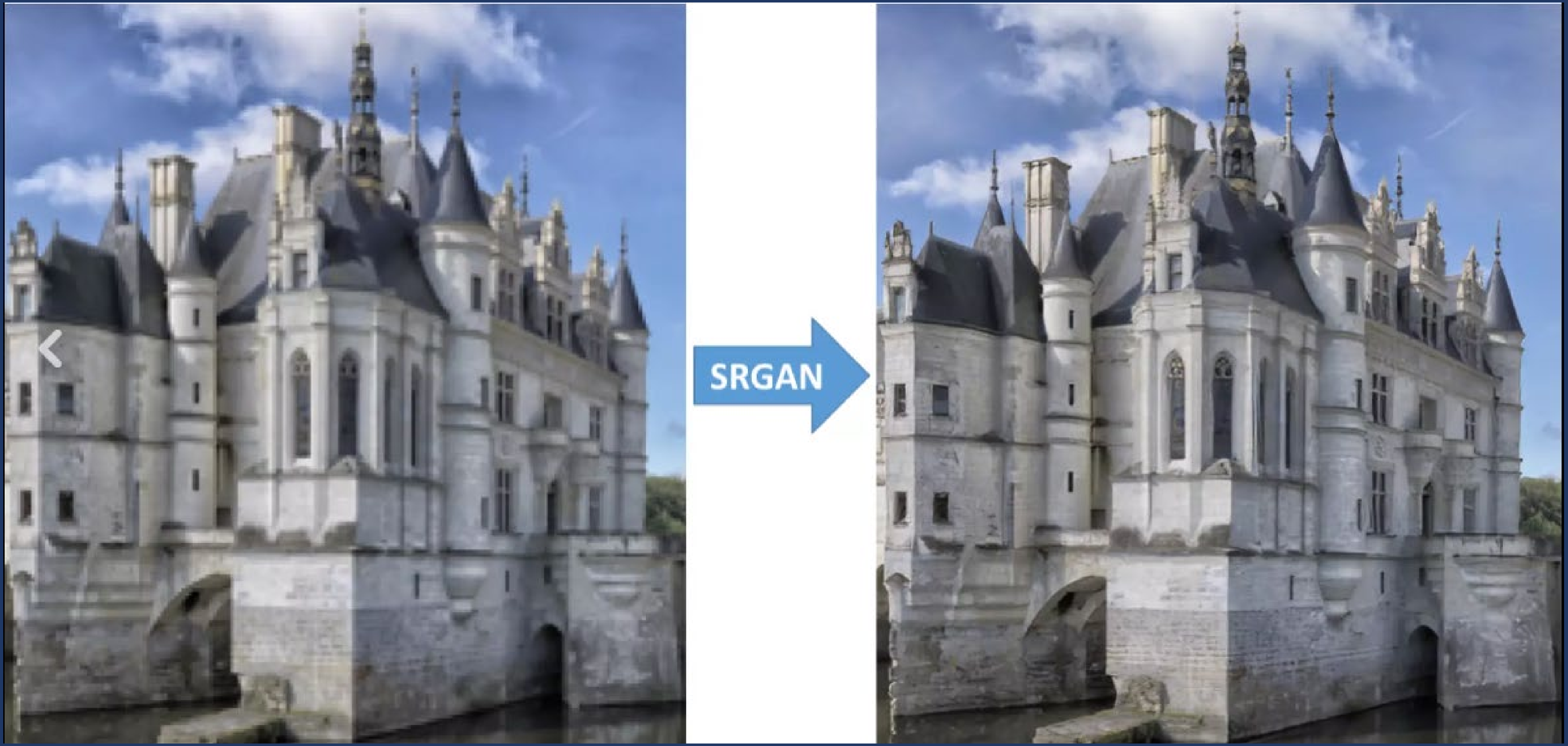


Figure 3 in the original paper.

Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). "**Face Aging With Conditional Generative Adversarial Networks**". arXiv preprint arXiv:1702.01983.

Image Superresolution:



Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *CVPR* 2017

Summary of GANs

- ➢ Don't work with an explicit density function
- ➢ Take game-theoretic approach: learn to generate from training distribution through 2-player game

Pros:

- ➢ - Beautiful, state-of-the-art samples!

Cons:

- ➢ - Trickier / more unstable to train
- ➢ - Can't solve inference queries such as p(x), p(z|x)

Active areas of research:

- ➢ - Better loss functions, more stable training (Wasserstein GAN, LSGAN, many others)
- ➢ - Conditional GANs, GANs for all kinds of applications

Acknowledgements

There are lots of excellent references on GANs :

- https://cs236g.stanford.edu/
- Sebastian Nowozin's presentation at MLSS 2018.
- NIPS 2016 tutorial on GANs by Ian Goodfellow.

by Alex Irpan.

References

- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. Generative adversarial nets, NIPS (2014).
- Goodfellow, Ian NIPS 2016 Tutorial: Generative Adversarial Networks, NIPS (2016).
- Radford, A., Metz, L. and Chintala, S., Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434. (2015).
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. Improved techniques for training gans. NIPS (2016).
- Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets, NIPS (2016).
- Zhao, Junbo, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126 (2016).
- Mirza, Mehdi, and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014).
- Liu, Ming-Yu, and Oncel Tuzel. Coupled generative adversarial networks. NIPS (2016).
- Denton, E.L., Chintala, S. and Fergus, R., 2015. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. NIPS (2015)
- Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., & Courville, A. Adversarially learned inference. arXiv preprint arXiv:1606.00704 (2016).

Applications:

- Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004. (2016).
- Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. Generative adversarial text to image synthesis. JMLR (2016).
- Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). Face Aging With Conditional Generative Adversarial Networks. arXiv preprint arXiv:1702.01983.

Coming up…

- Recurrent Neural Networks
- Transformers and Self Attention

Deep Learning

Summer semester '24

5. Autoencoders & Generative Adversarial Networks