Generative Adversarial Networks (GANs)

Slides based on :Ian Goodfellow’s Tutorial and Deep Learning Book of Ian Goodfellow et al.

In This Session

- Introduction to Generative Adversarial Network (GAN)
- Challenges with GAN
- Applications of GAN
- Recent developments using GAN

GANs

- Generative Adversarial Networks
  - Learn a generative model, trained in an adversarial setting. Deep Neural Networks are used.
- Generative models
  - Learn to generate data samples that are like the real data samples on which the model is trained.

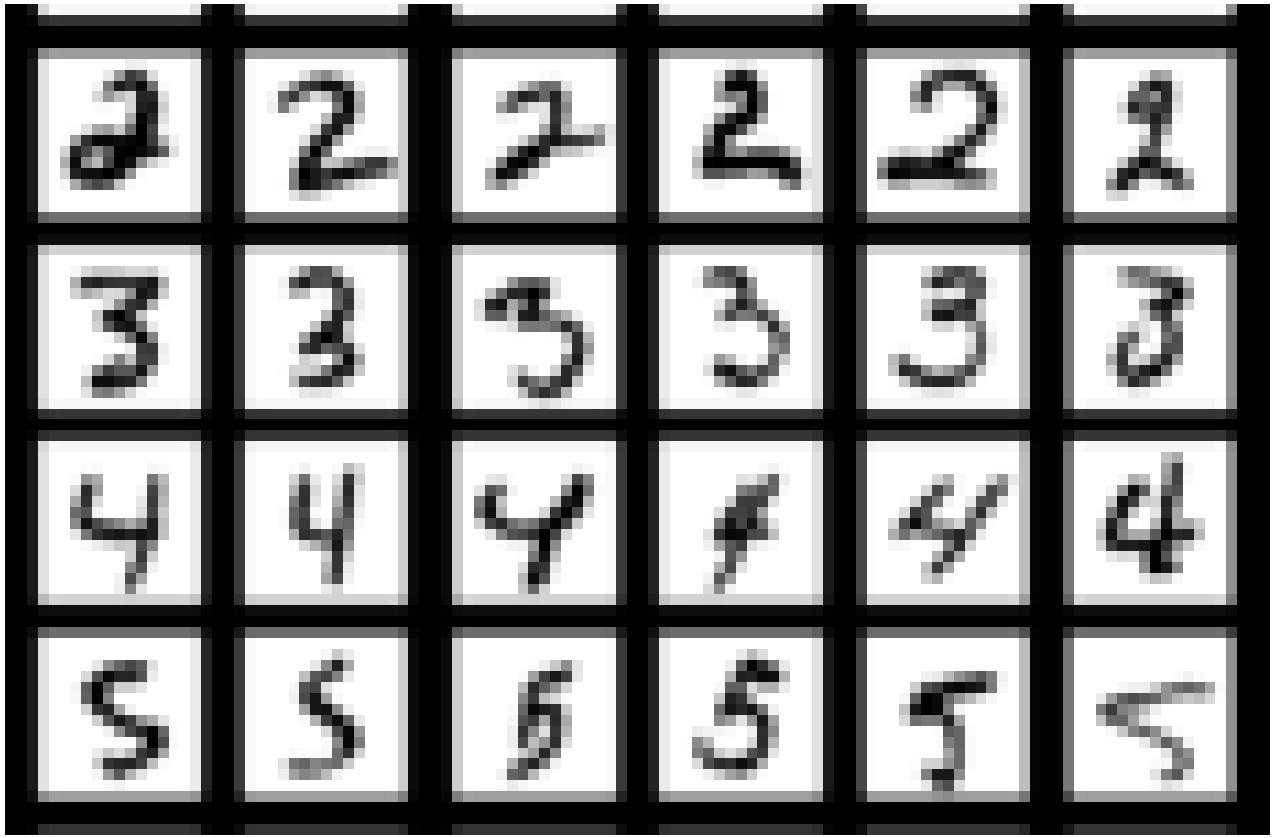
Discriminative v/s Generative Models

- The discriminative models are trained to predict a label Y, Given a data sample X.
- They are trained to model the conditional probability of the target Y, given an observation x of a random variable X, $P(Y|X=x)$.

- Generative models are trained to model the conditional probability of the observable X, given a target y, i.e., $P(X|Y=y)$.
- They are the statistical models of the joint probability distribution $P(X, Y)$.

Generative Models

- Example: Given a data sample of handwritten digits, the generative models learns the patterns and their variation for each class to create its own samples.



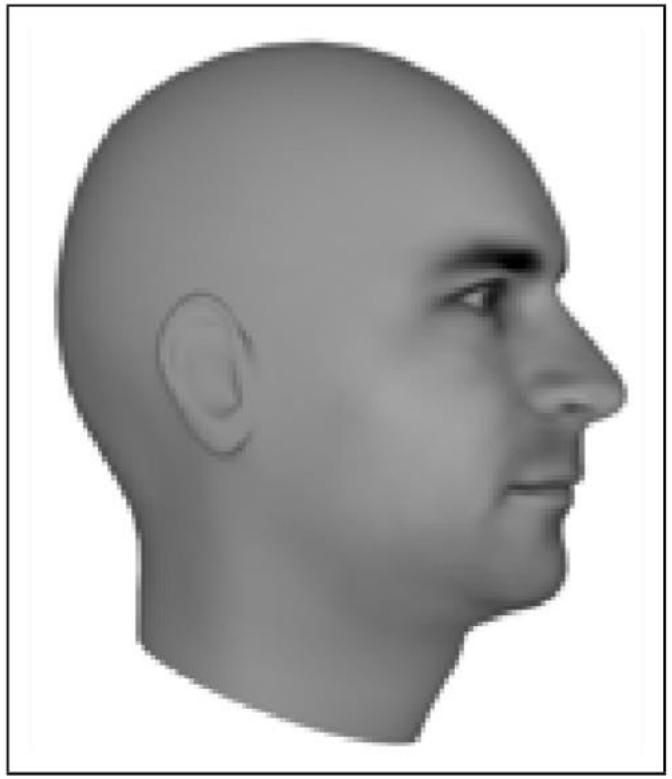
GAN Generated Output…

Can you guess which one is real ?

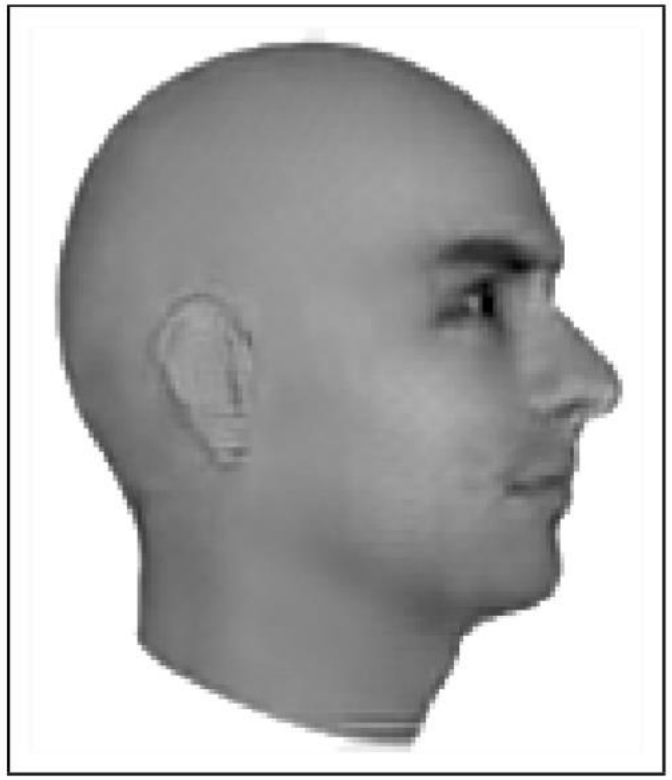
Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *arXiv preprint arXiv:1609.04802* (2016).

GAN Generated Output

Ground Truth

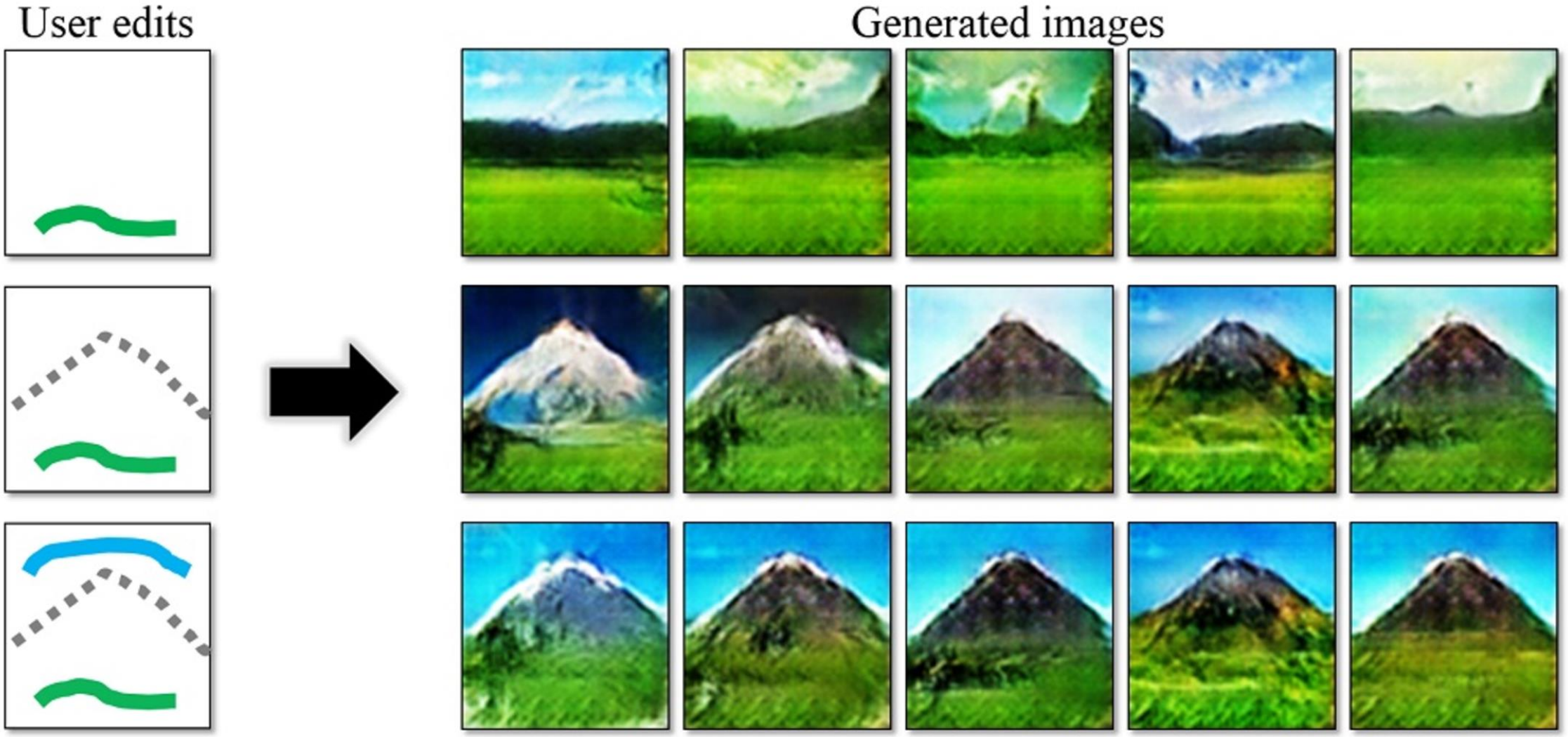


Adversarial



Lotter, William, Gabriel Kreiman, and David Cox. "Unsupervised learning of visual structure using predictive generative networks." *arXiv preprint arXiv:1511.06380* (2015).

GAN Generated Output…



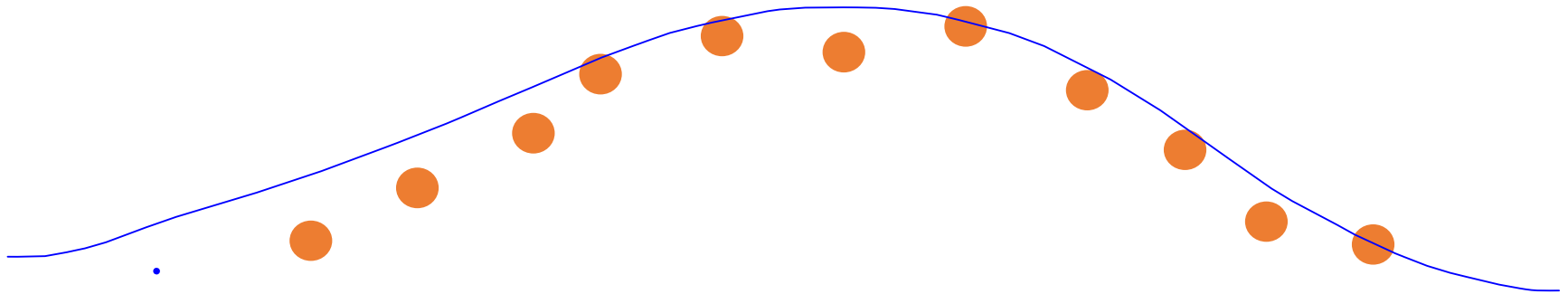
http://people.eecs.berkeley.edu/~junyanz/projects/gvm/

Generative Model for Density Estimation

- Addresses a core problem in unsupervised learning: Density estimation.
- What is density estimation ?
- Density estimation deals with estimation of the probability distribution of data samples –what is the likely data distribution ?

Density Estimation

- For example, you would like to estimate the density function of COVID affected people of different age groups in a state by collecting data through randomly selected locations in a survey.



Taxonomy of Generative Models

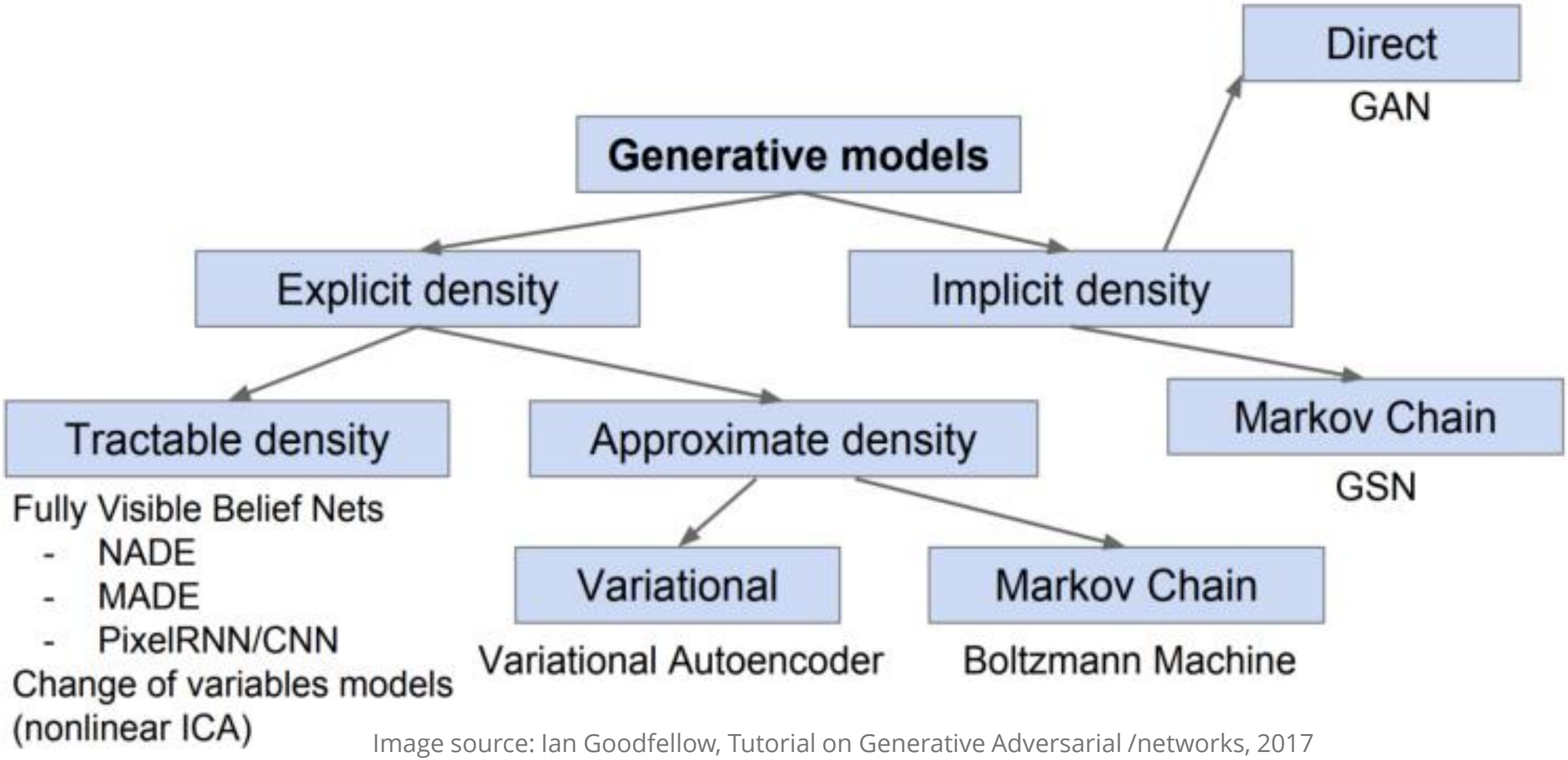


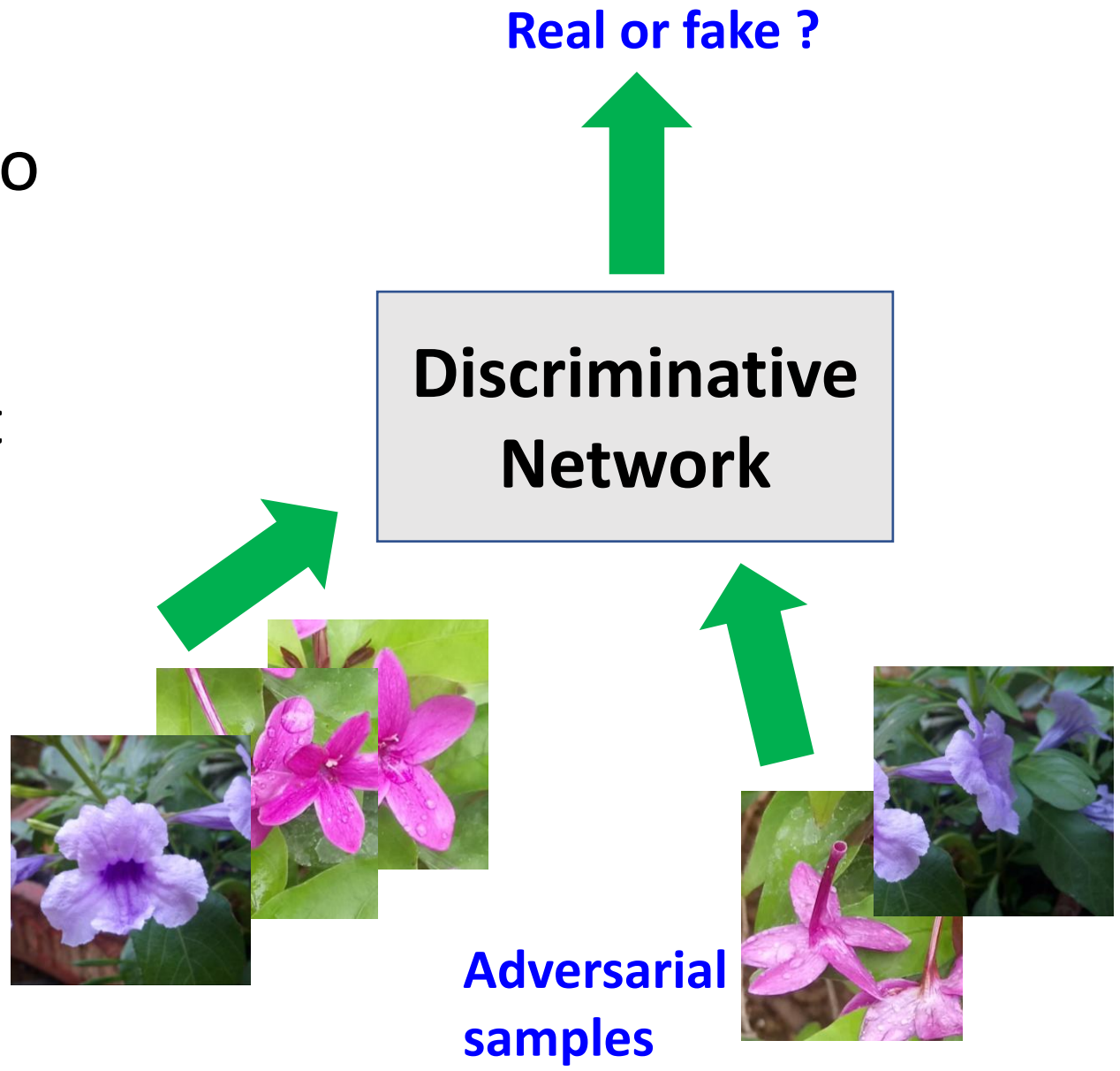
Image source: Ian Goodfellow, Tutorial on Generative Adversarial /networks, 2017

Generative Models

- **Explicit density estimation**: Explicitly define and solve for $P_{model}(x)$.
- **Implicit density estimation**: learn a model that can sample from $P_{model}(x)$ without explicitly defining it.
- **Some Commonly used models** –
  - Explicit models
    - Variational Autoencoder
    - PixelRNN/CNN
  - Implicit models
    - GAN
    - Generative Stochastic Model

Adversarial Training

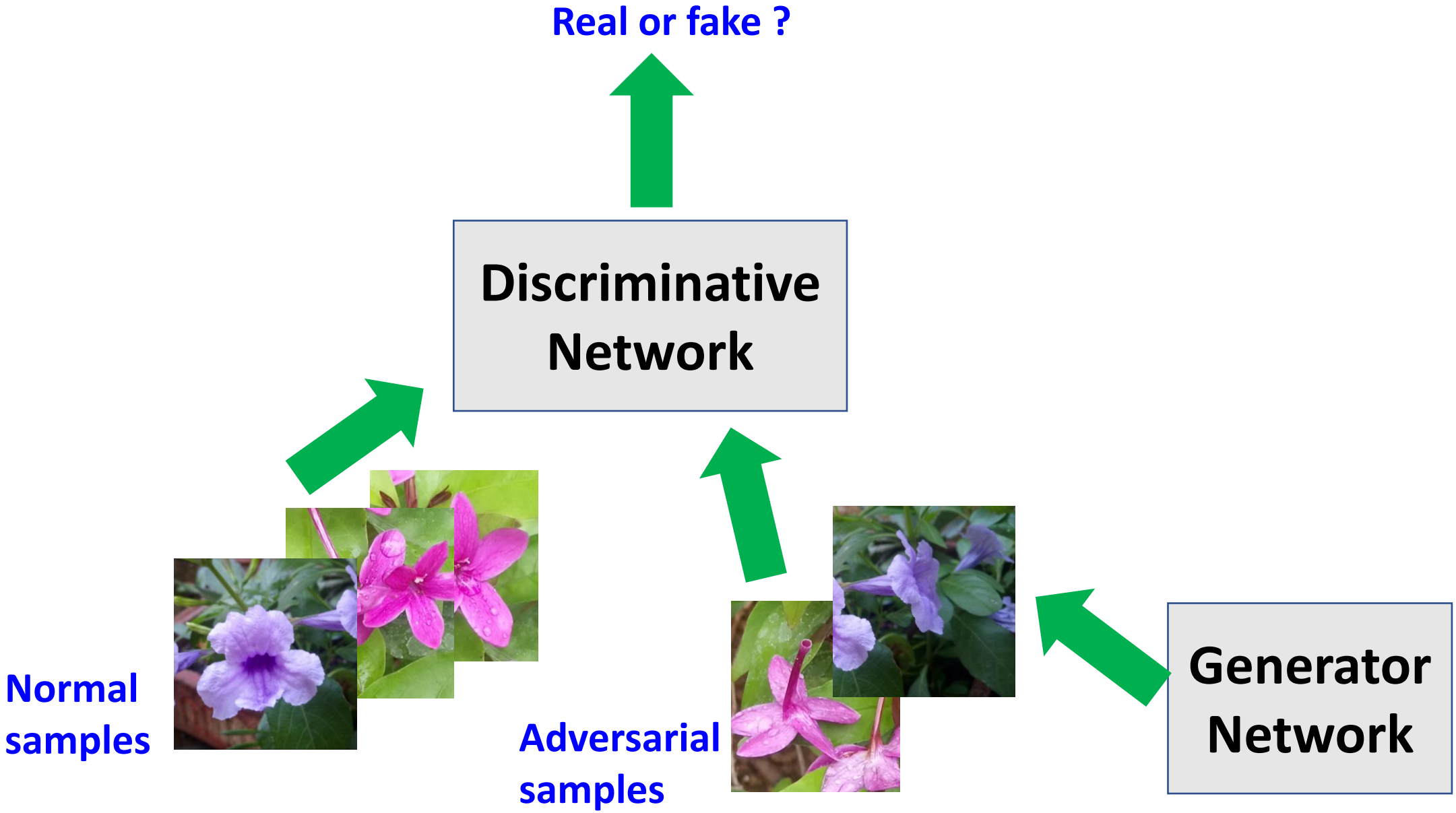
- **In adversarial Training:** We generate adversarial samples to fool a discriminative model
  - We can use those adversarial samples to make models robust
  - We then require more effort to generate adversarial samples
  - Repeat this and we get better discriminative model



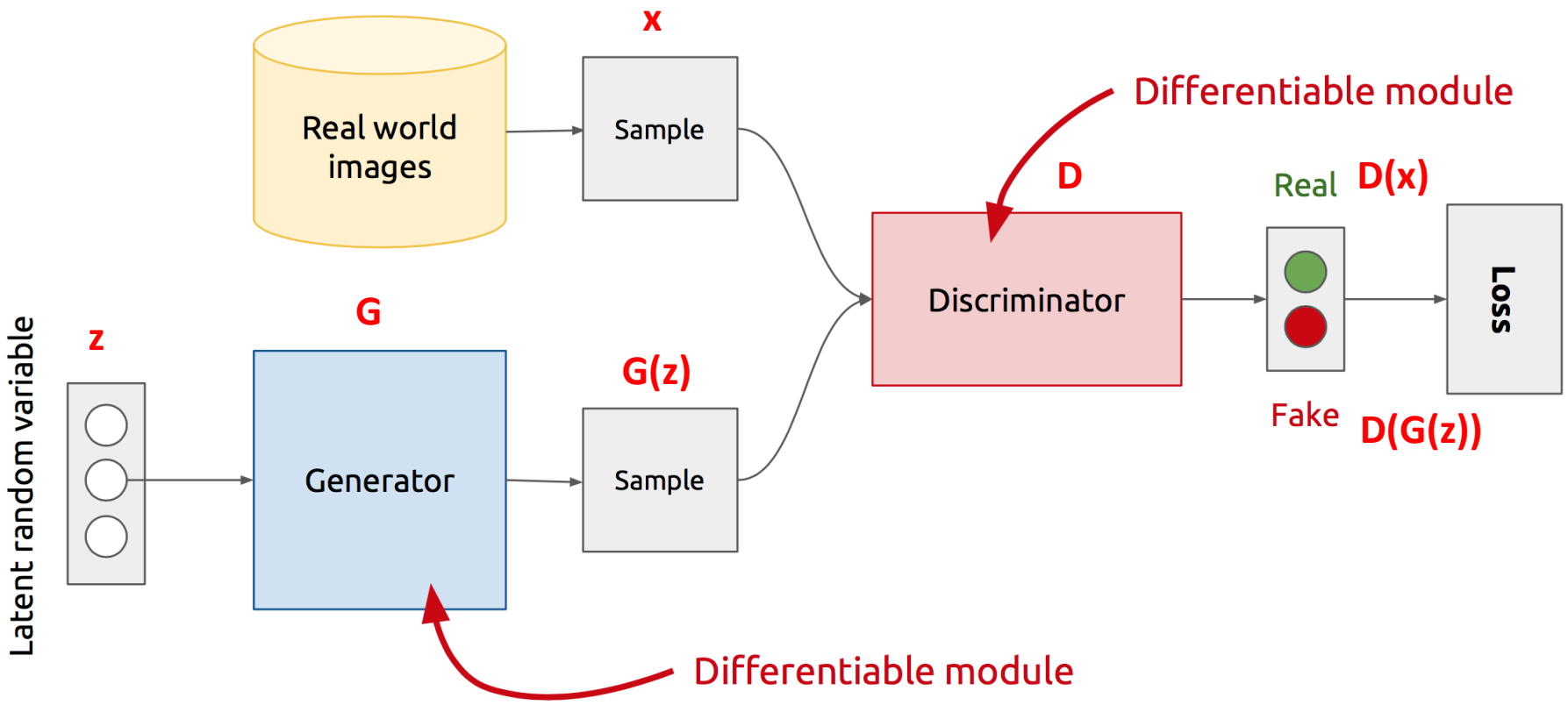
Adversarial Training

- **In adversarial Training of GAN**
- Generator: generate fake samples, tries to fool the Discriminator
- Discriminator: tries to distinguish between real and fake samples
- Train them against each other
- Repeat this and we get better Generator and Discriminator

Adversarial Training of GAN



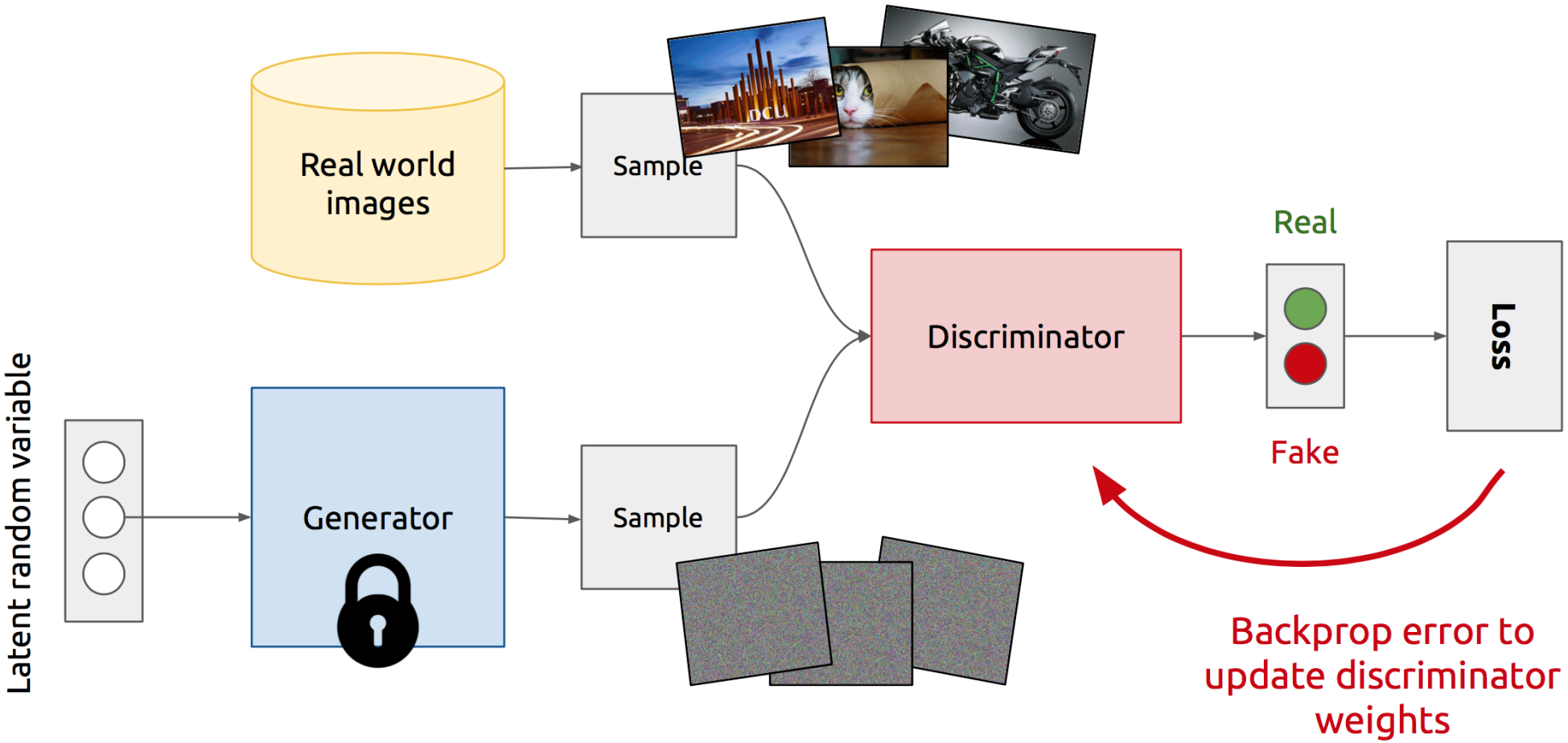
GAN's Architecture



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

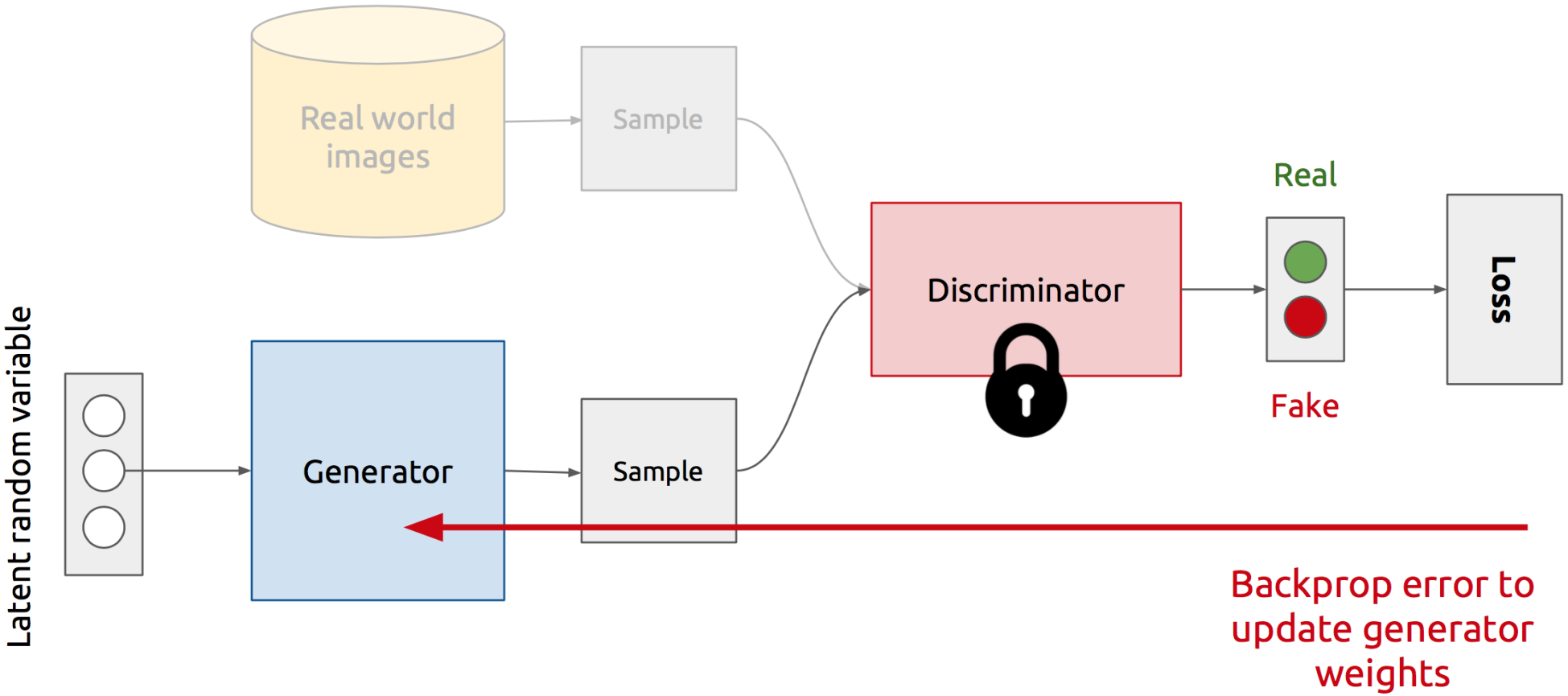
https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Training Discriminator



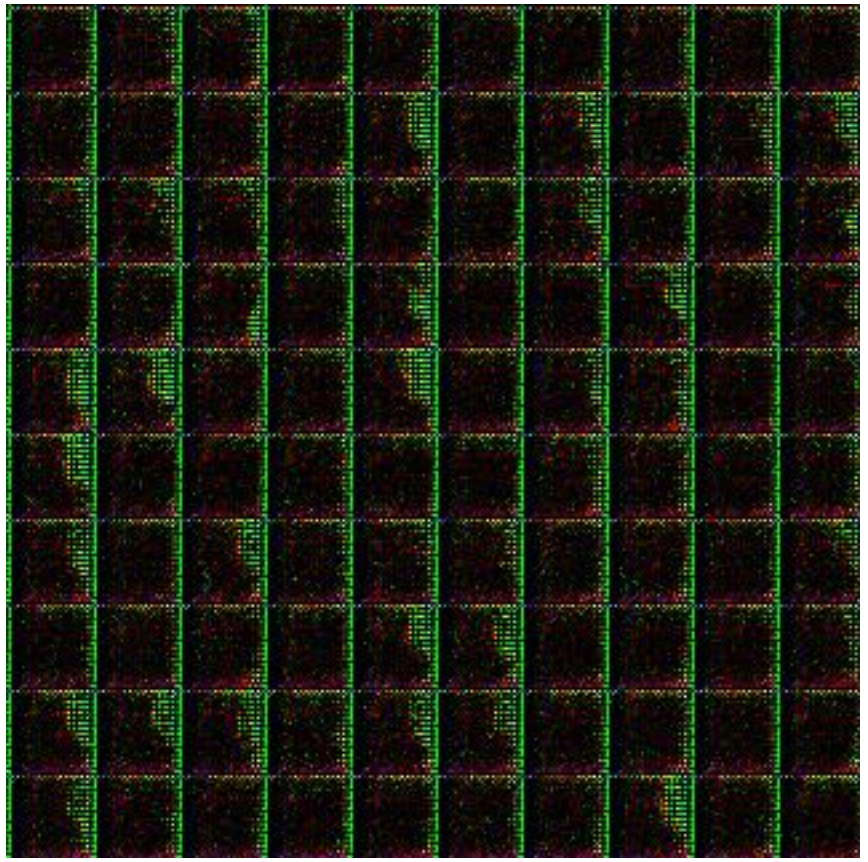
https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Training Generator



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

Generator in Action



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

GAN's Formulation

$$\min_{G} \max_{D} V(D, G)$$

- It is formulated as a **minimax game**, where:
  - The Discriminator is trying to maximize its reward $\boldsymbol{V(D, G)}$
  - The Generator is trying to minimize Discriminator's reward (or maximize its loss)

$$V(D, G) = \mathbb{E}_{x \sim p(x)}[\log D(x)] + \mathbb{E}_{z \sim q(z)}\left[\log\left(1 - D(G(z))\right)\right]$$

- The Nash equilibrium of this particular game is achieved at:
  - $P_{data}(x) = P_{gen}(x) \;\; \forall x$
  - $\mathrm{D}(x) = \frac{1}{2} \;\; \forall x$

Mental Break 1

Who introduced GAN?

Ian Goodfellow

Image source: https://www.technologyreview.com/

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.

Discriminator updates

Generator updates

Vanishing Gradient Problem

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

$$V(D,G) = \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim q(z)}\left[\log\left(1 - D(G(z))\right)\right]$$

$$\nabla_{\theta_G} V(D,G) = \nabla_{\theta_G} \mathbb{E}_{z\sim q(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

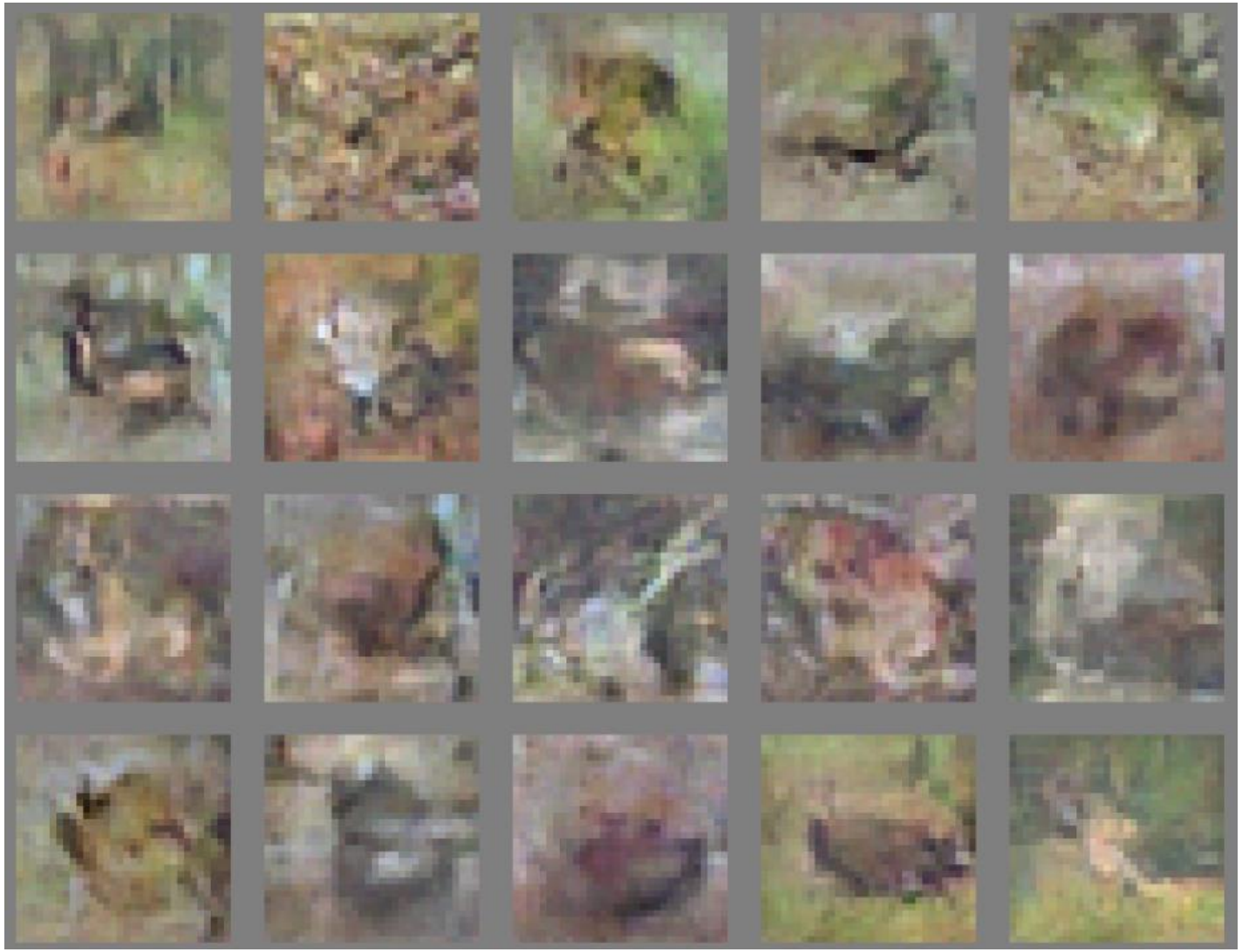
- $\nabla_a \log\left(1-\sigma(a)\right) = \frac{-\nabla_a\,\sigma(a)}{1-\sigma(a)} = \frac{-\sigma(a)\,(1-\sigma(a))}{1-\sigma(a)} = -\sigma(a) = -D\left(G(z)\right)$
- Gradient goes to 0 if D is confident, i.e. $D\left(G(z)\right) \to 0$

- Minimize $-\mathbb{E}_{z\sim q(z)}\left[\log D\left(G(z)\right)\right]$ for **Generator** instead (keep Discriminator as it is)

Faces

Goodfellow, Ian, et al. **"Generative adversarial nets."** *Advances in neural information processing systems*. 2014.

CIFAR



Goodfellow, Ian, et al. **"Generative adversarial nets."** *Advances in neural information processing systems*. 2014.

Advantages of GANs

- Earlier work on Deep Generative models –
  - Boltzmann Machine
  - Deep Belief Nets
  - Variational AutoEncoders (VAE)

- GANs are more powerful and simple to implement –
  - Generation of real-like images is straightforward.
  - Training does not involve Maximum Likelihood Estimation.
  - Robust to Overfitting since Generator never sees the training data.
  - Empirically, GANs are good at capturing the modes of the distribution.

Challenges with GANs

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

- It is not easy to train GANs.
  - DC-GAN - One of the first architectures that could train GANs to generate good images.

Challenges with GAN Training

- Non-convergence: the model parameters oscillate, destabilize and never converge,
- Mode collapse: the generator collapses which produces limited varieties of samples,
- Diminished gradient: the discriminator gets too successful that the generator gradient vanishes and learns nothing,
- Highly sensitive to the hyperparameter selections.

Non-Convergence

- GANs involve two players

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

  - Discriminator's goal - maximize its reward.
  - Generator's goal - minimize Discriminator's reward.

- In such a minimax problem, Nash equilibrium of the game is required.
  - Problem: We might not converge to the Nash equilibrium at all, as SGD is not designed to provide convergence to Nash equilibrium point (solution).

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 $V(x, y) = xy$

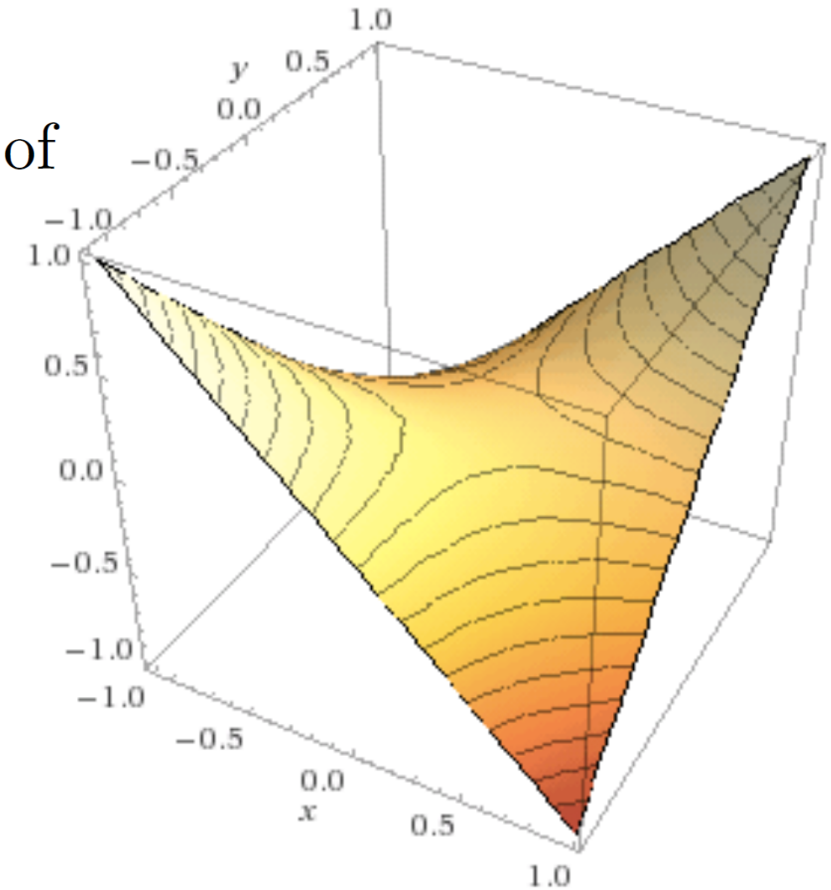
• State 1:	x > 0	y > 0	V > 0	Increase y	Decrease x
• State 2:	x < 0	y > 0	V < 0	Decrease y	Decrease x
• State 3:	x < 0	y < 0	V > 0	Decrease y	Increase x
• State 4 :	x > 0	y < 0	V < 0	Increase y	Increase x
• State 5:	x > 0	y > 0	V > 0	Increase y	Decrease x

== State 1

Non-Convergence

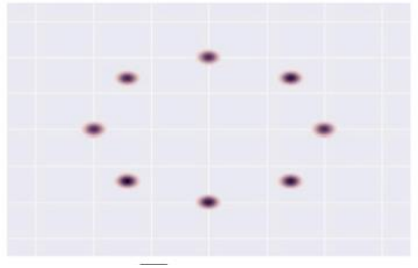
This is the canonical example of a saddle point.

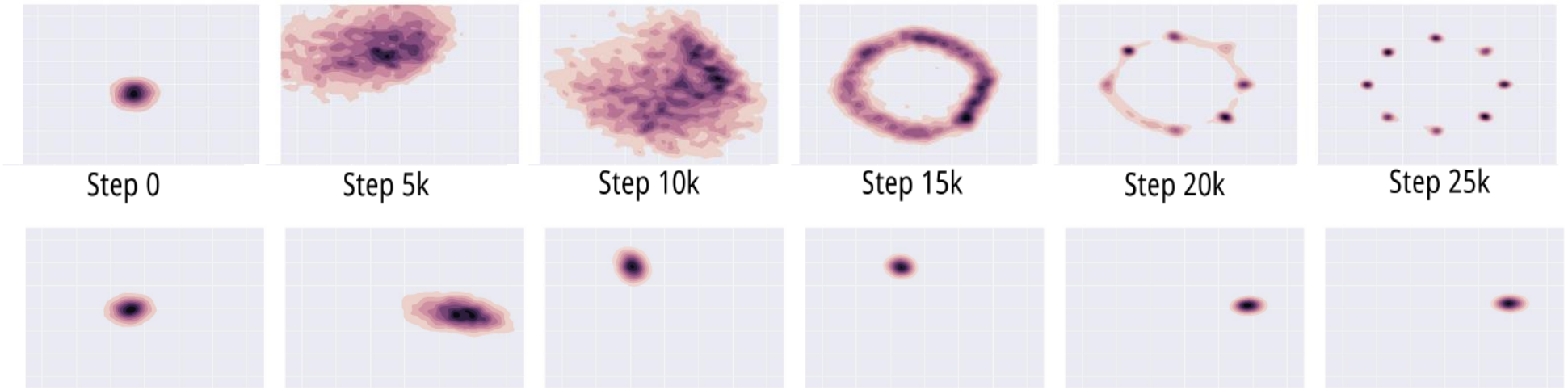
There is an equilibrium, at $x = 0,\ y = 0$.



Mode Collapse

- During training, the generator collapses to produce only a single sample or a small family of very similar samples.





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

Mode Collapse Causes Low Output Diversity

this small bird has a pink breast and crown, and black primaries and secondaries.

this magnificent fellow is almost all black with a red crest, and white cheek patch.

the flower has petals that are bright pinkish purple with white stigma

this white and yellow flower have thin white petals and a round yellow stamen

(Reed et al 2016)

Mode Collapse

(Reed et al, submitted to ICLR 2017)

Some Solutions

- Mini-Batch GANs
- Supervision with labels

- Some attempts :-
  - Unrolled GANs
  - W-GANs
  - DC-GANs
  - StyleGAN
  - StarGAN

Some Basic Solutions

- 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.

Some Basic Solutions

- 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

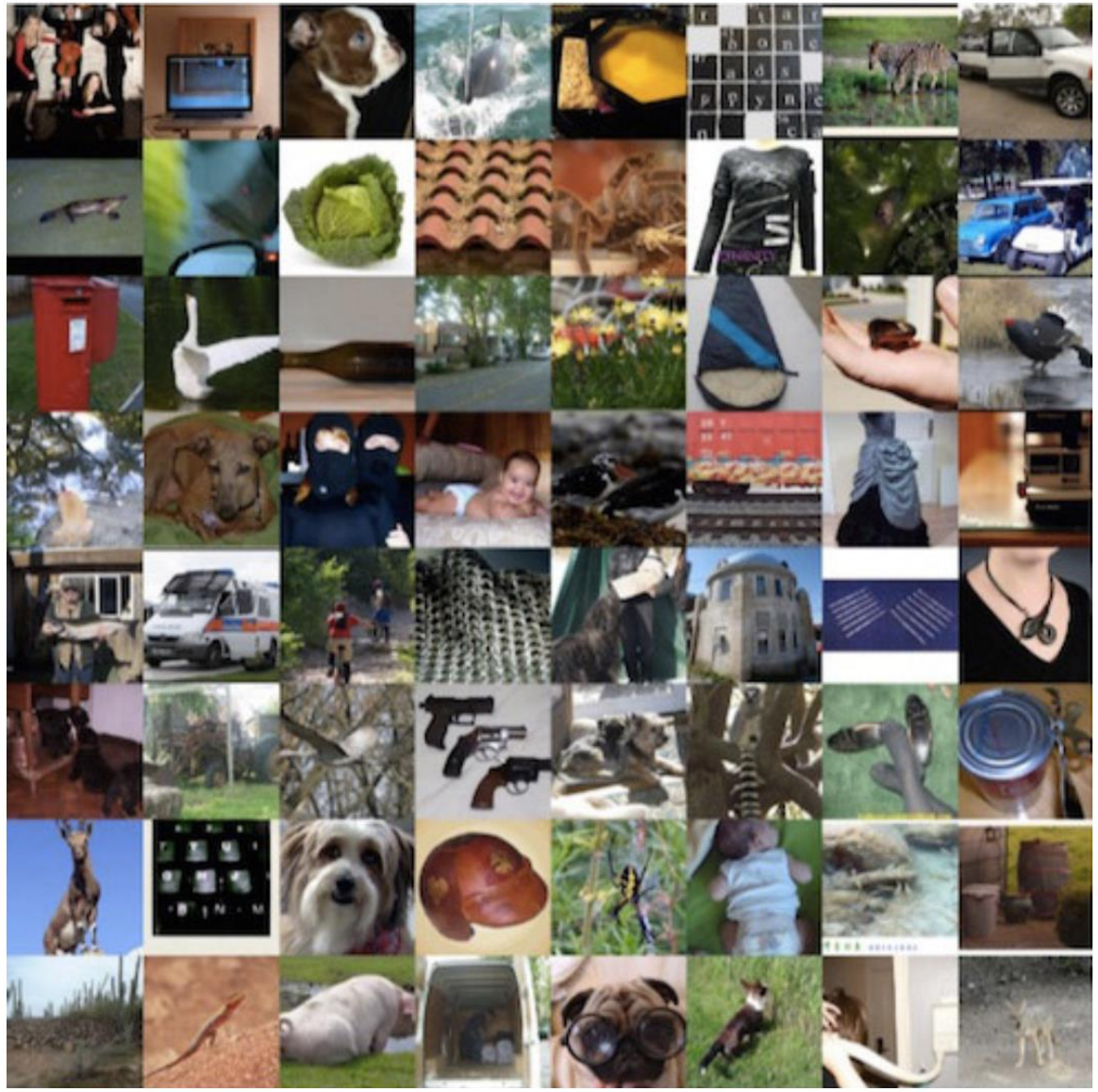
- Thus,
- Generator will be forced to produce diverse samples.

Mini-Batch GANs

- Extract features that capture diversity in the mini-batch
  - Example. L2 norm of difference between all pairs of samples 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.

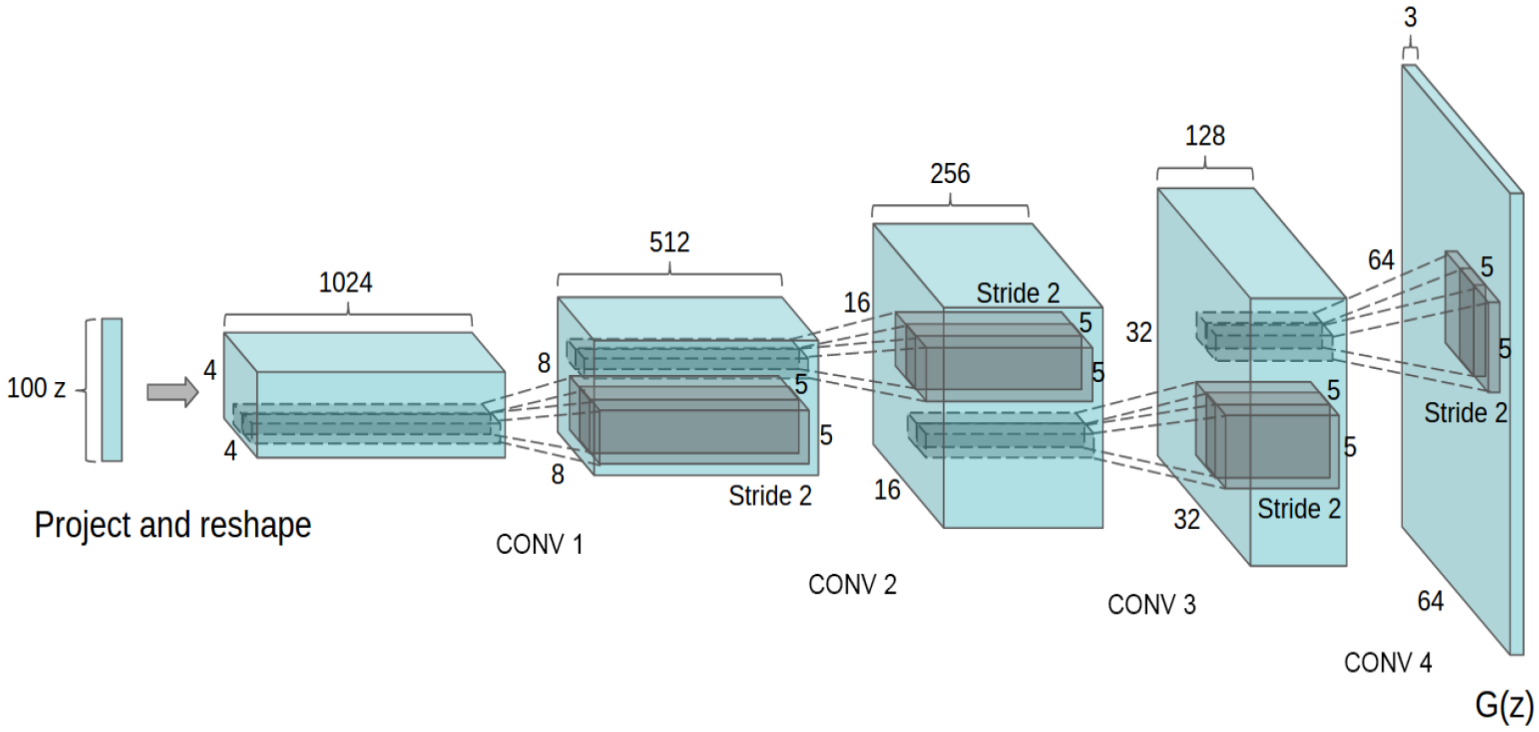
Minibatch GAN on ImageNet



(Salimans et. al 2016)

Deep Convolutional GANs (DCGANs)

Generator Architecture



Key ideas:

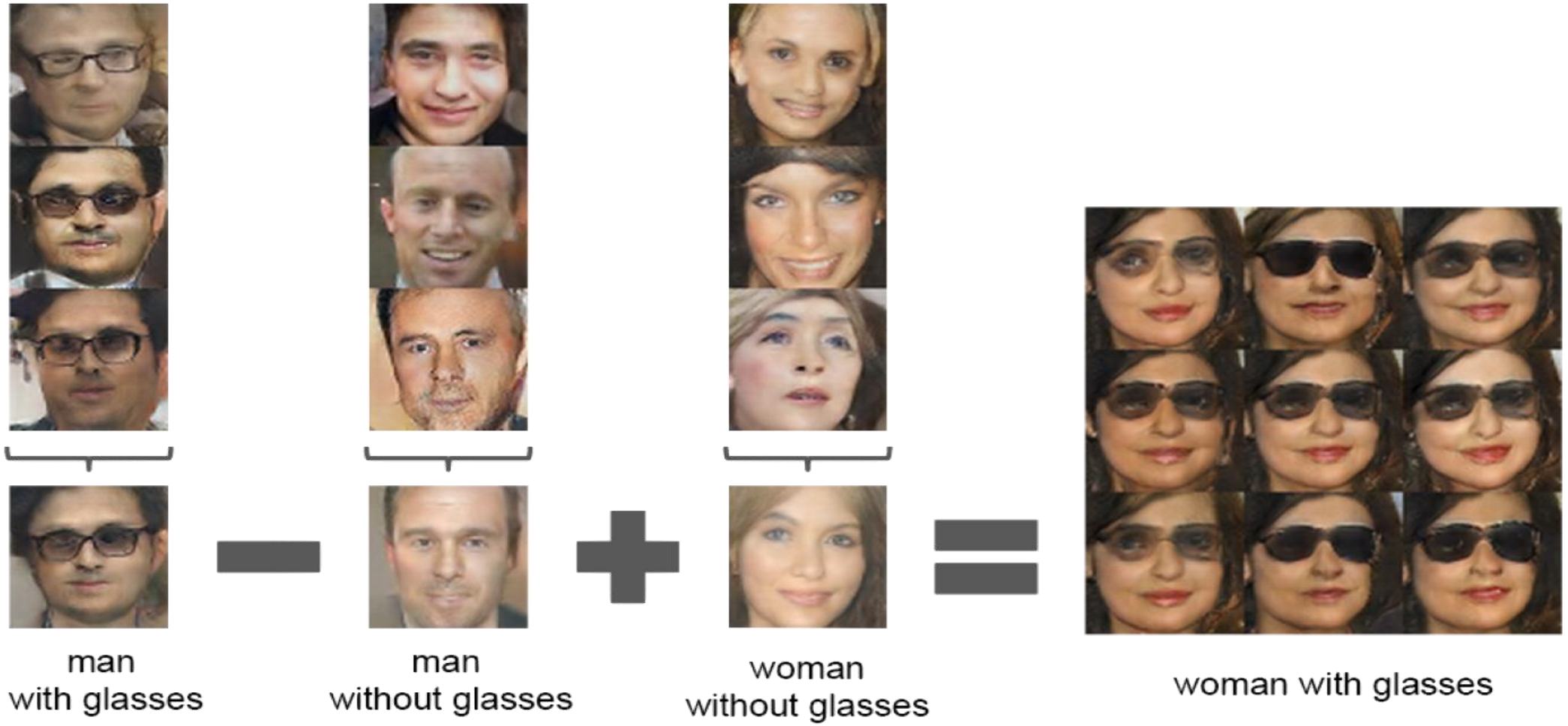
- Replace FC hidden layers with Convolutions
  - **Generator:** Fractional-Strided convolutions
- Use Batch Normalization after each layer
- **Inside Generator**
  - Use ReLU for hidden layers
  - Use Tanh for the output layer

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv:1511.06434 (2015).

DCGAN: Bedroom Image Generation

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv:1511.06434 (2015).

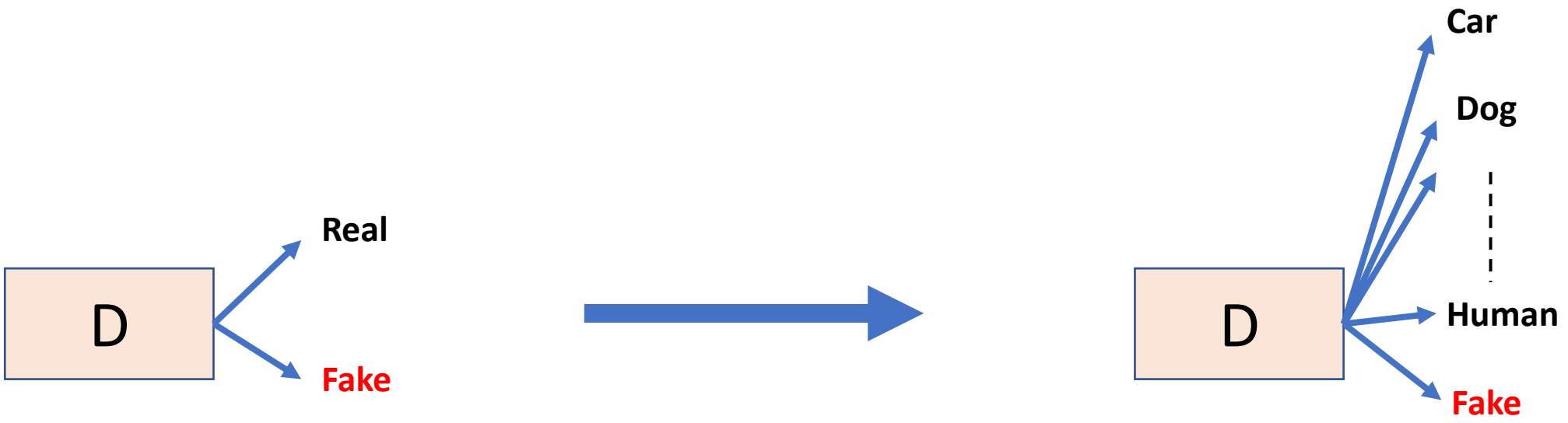
Latent Vectors



Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv:1511.06434 (2015).

Other Heuristic Solutions: Supervision with Labels

- Label information of the real data to be given to discriminator.



- Empirically generates much better samples

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

Alternate view of GANs

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

$$V(D,G) = \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim q(z)}\left[\log\left(1 - D(G(z))\right)\right]$$

$$D^* = \arg\max_D V(D,G) \qquad G^* = \arg\min_G V(D,G)$$

- In this formulation, Discriminator's strategy was $D(x) \to 1, \quad D\left(G(z)\right) \to 0$
- Alternatively, we can flip the binary classification labels i.e. **Fake = 1, Real = 0**

$$V(D,G) = \mathbb{E}_{x\sim p(x)}\left[\log\left(1 - D(x)\right)\right] + \mathbb{E}_{z\sim q(z)}\left[\log\left(D(G(z))\right)\right]$$

- In this new formulation, Discriminator's strategy will be $D(x) \to 0, \quad D\left(G(z)\right) \to 1$

Zhao, Junbo, Michael Mathieu, and Yann LeCun. **"Energy-based generative adversarial network."** arXiv preprint arXiv:1609.03126 (2016)

Alternate view of GANs (Contd.)

- If all we want to encode is $D(x) \rightarrow 0, \quad D\big(G(z)\big) \rightarrow 1$

$$D^* = argmax_D \, \mathbb{E}_{x \sim p(x)}\big[\log\big(1 - D(x)\big)\big] + \mathbb{E}_{z \sim q(z)}\big[\log\big(D(G(z))\big)\big]$$

We can use the following form

$$D^* = argmin_D \, \mathbb{E}_{x \sim p(x)} \log\big(D(x)\big) + \mathbb{E}_{z \sim q(z)} \left[\log\left(1 - D\big(G(z)\big)\right)\right]$$

Zhao, Junbo, Michael Mathieu, and Yann LeCun. **"Energy-based generative adversarial network."** arXiv preprint arXiv:1609.03126 (2016)

Alternate view of GANs (Contd.)

Zhao et al (2016) : Replace the cross-entropy with any loss function **(**Hinge Loss) for the discriminator.

$$D^* = argmin_D \, \mathbb{E}_{x \sim p(x)} D(x) + \mathbb{E}_{z \sim q(z)} \max\Big(0, m - D\big(G(z)\big)\Big)$$

- Here m is the positive margin of classifier.

- And loss function for the Generator is $D\big(G(z)\big)$.
- Minimizing generator's loss is equivalent to maximizing the second term in *D**.
- Thus the discriminator is required to output -
  - High values for fake samples
  - Low values for real samples

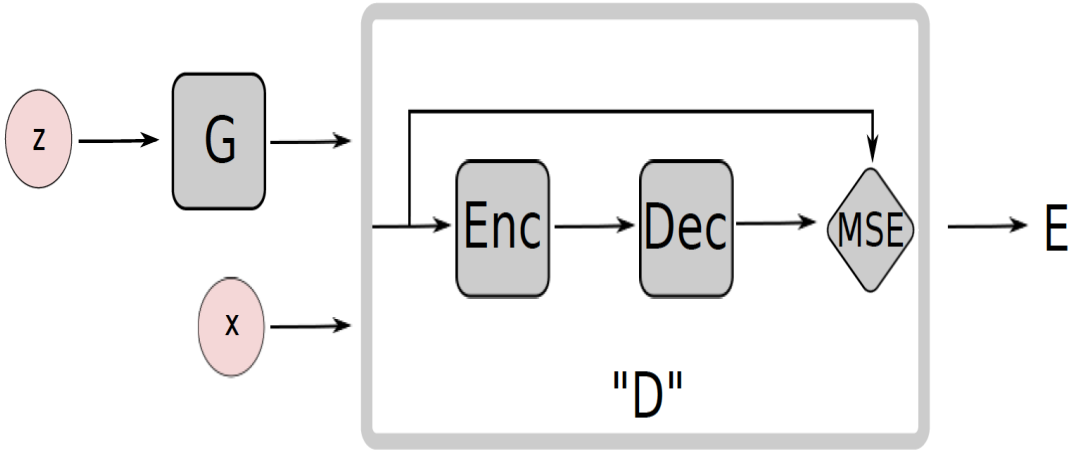
Zhao, Junbo, Michael Mathieu, and Yann LeCun. **"Energy-based generative adversarial network."** arXiv preprint arXiv:1609.03126 (2016)

Energy-Based GANs

- Energy Based GANs
  - **Generator** will try to generate samples with low values.
  - **Discriminator** will try to assign high scores to fake values.

$$D(x) = ||Dec\big(Enc(x)\big) - x||_{MSE}$$

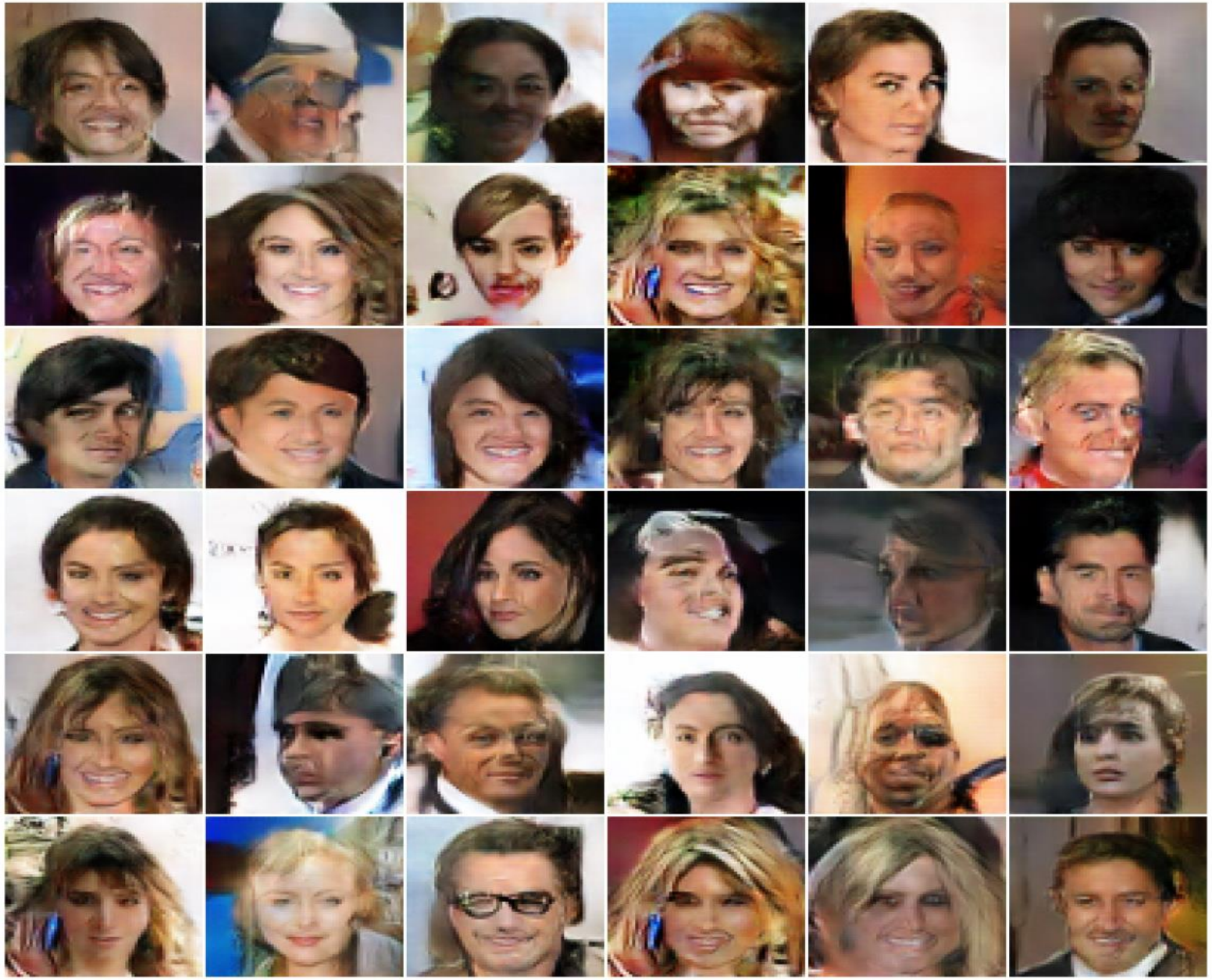
- **Use AutoEncoder inside the Discriminator**



- **Use Mean-Squared Reconstruction error as $D(x)$**
  - High Reconstruction Error for Fake samples
  - Low Reconstruction Error for Real samples

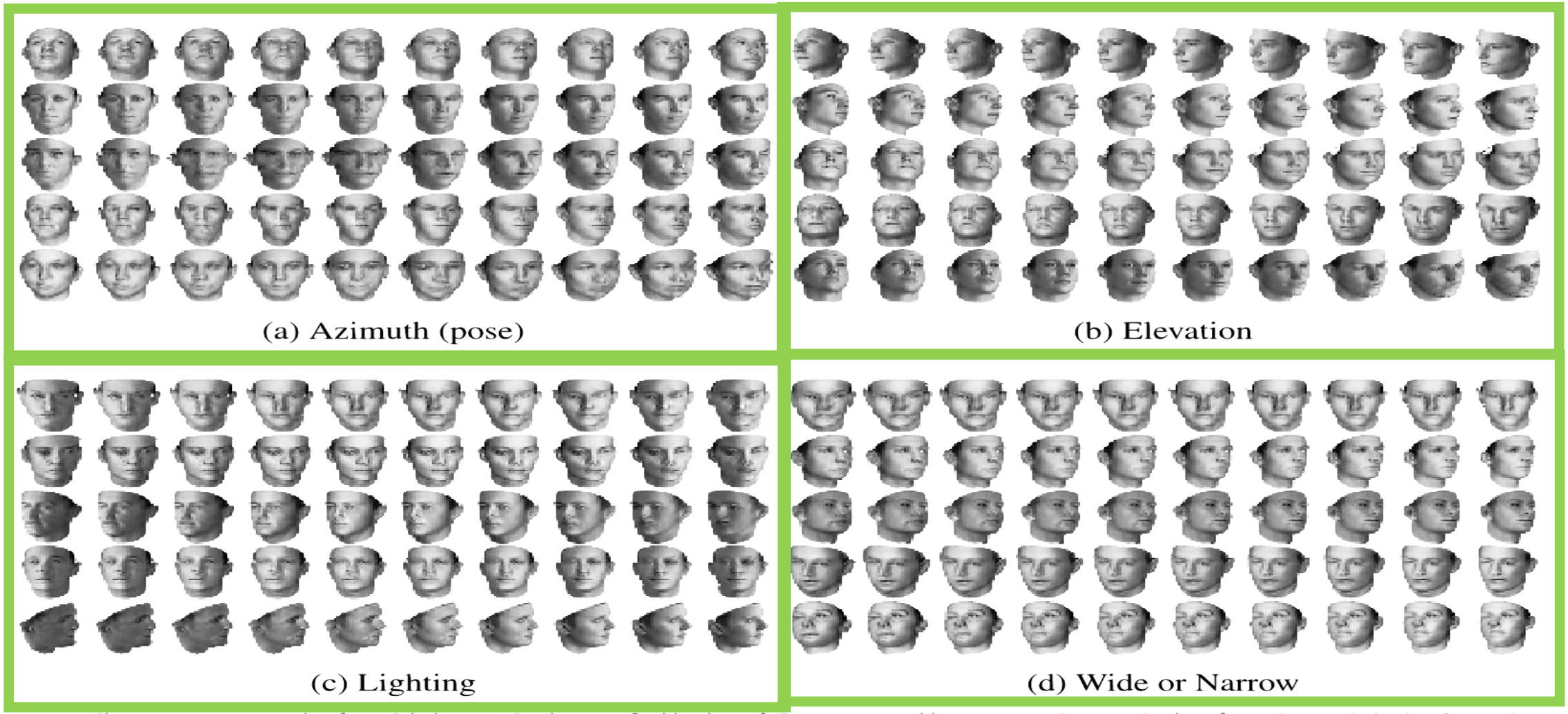
Zhao, Junbo, Michael Mathieu, and Yann LeCun. **"Energy-based generative adversarial network."** arXiv preprint arXiv:1609.03126 (2016)

Celebrity Image Generation



Generating Different Elevation

3D Faces



Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets, NIPS (2016).

Generating Different Elevation

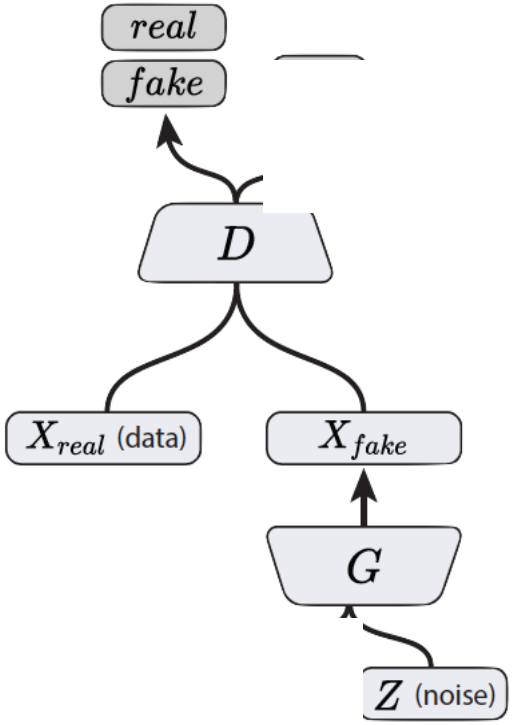
3D Chairs

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets, NIPS (2016).

How to reward Disentanglement?

- Disentanglement means individual dimensions independently capturing key attributes of the image
- **Let's partition the noise vector into 2 parts :-**
  - **z** vector will capture slight variations in the image
  - **c** vector will capture the main attributes of the image
  - For e.g. **Digit, Angle** and **Thickness** of images in MNIST
- If **c** vector captures the key variations in the image,

  Will c and x_{fake} be highly correlated or weakly correlated?



InfoGAN
(Chen, et al., 2016)

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets

Mutual Information

- Mutual Information captures the mutual dependence between two variables.
- Mutual information between two variables X, Y is defined as:

$$I(X;Y) = \sum_{x,y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

$$I(X;Y) = H(X) - H(X|Y) = \mathrm{H(Y)} - \mathrm{H(Y|X)}$$

  - Here H(X) and H(Y) are marginal entropies and H(X|Y) and H(Y|X) are conditional entropies.

Entropy

- Entropy of a random variable is defined as the average level of the amount of information, uncertainty or surprise it contains.
- Given a random variable X with possible outcomes $\{ x_1, x_2, \ldots, x_n \}$, and the corresponding probability of their occurrence as $\{ p(x_1), p(x_2), \ldots, p(x_n) \}$, its entropy is defined as

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

- Conditional Entropy of two random variables is defined as

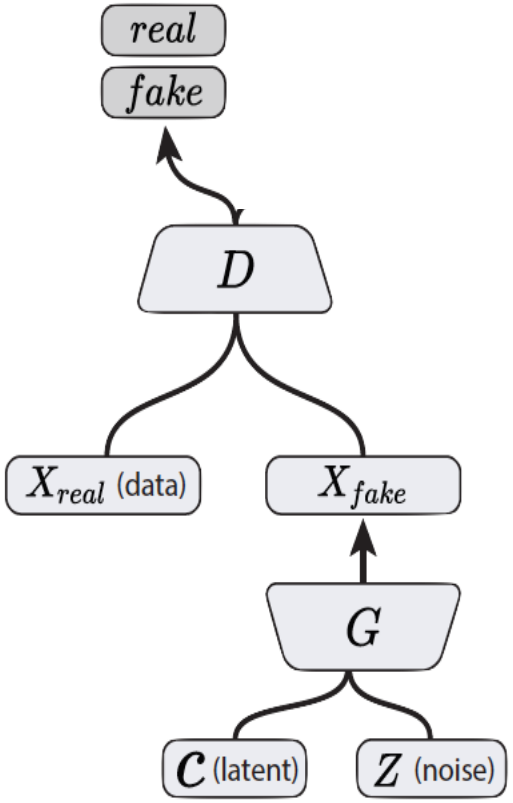
$$H(X|Y) = -\sum_{i=1}^{n} p(x_i, y_i) \log \frac{p(x_i, y_i)}{p(y_i)}$$

InfoGAN

- We want to maximize the mutual information I between $\boldsymbol{c}$ and $\mathbf{x} = \boldsymbol{G}(\boldsymbol{z}, \boldsymbol{c})$

- Incorporate in the value function of the minimax game.

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda\, I\big(c; G(z,c)\big)$$

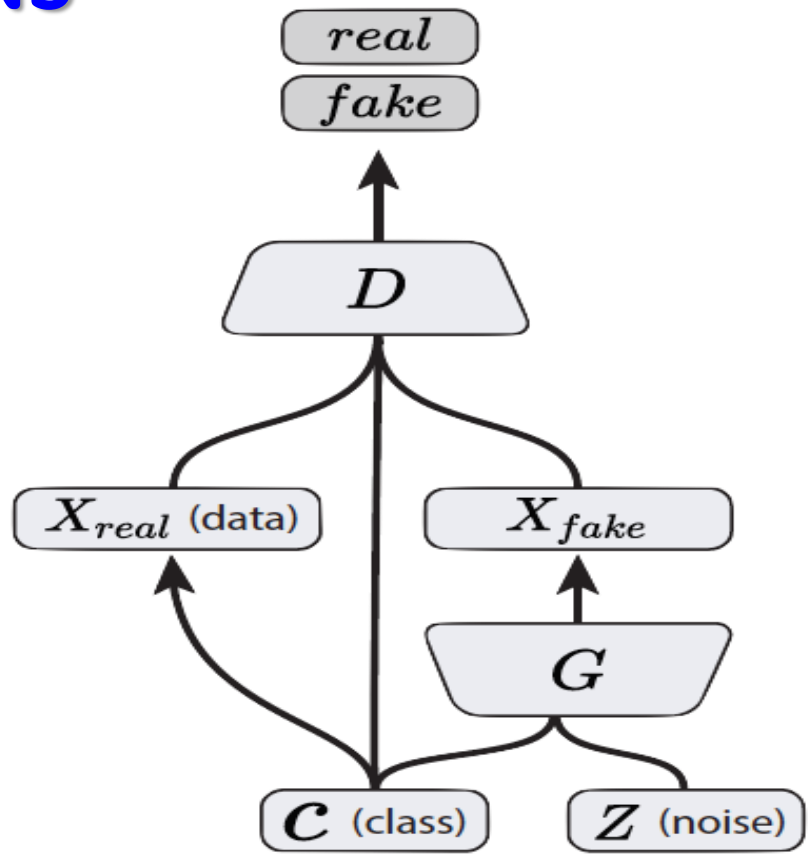


Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets, NIPS (2016).

Conditional GANs

- Simple modification to the original GAN framework that conditions the model on *additional information* for better multi-modal learning.

- Many practical applications of GANs explicit *supervision* is available.



Conditional GAN
(Mirza & Osindero, 2014)

Mirza, Mehdi, and Simon Osindero. "**Conditional generative adversarial nets**". arXiv preprint arXiv:1411.1784 (2014).

Conditional GANs

MNIST digits generated conditioned on their class label.

$[1, 0, 0, 0, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 1, 0, 0, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 0, 1, 0, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 0, 0, 1, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 0, 0, 0, 1, 0, 0, 0, 0, 0] \rightarrow$

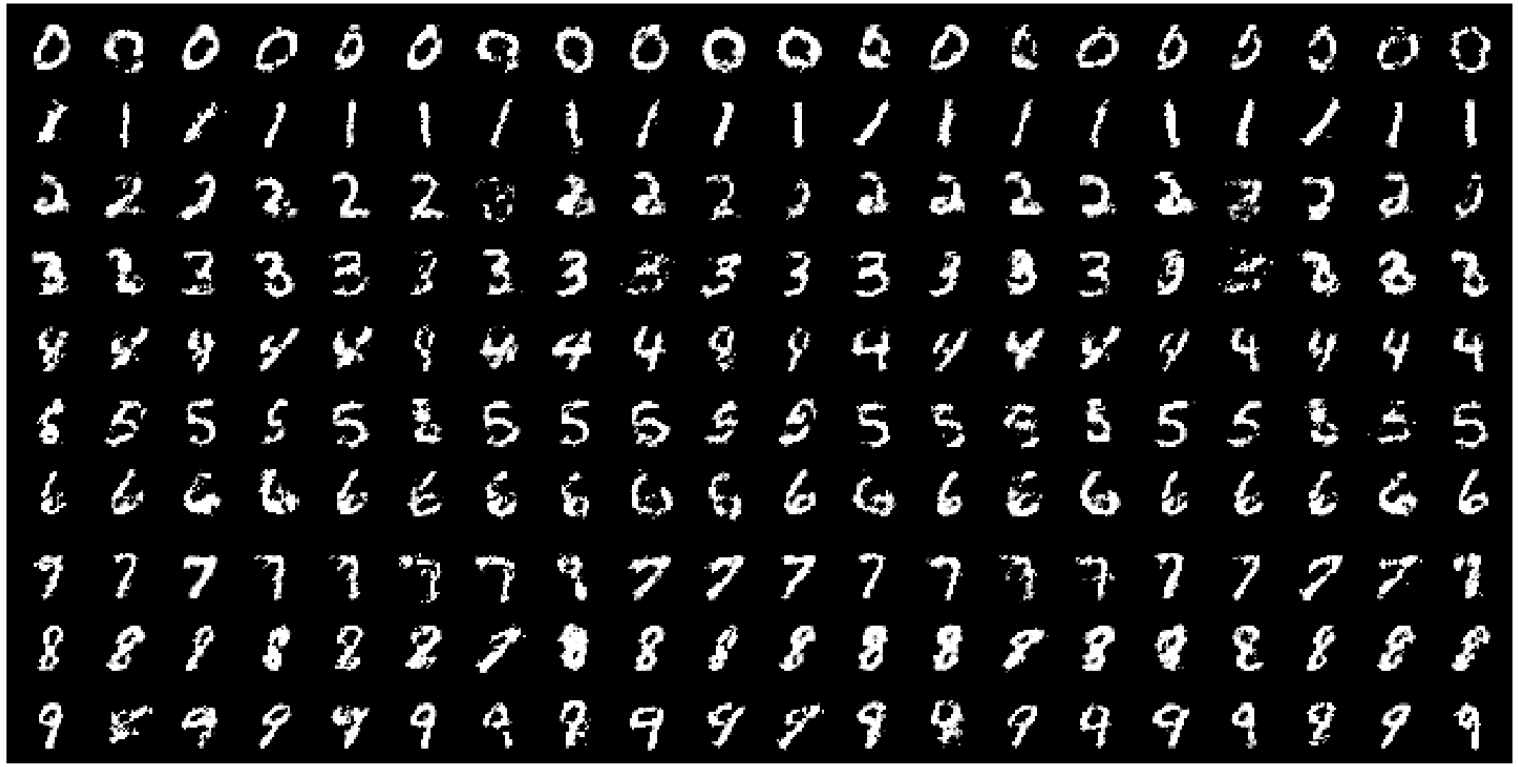
$[0, 0, 0, 0, 0, 1, 0, 0, 0, 0] \rightarrow$

$[0, 0, 0, 0, 0, 0, 1, 0, 0, 0] \rightarrow$

$[0, 0, 0, 0, 0, 0, 0, 1, 0, 0] \rightarrow$

$[0, 0, 0, 0, 0, 0, 0, 0, 1, 0] \rightarrow$

$[0, 0, 0, 0, 0, 0, 0, 0, 0, 1] \rightarrow$



Mirza, Mehdi, and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014).

Important Applications

- Image-to-Image Translation
- Text-to-Image Synthesis
- Face Aging

Image-to-Image Translation

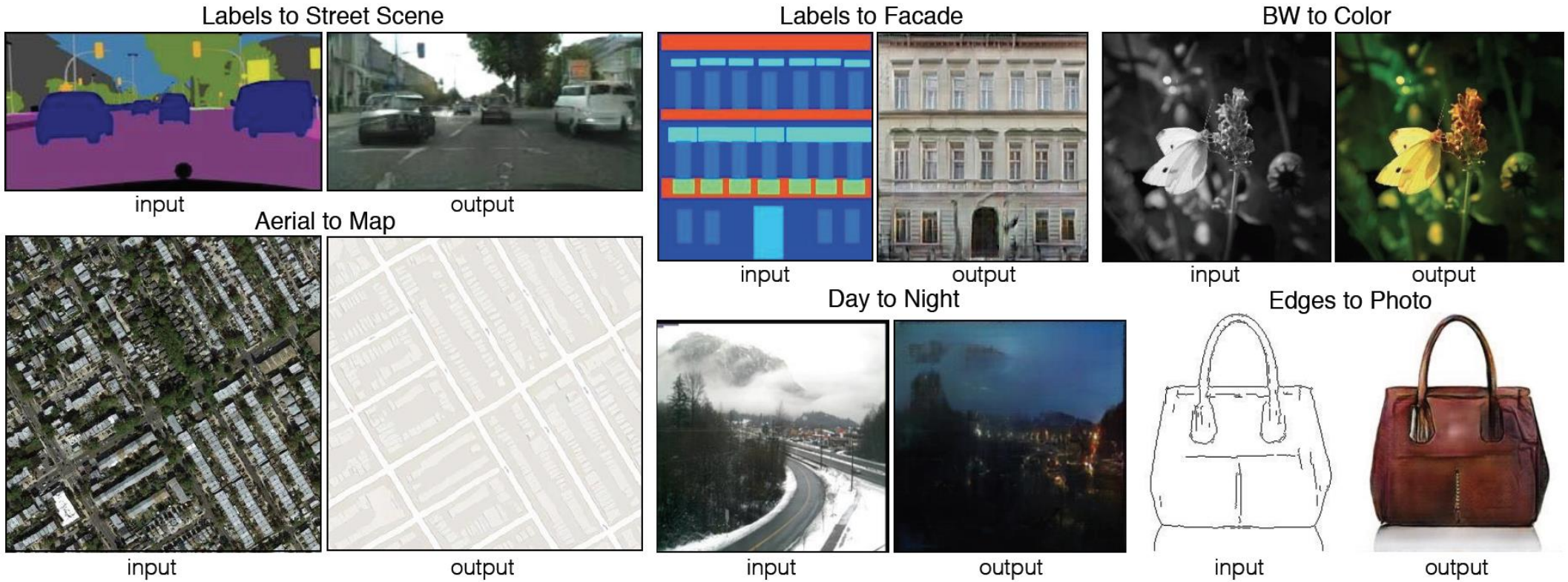


Figure 1 in the original paper.

[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).

Image-to-Image Translation

- Architecture: *DCGAN*-based architecture
- Training is conditioned on the images from the source domain.
- Conditional GANs provide an effective way to handle many complex domains without worrying about designing *structured loss* functions explicitly.

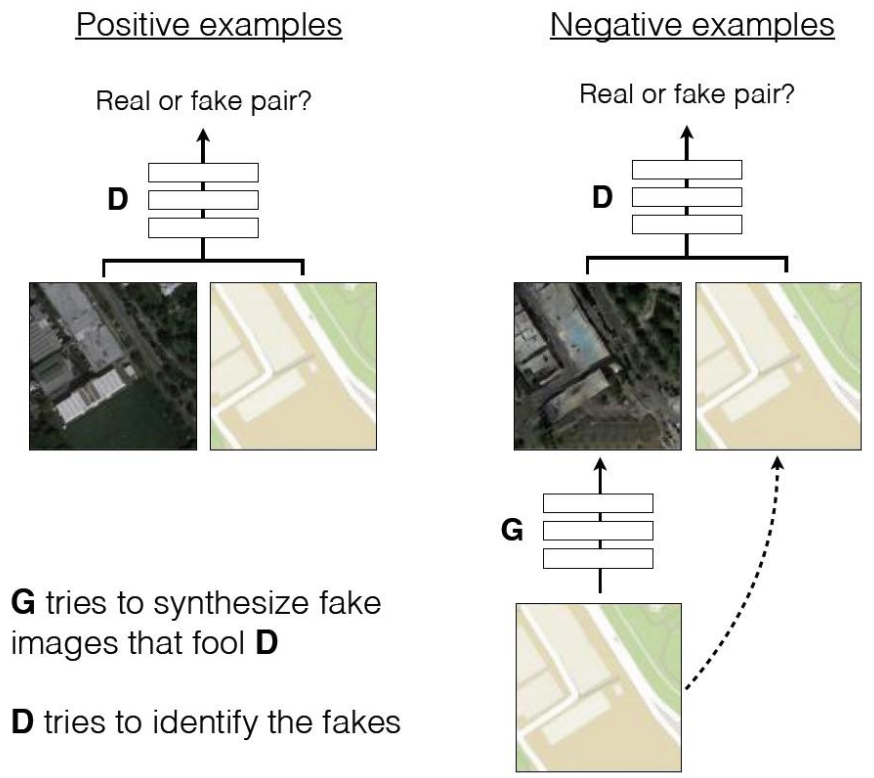


Figure 2 in the original paper.

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 conditioned on “dense” text embedding.

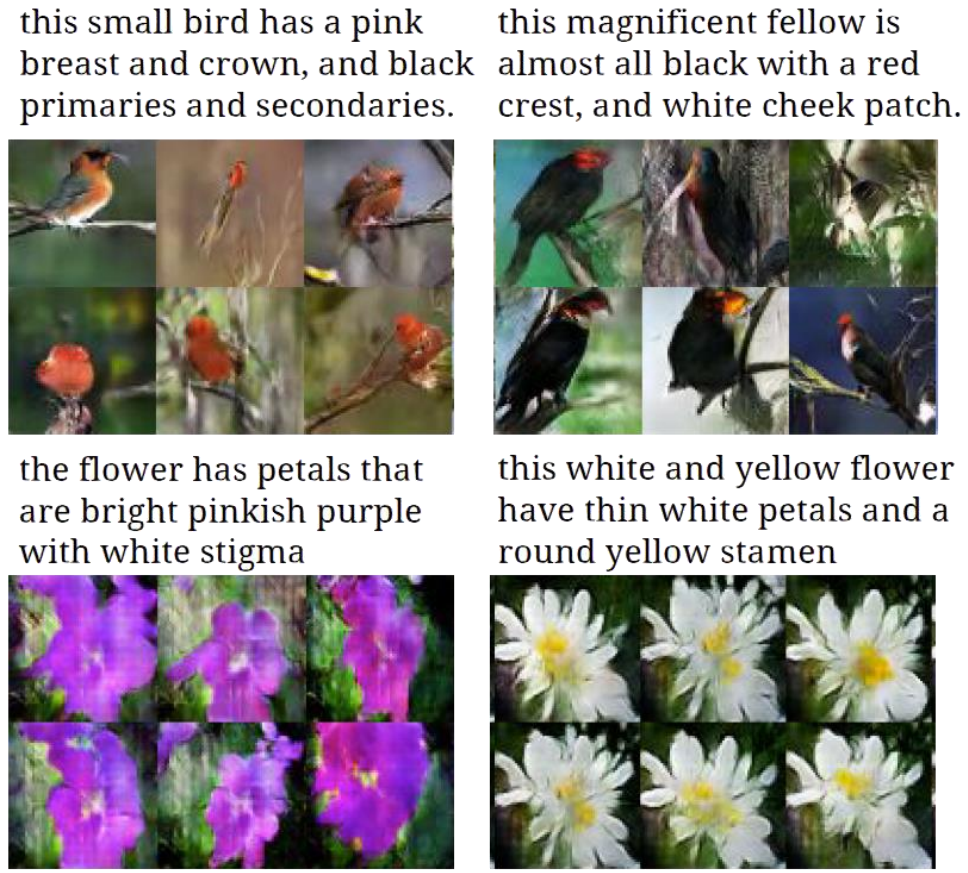


Figure 1 in the original paper.

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

Text-to-Image Synthesis

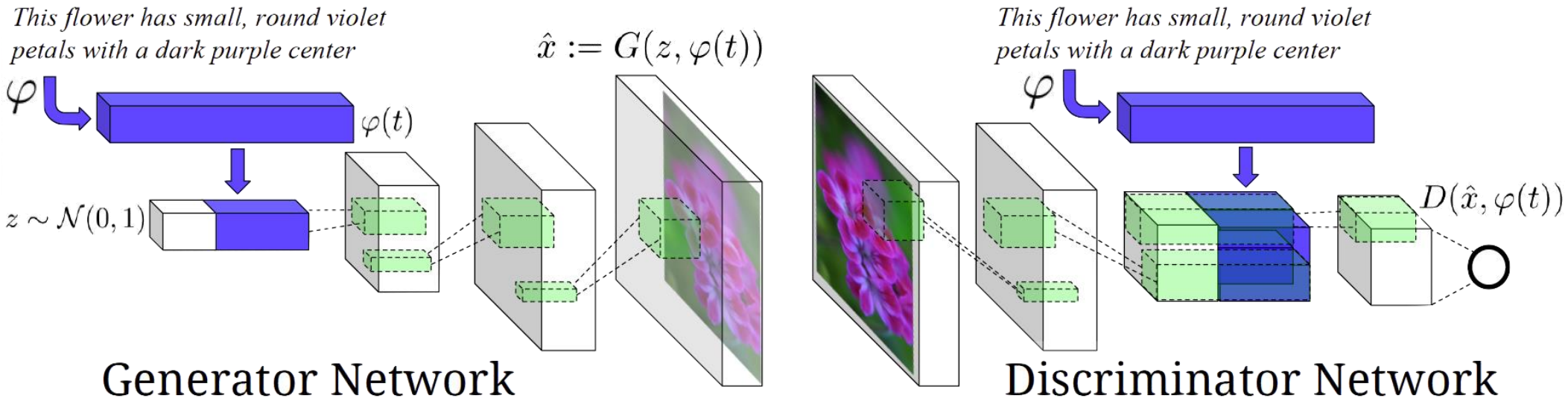


Figure 2 in the original paper.

Positive Example:

Real Image, Right Text

Negative Examples:

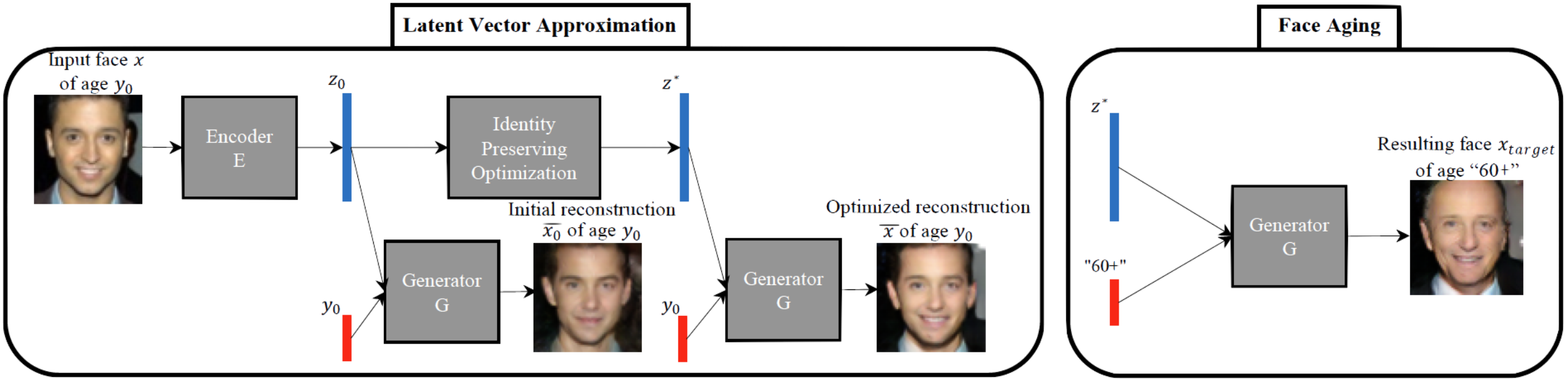
Real Image, Wrong Text

Fake Image, Right Text

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

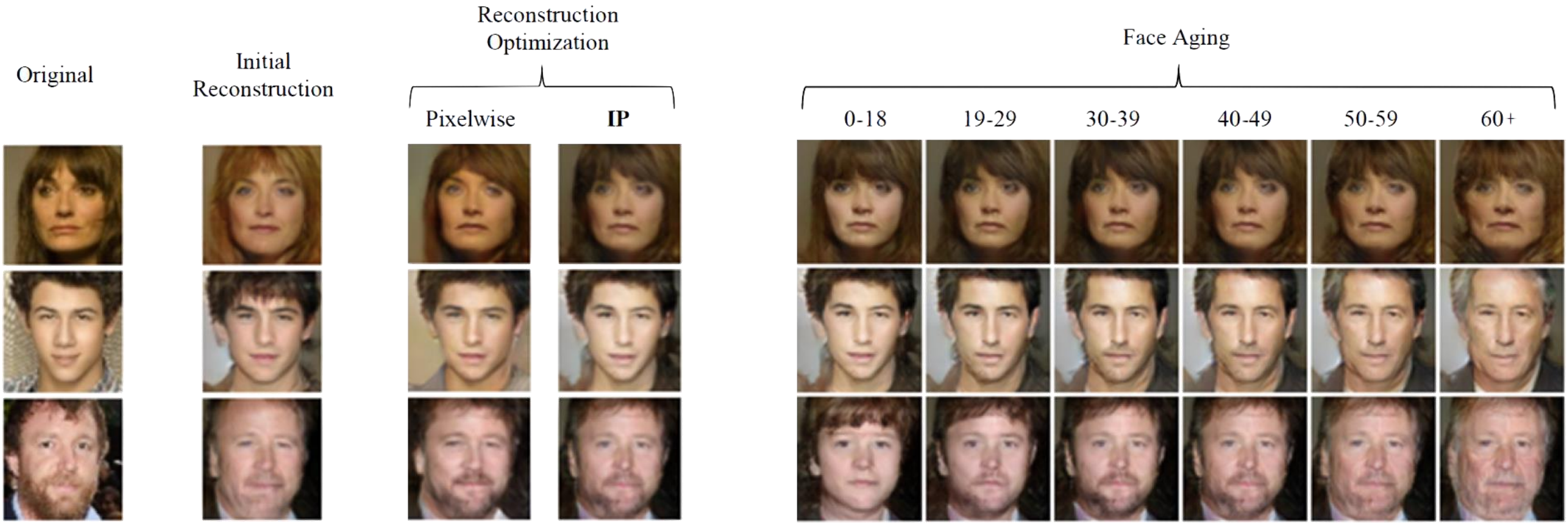
Face Aging with Conditional GANs

- 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.



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

Face Aging with Conditional GANs

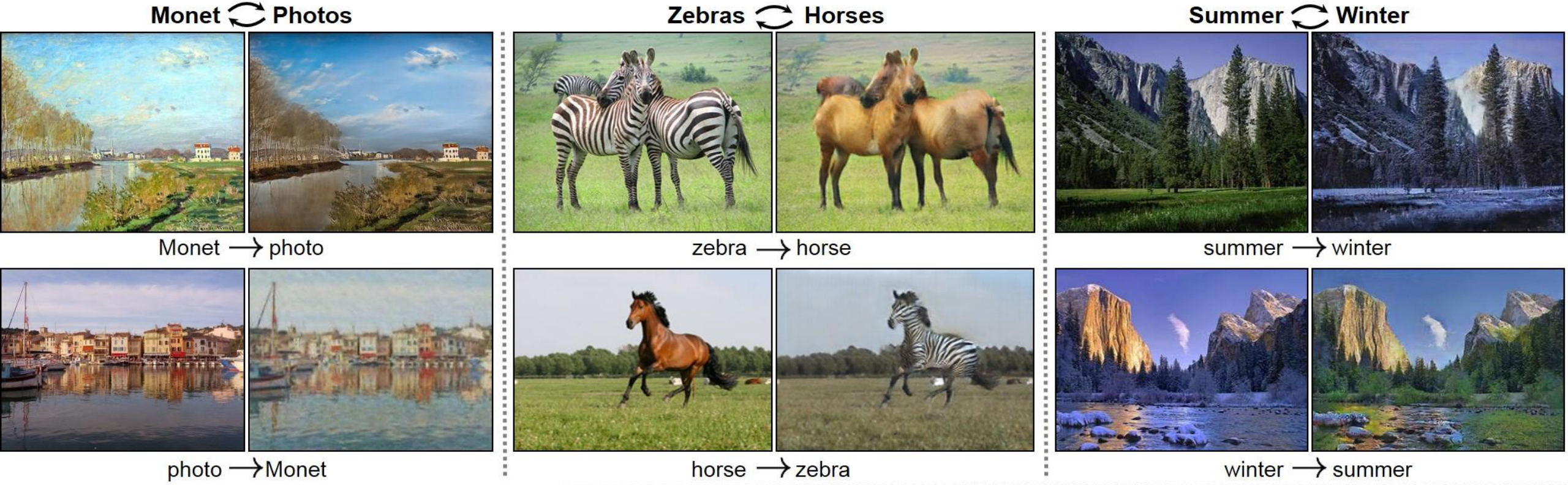


Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). "**Face Aging With Conditional Generative Adversarial Networks**". arXiv preprint arXiv:1702.01983.

Other Recent Developments

CycleGAN (Zhu et al., ICCV 2017)

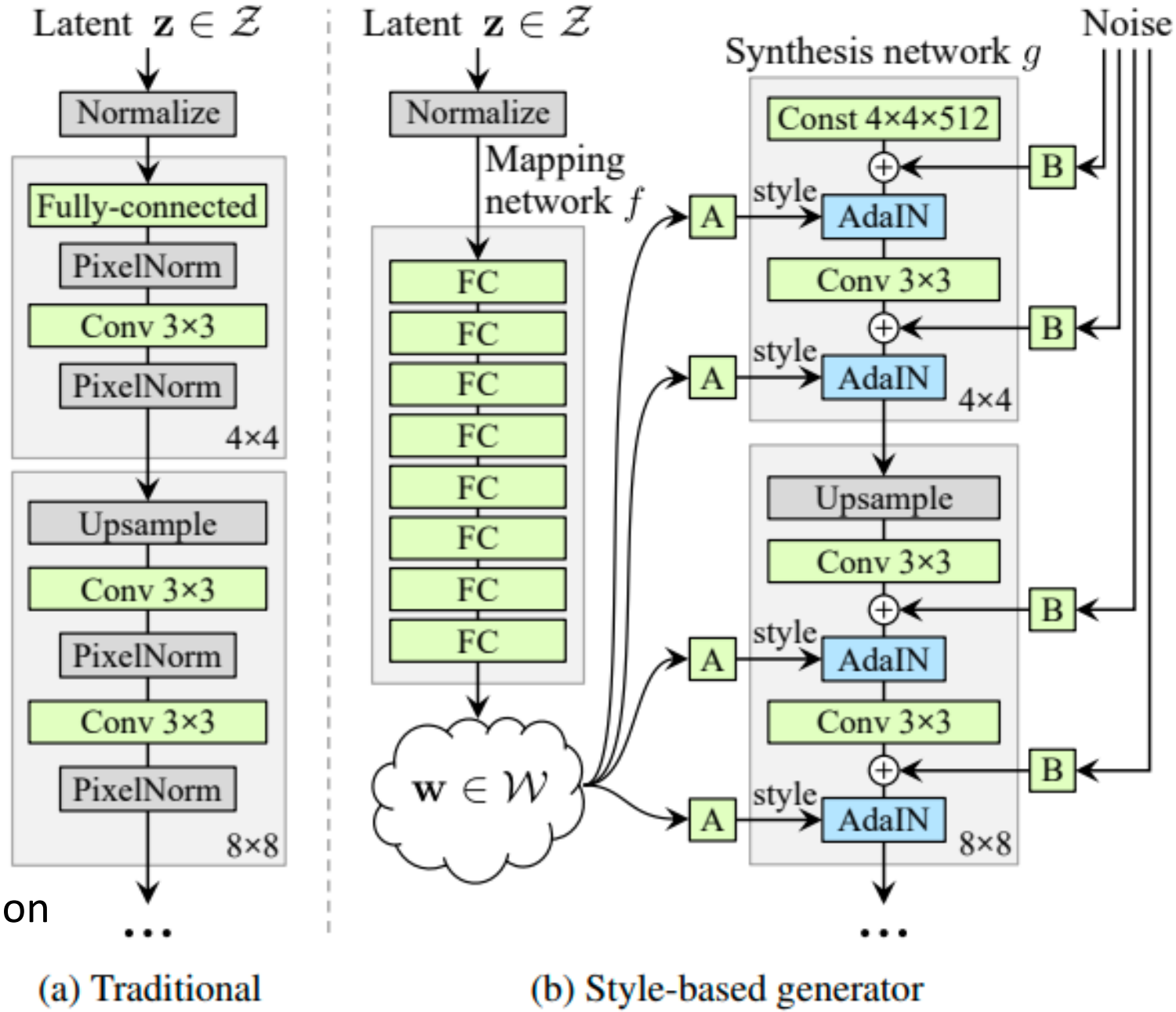
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks



StyleGAN (2018): A Popular GAN Model

- Developed by NVIDIA researchers in December 2018, code was made public in 2019.
- A demo

StyleGAN Generator Architecture



(a) Traditional

(b) Style-based generator

Image source: https://arxiv.org/abs/1406.2661

AdaIn:
adaptive instance normalization

StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation

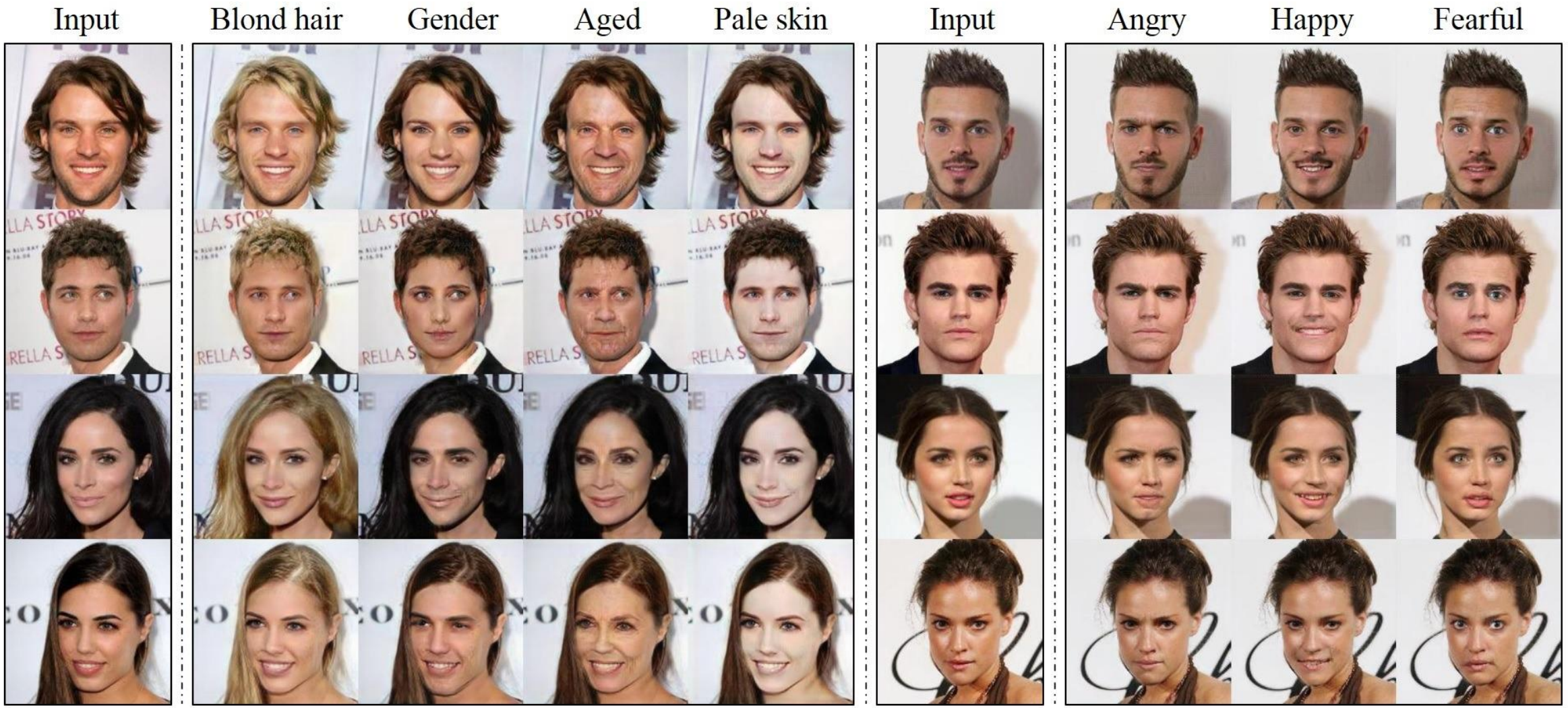
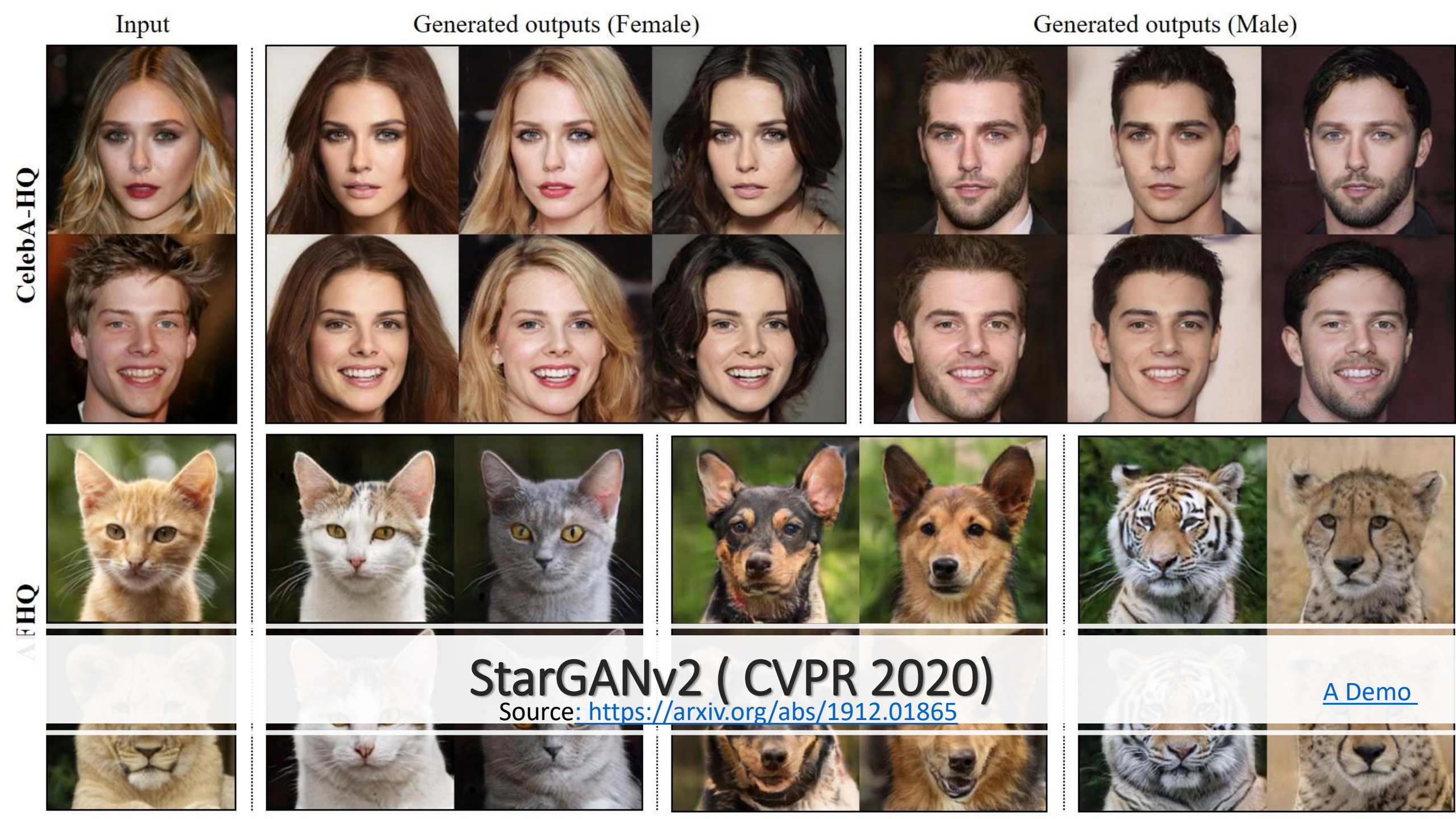


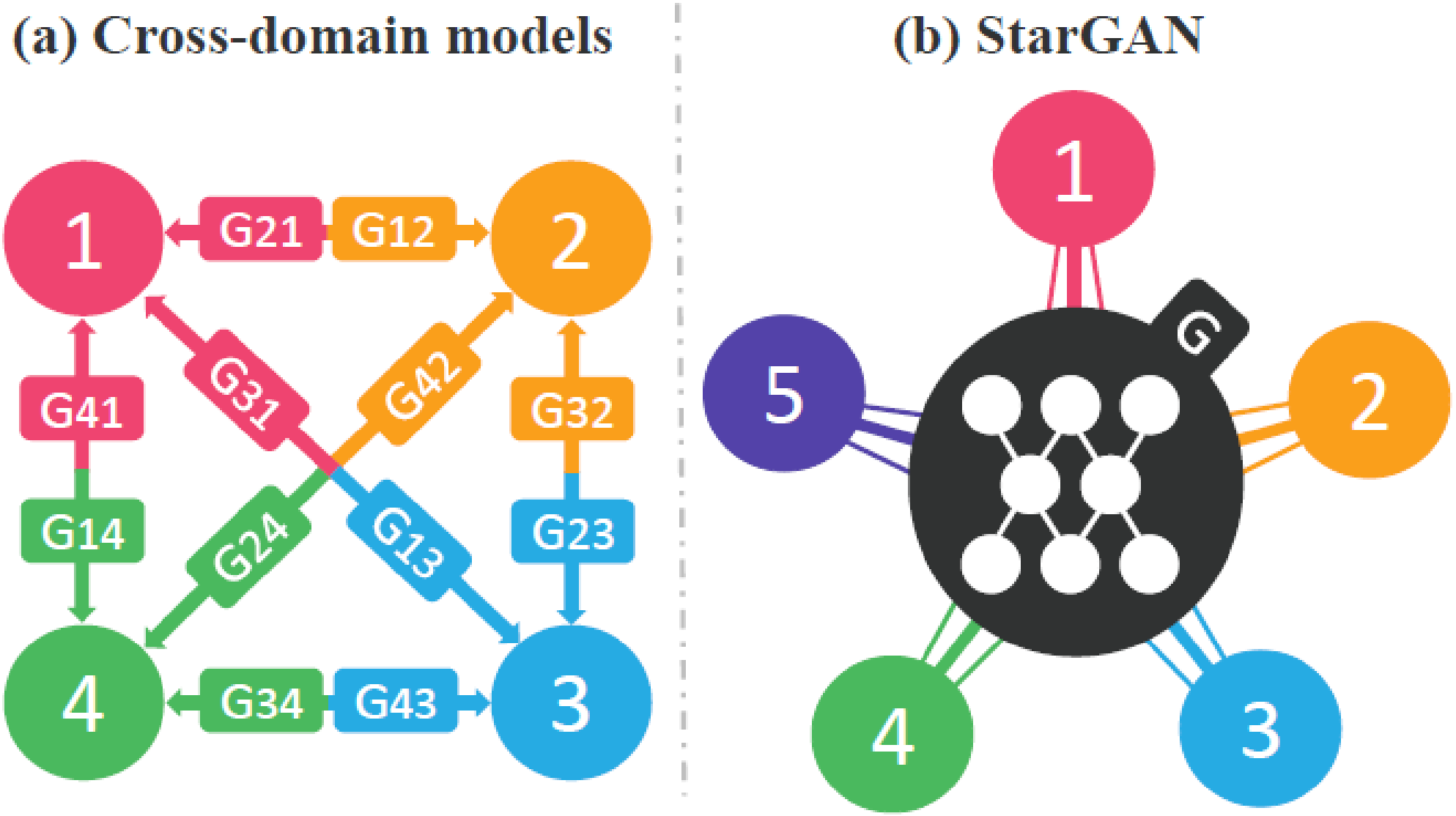
Image source: https://arxiv.org/abs/1711.09020



StarGANv2 (CVPR 2020)

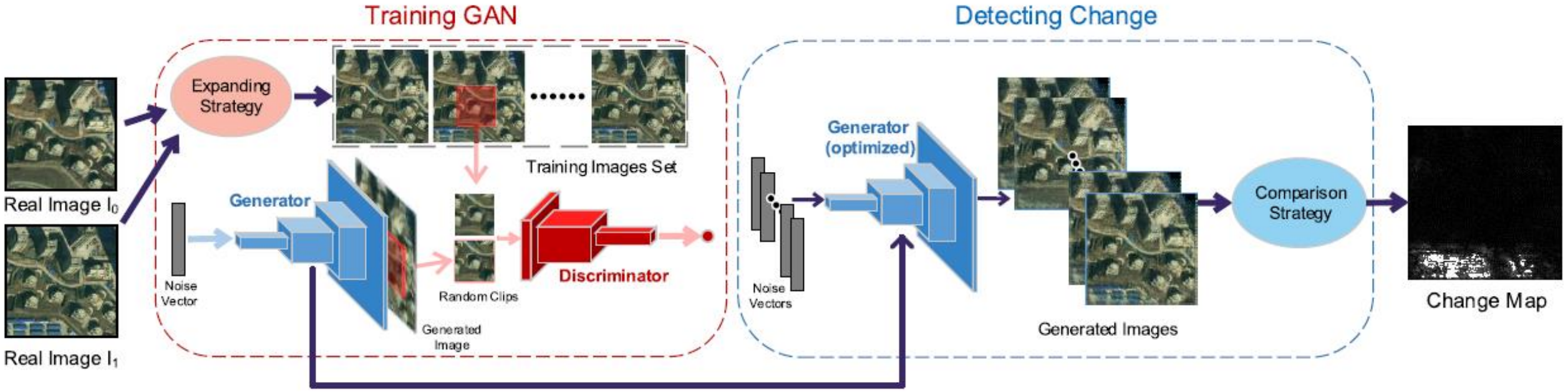
Source: https://arxiv.org/abs/1912.01865

A Demo



Source: https://arxiv.org/abs/1912.01865

GAN for Change Detection in Satellite Images



- CHANGE detection in Earth vision aims at generating the change map that localizes the changed area in two satellite images which were taken at different times.
- Applications include urbanization monitoring, natural disaster analyzing, detecting suspicious activities etc.

Caijun Ren et al., IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, 2021

Summary

- GANs are generative models that are implemented using two neural network modules: **Generator** and **Discriminator**.
- **Generator** tries to generate samples from random noise as input.
- **Discriminator** tries to distinguish the samples from Generator and samples from the real data distribution.
- Both networks are trained adversarially (in tandem) to fool the other component. In this process, both models become better at their respective tasks.
- Plethora of GAN models are proposed in recent years with a variety of different applications. Hot area of research for applications in different domains.

References

- Most of the slides are taken from the original lecture slides of Ian Goodfellow of GAN.
- 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).
- S. Lazebnik , Illinois University , Lecture slides
- 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.