Generative Adversarial Networks (GANs)

From **Ian Goodfellow et al.**

A short tutorial by :-

Binglin, Shashank & Bhargav

Outline

- **Part 1:** **Introduction to GANs**
- **Part 2:** **Some challenges with GANs**
- **Part 3:** **Applications of GANs**

Part 1

- **Motivation for Generative Models**
- **From Adversarial Training to GANs**
- **GAN’s Architecture**
- **GAN’s objective**
- **DCGANs**

GANs

- **Generative**
  - Learn a generative model
- **Adversarial**
  - Trained in an adversarial setting
- **Networks**
  - Use Deep Neural Networks

Why Generative Models?

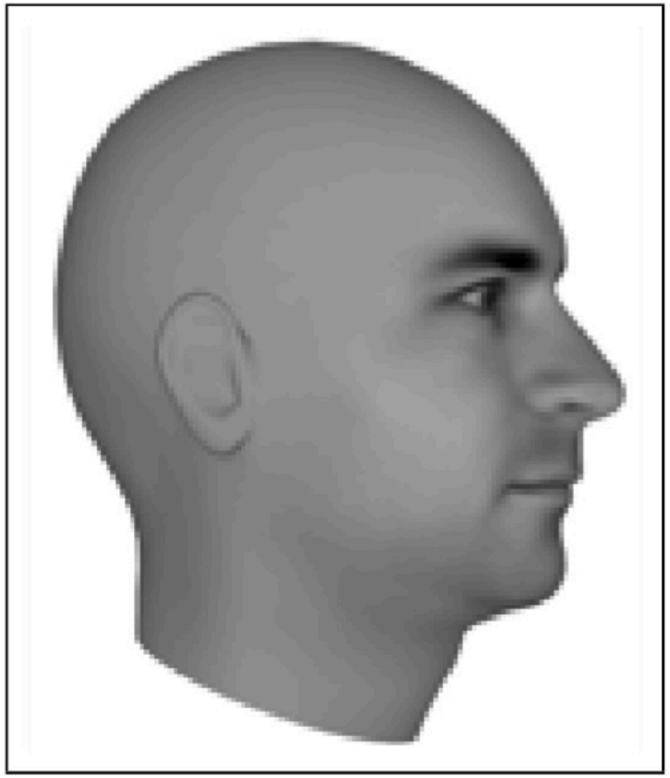
- **We've only seen discriminative models so far**
  - Given an image **X**, predict a label **Y**
  - Estimates **P(Y|X)**

- **Discriminative models have several key limitations**
  - Can't model **P(X)**, i.e. the probability of seeing a certain image
  - Thus, can't sample from **P(X)**, i.e. **can't generate new images**

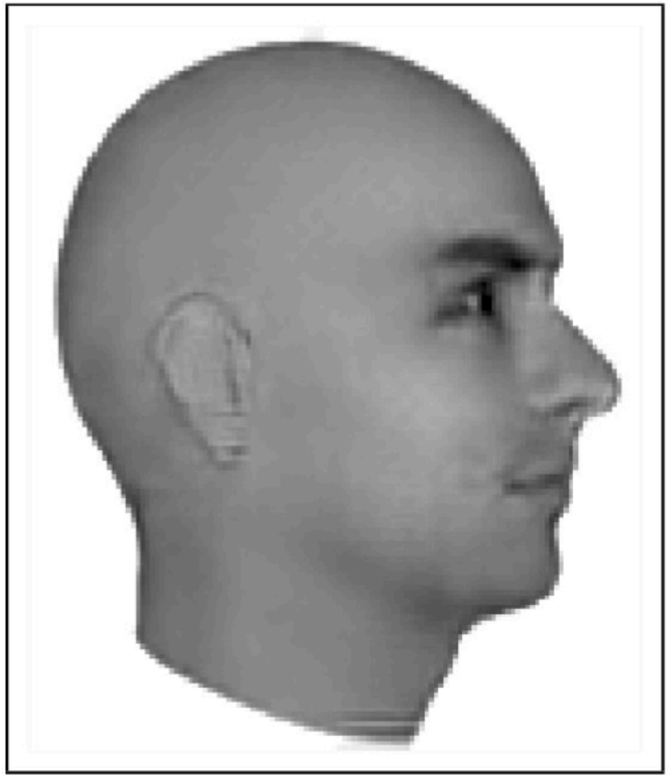
- **Generative models (in general) cope with all of above**
  - Can model **P(X)**
  - Can generate new images

Magic of GANs…

Ground Truth



Adversarial



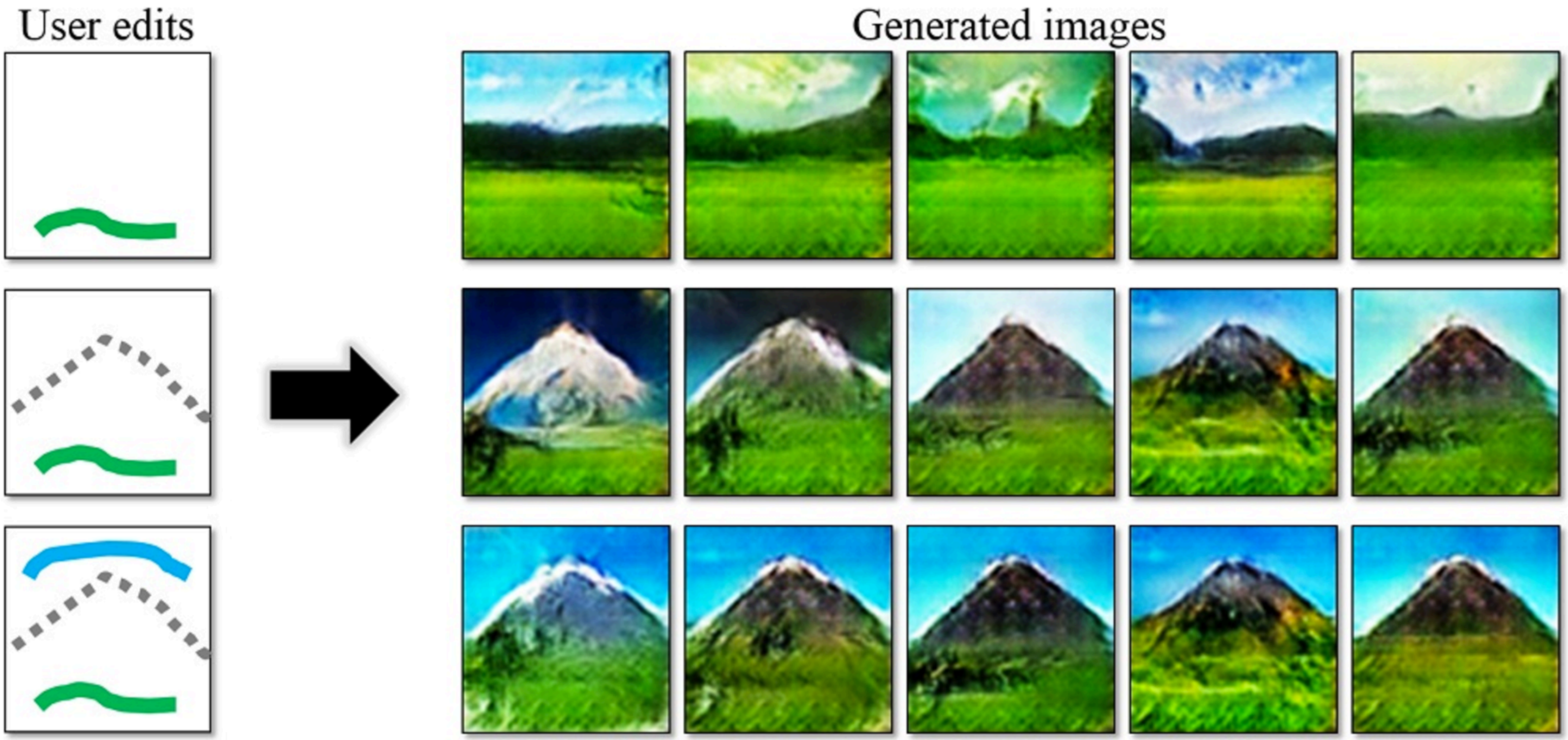
Lotter, William, Gabriel Kreiman, and David Cox. "Unsupervised learning of visual structure using predictive generative networks." *arXiv preprint arXiv:1511.06380* (2015).

Magic of GANs...

Which one is Computer generated?

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

Magic of GANs…



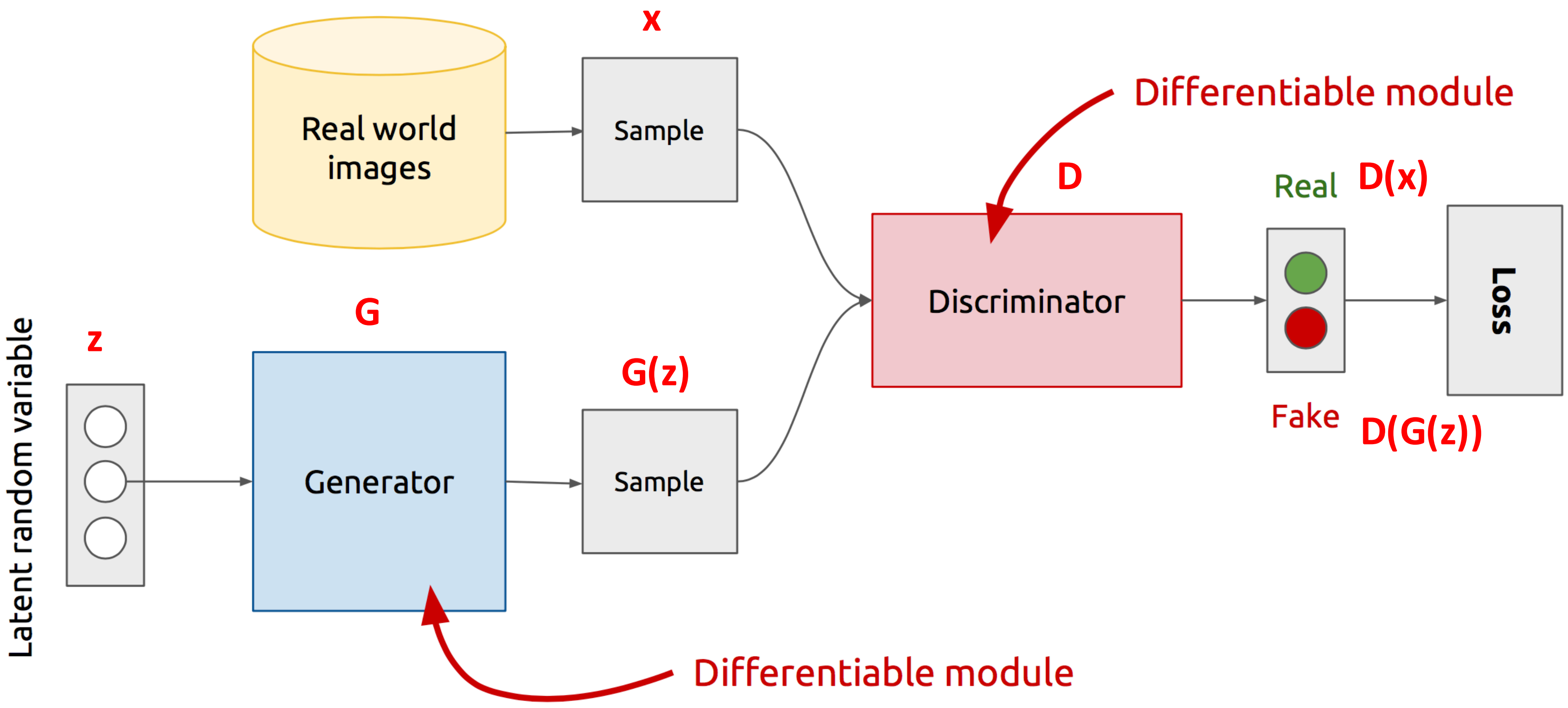
http://people.eecs.berkeley.edu/~junyanz/projects/gvm/

Adversarial Training

- **In the last lecture, we saw:**
  - We can 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

- **GANs extend that idea to generative models:**
  - 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

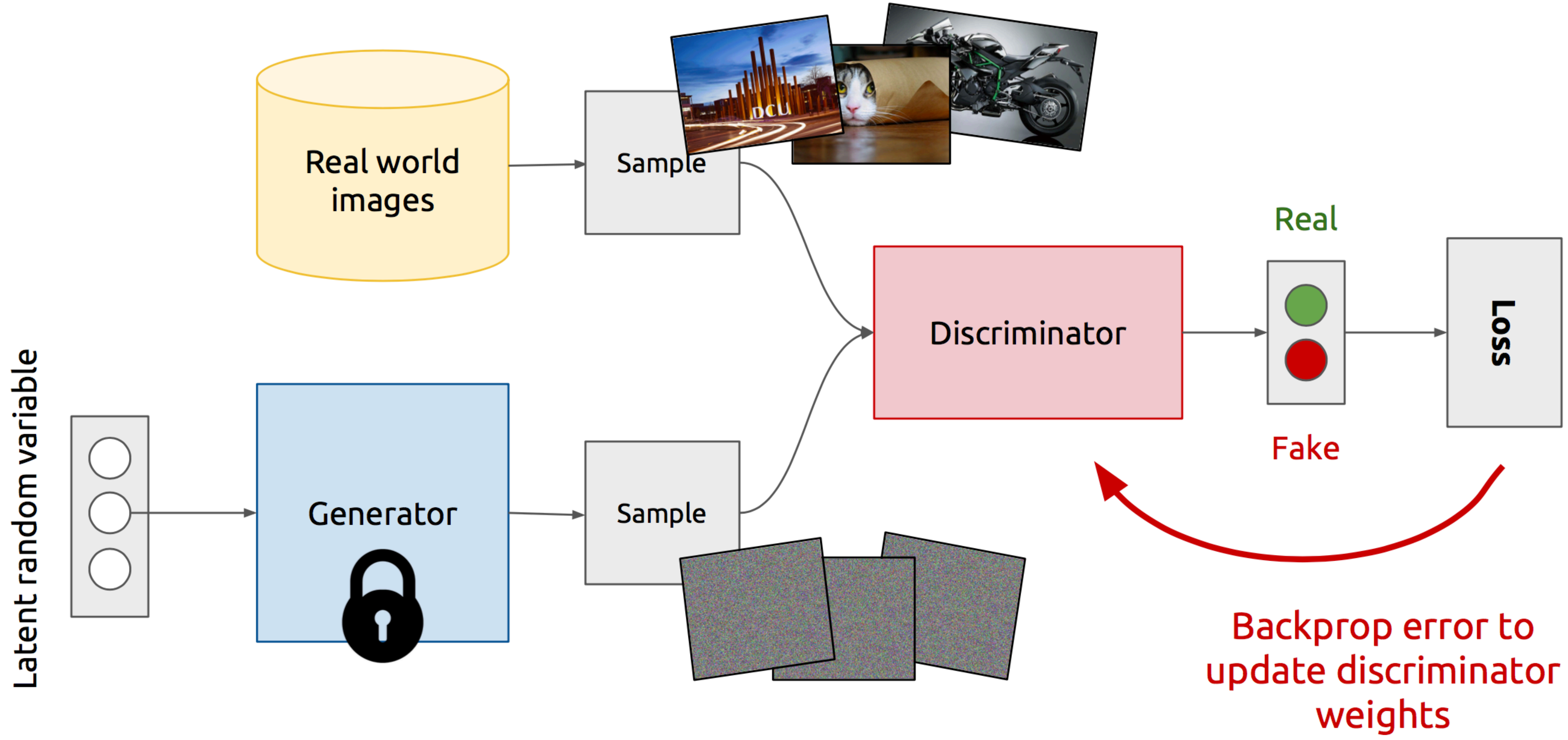
GAN's Architecture



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

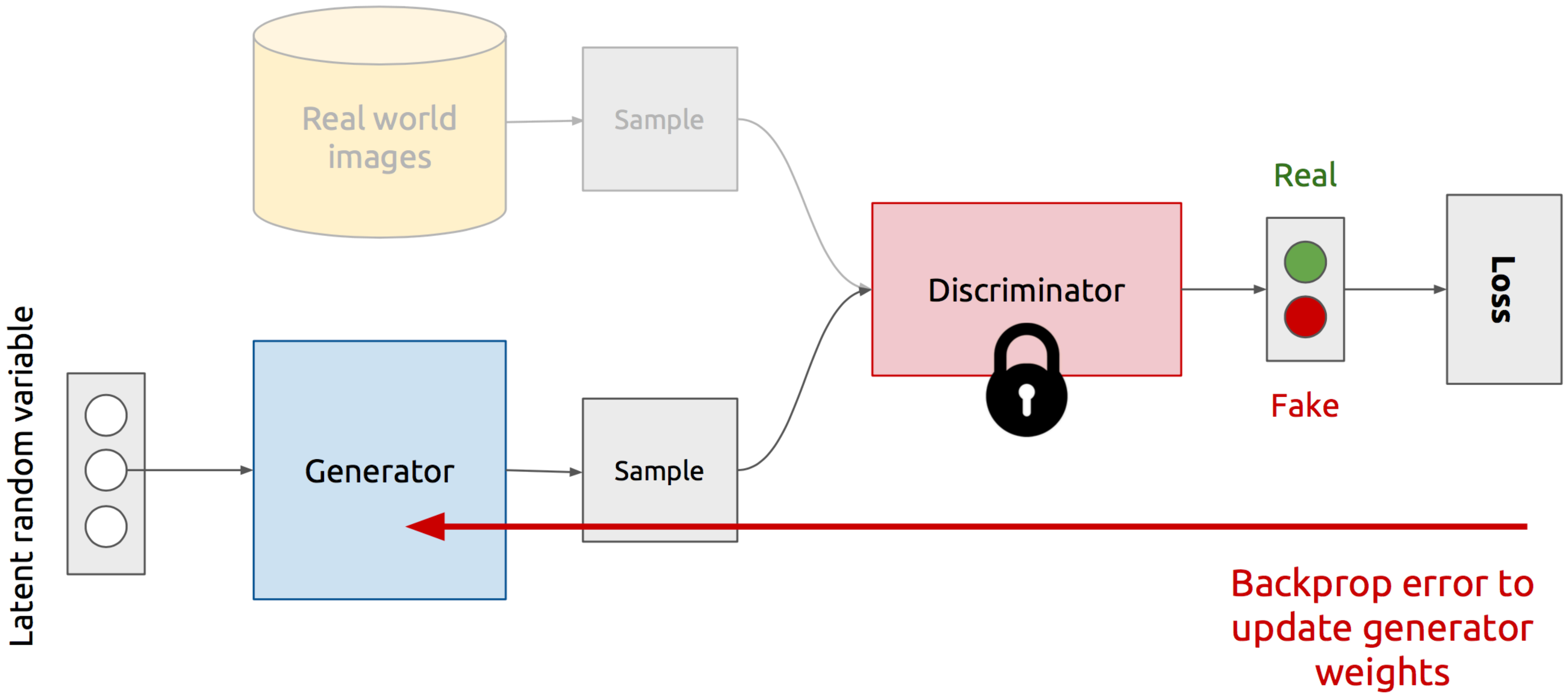
https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Training Discriminator



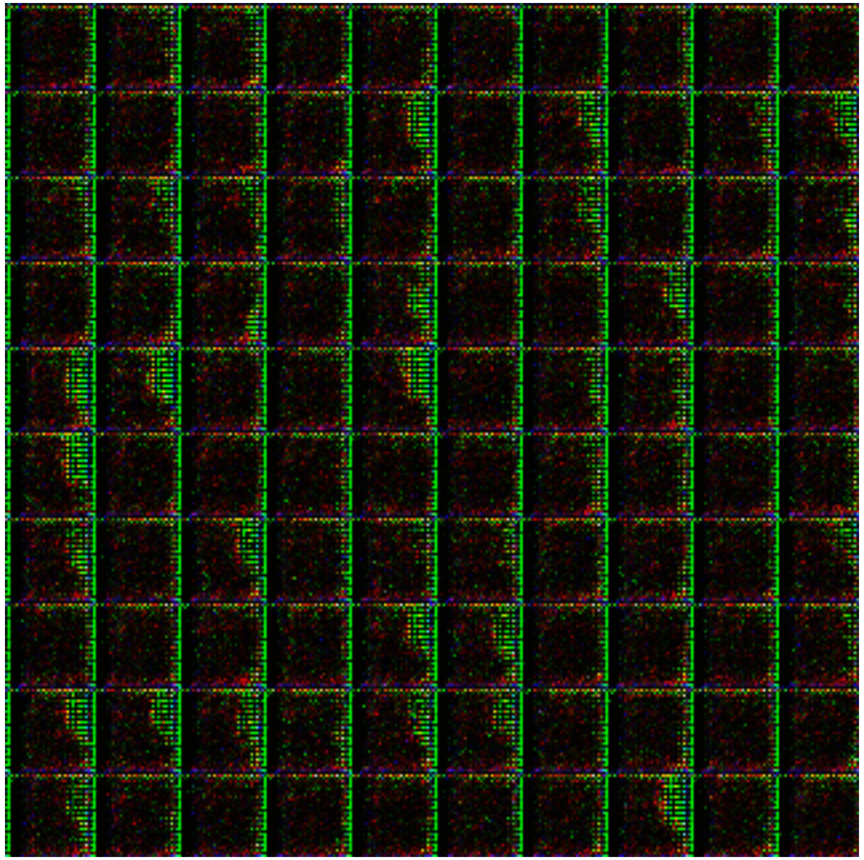
https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

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)

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

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

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.

Vanishing gradient strikes back again…

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

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

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

- $\nabla_a \log\left(1 - \sigma(a)\right) = \frac{-\nabla_a \, \sigma(a)}{1 - \sigma(a)} = \frac{-\sigma(a)\left(1 - \sigma(a)\right)}{1 - \sigma(a)} = -\sigma(a) = -D\left(G(z)\right)$

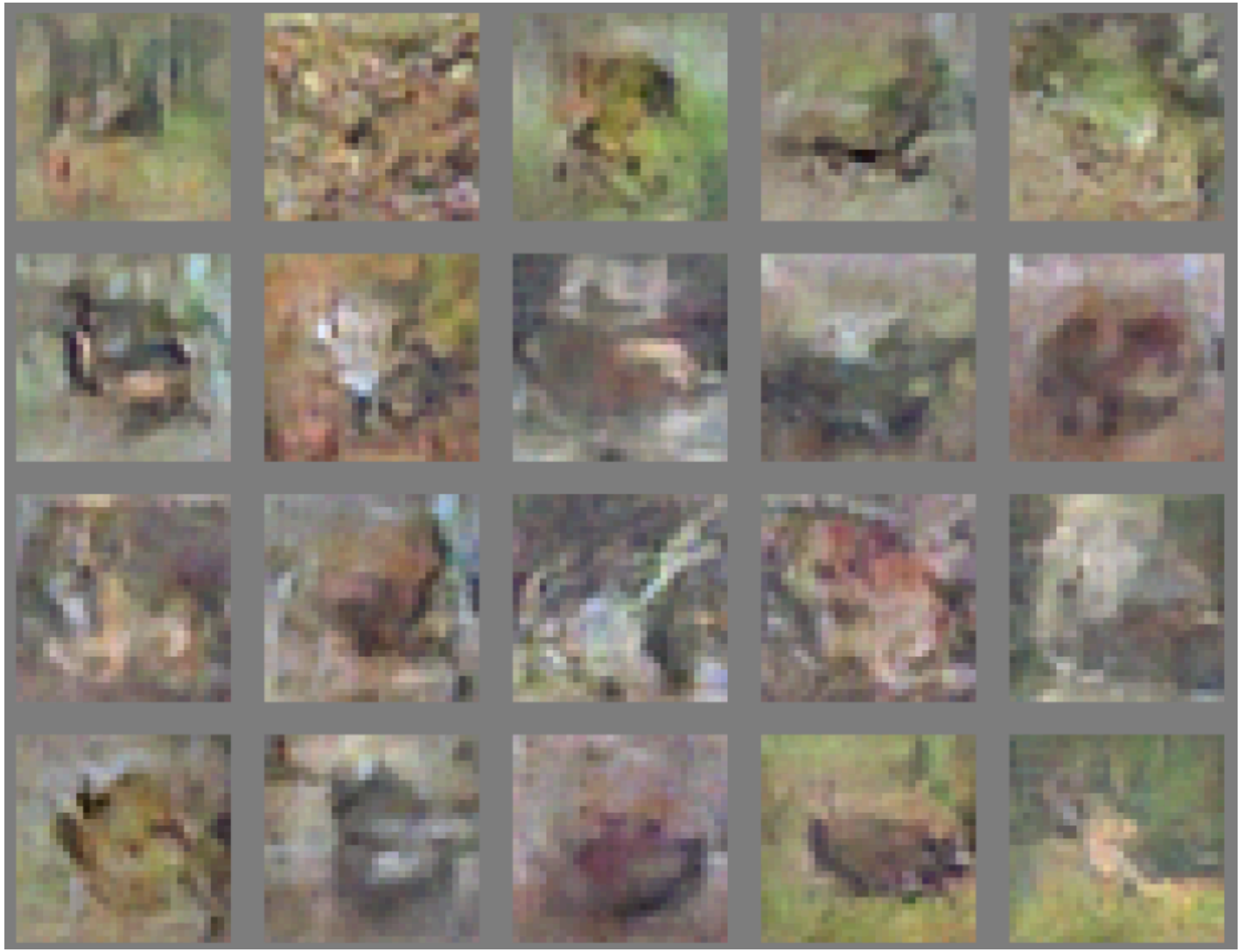
- Gradient goes to 0 if D is confident, i.e. $D\left(G(z)\right) \rightarrow 0$

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

Faces

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

CIFAR



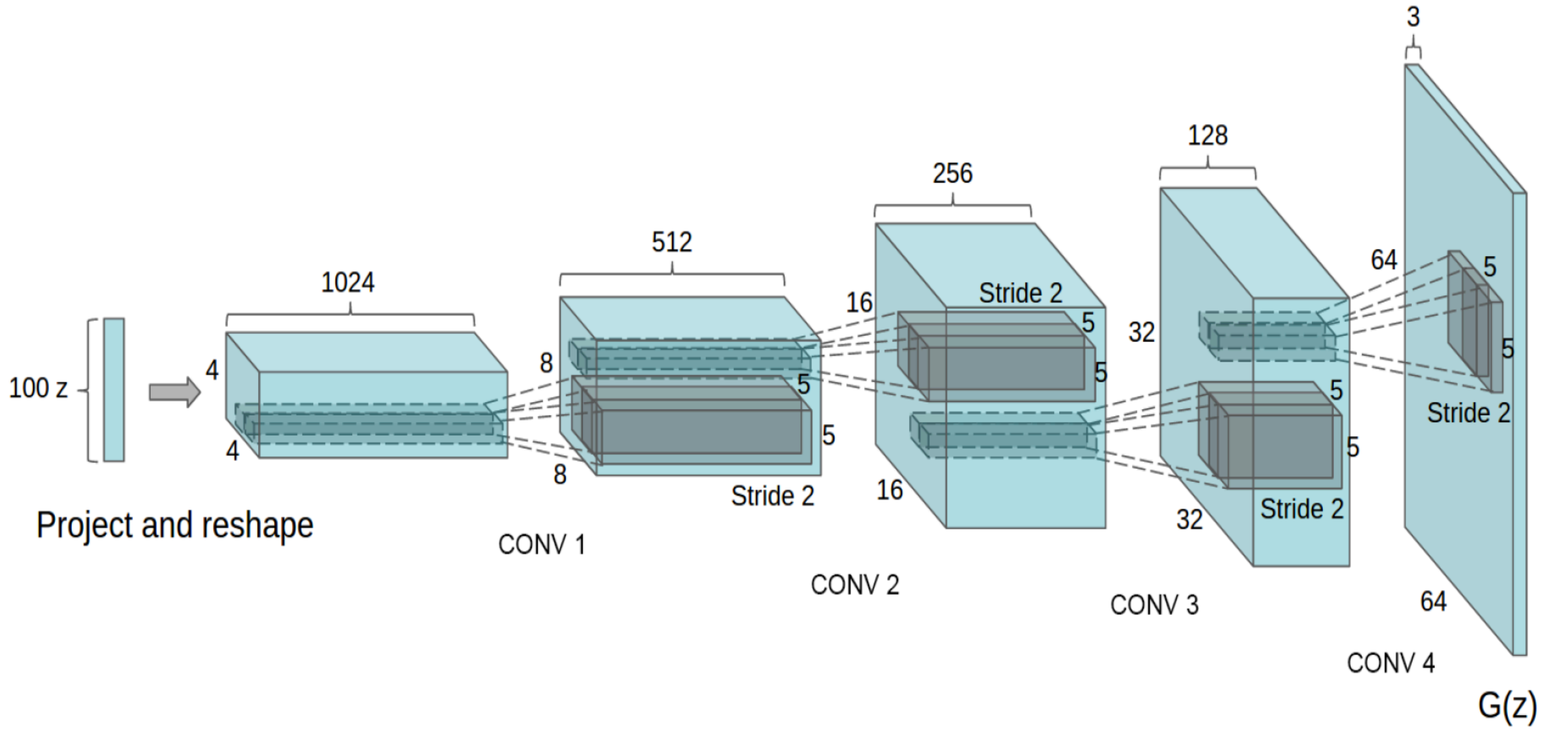
Goodfellow, Ian, et al. **"Generative adversarial nets."** *Advances in neural information processing systems*. 2014.

DCGAN: Bedroom images

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

Deep Convolutional GANs (DCGANs)

Generator Architecture

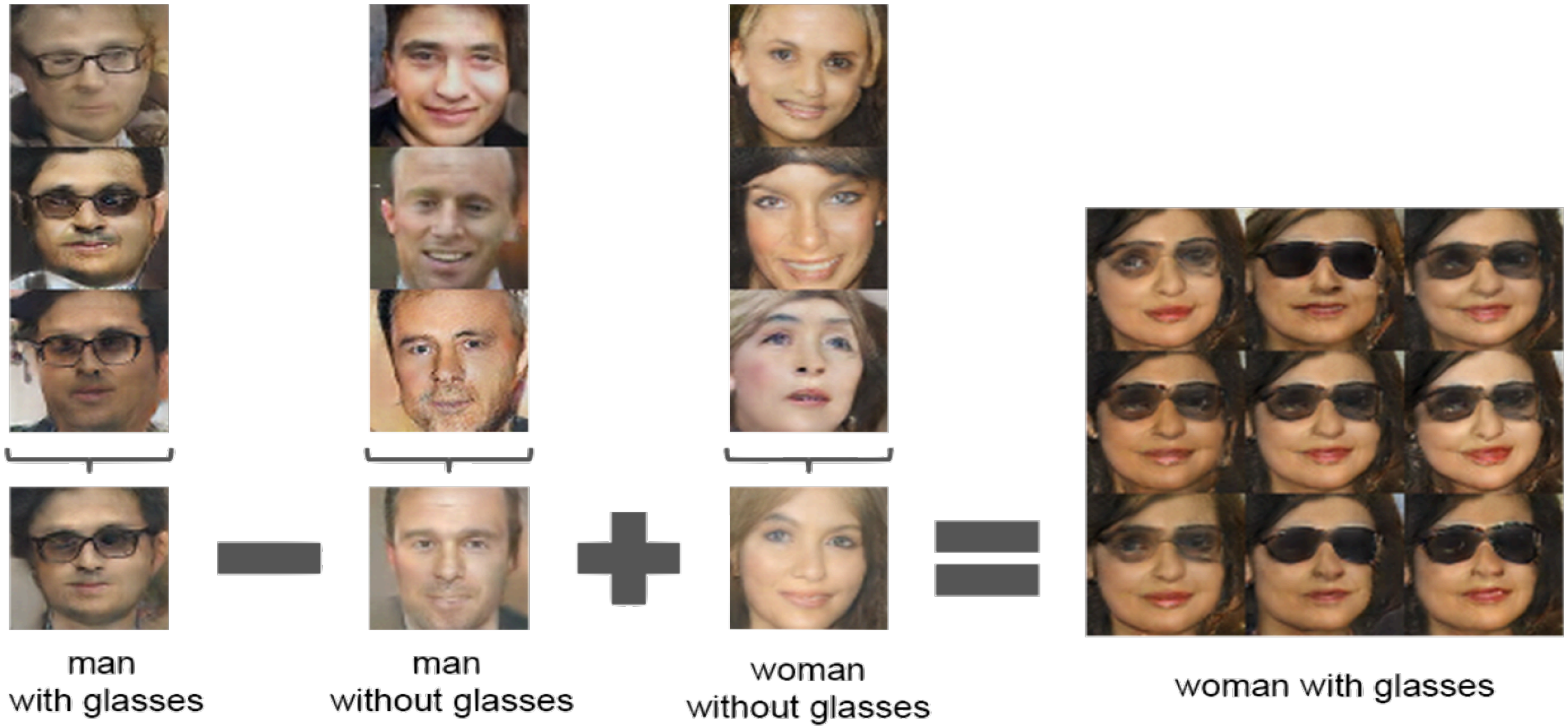


Key ideas:

- 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).

Latent vectors capture interesting patterns...



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

Part 2

- **Advantages of GANs**
- **Training Challenges**
  - Non-Convergence
  - Mode-Collapse
- **Proposed Solutions**
  - Supervision with Labels
  - Mini-Batch GANs
- **Modification of GAN's losses**
  - Discriminator **(EB-GAN)**
  - Generator **(InfoGAN)**

Advantages of GANs

- **Plenty of existing work on Deep Generative Models**
  - Boltzmann Machine
  - Deep Belief Nets
  - Variational AutoEncoders (VAE)

- **Why GANs?**
  - Sampling (or generation) is straightforward.
  - Training doesn't 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.

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

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

- **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

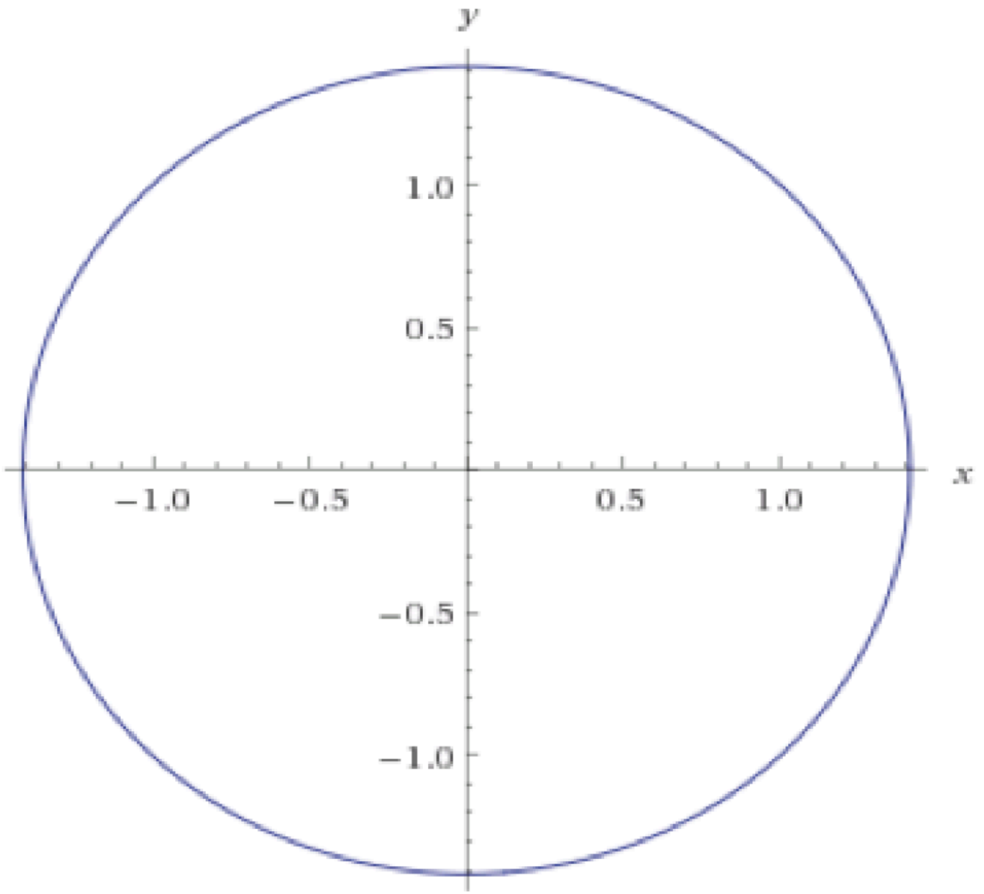
- State 5: == State 1

x > 0	y > 0	V > 0	Increase y	Decrease x

Non-Convergence

$$\min_{x} \max_{y} \; xy$$

- $\frac{\partial}{\partial x} = -\mathrm{y} \quad \ldots \quad \frac{\partial}{\partial y} = \mathrm{x}$
- $\frac{\partial^2}{\partial y^2} = \frac{\partial}{\partial x} = -y$
- Differential equation's solution has sinusoidal terms
- Even with a small learning rate, it will not converge



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

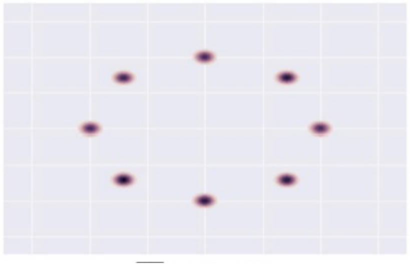
Problems with GANs

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

Mode-Collapse

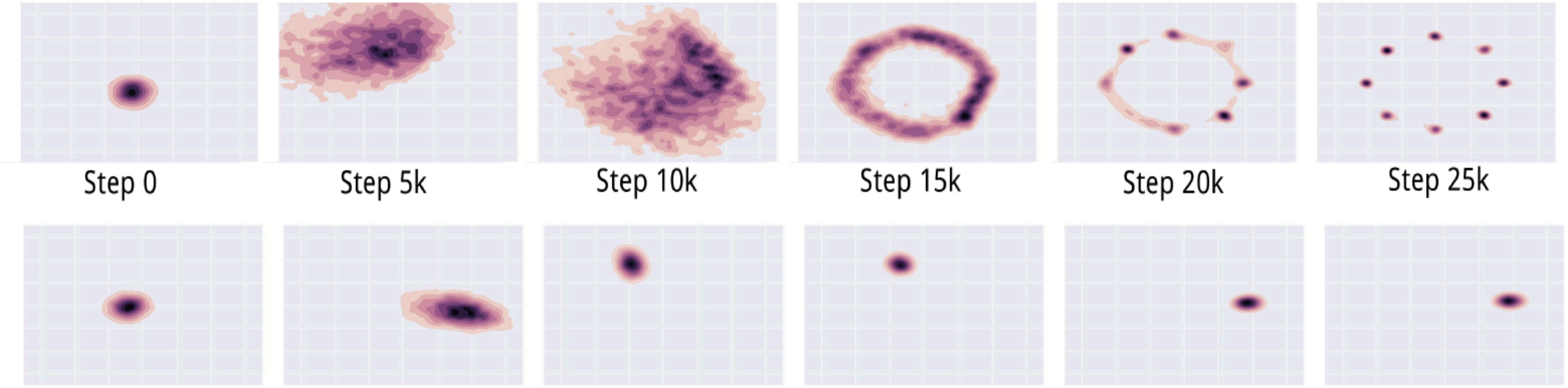
- **Generator fails to output diverse samples**

Target



Expected

Output



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

Some real examples

Reed, S., et al. ***Generating interpretable images with controllable structure.*** Technical report, 2016. 2, 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

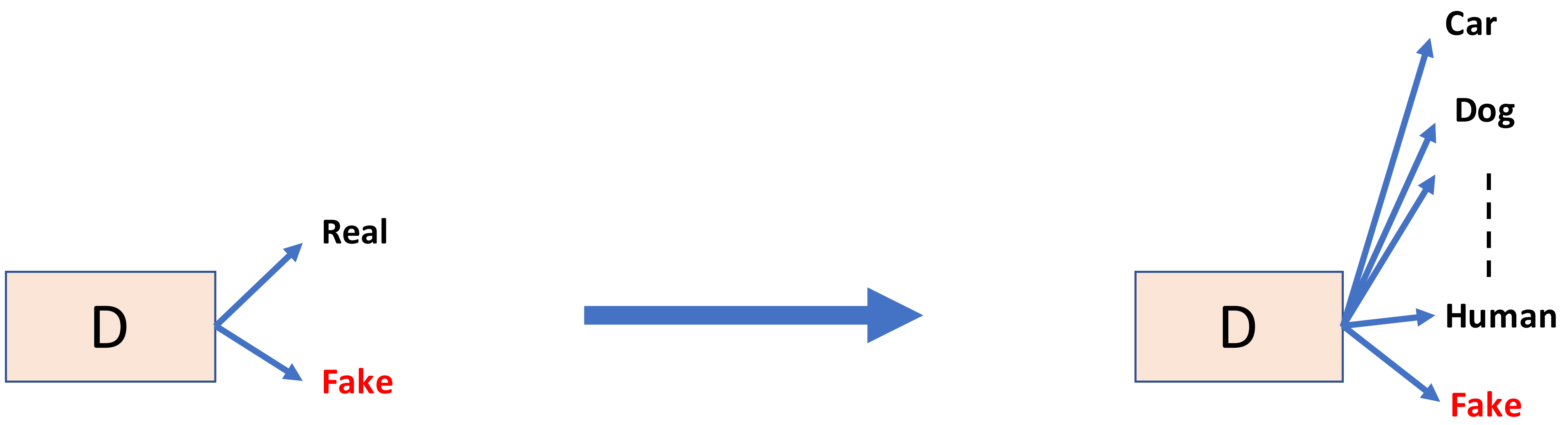
Salimans, Tim, et al. "Improved techniques for training gans." *Advances in Neural Information Processing Systems*. 2016.

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.

Alternate view of GANs

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

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

- In this formulation, Discriminator's strategy was $D(x) \rightarrow 1, \ \ D\left(G(z)\right) \rightarrow 0$

- 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 $D(x) \rightarrow 0, \ \ D\left(G(z)\right) \rightarrow 1$

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 $D(x) \rightarrow 0, \quad D\big(G(z)\big) \rightarrow 1$

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

We can use this

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

- Now, we can replace cross-entropy with any loss function **(Hinge Loss)**

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

- And thus, instead of outputting probabilities, Discriminator just has to output :-
  - High values for fake samples
  - Low values for real samples

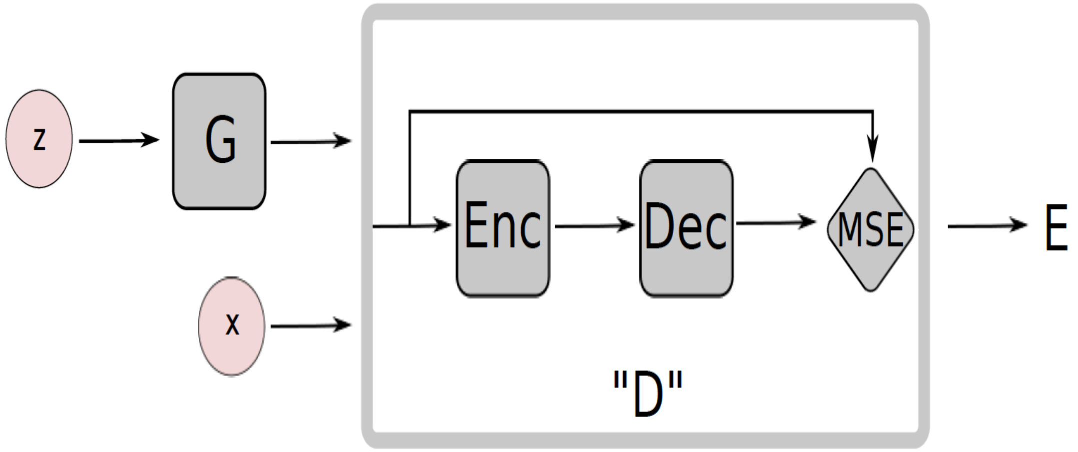
Zhao, Junbo, Michael Mathieu, and Yann LeCun. **"Energy-based generative adversarial network."** arXiv preprint arXiv:1609.03126 (2016)

Energy-Based GANs

- Modified game plans
  - **Generator** will try to generate samples with low values
  - **Discriminator** will try to assign high scores to fake values

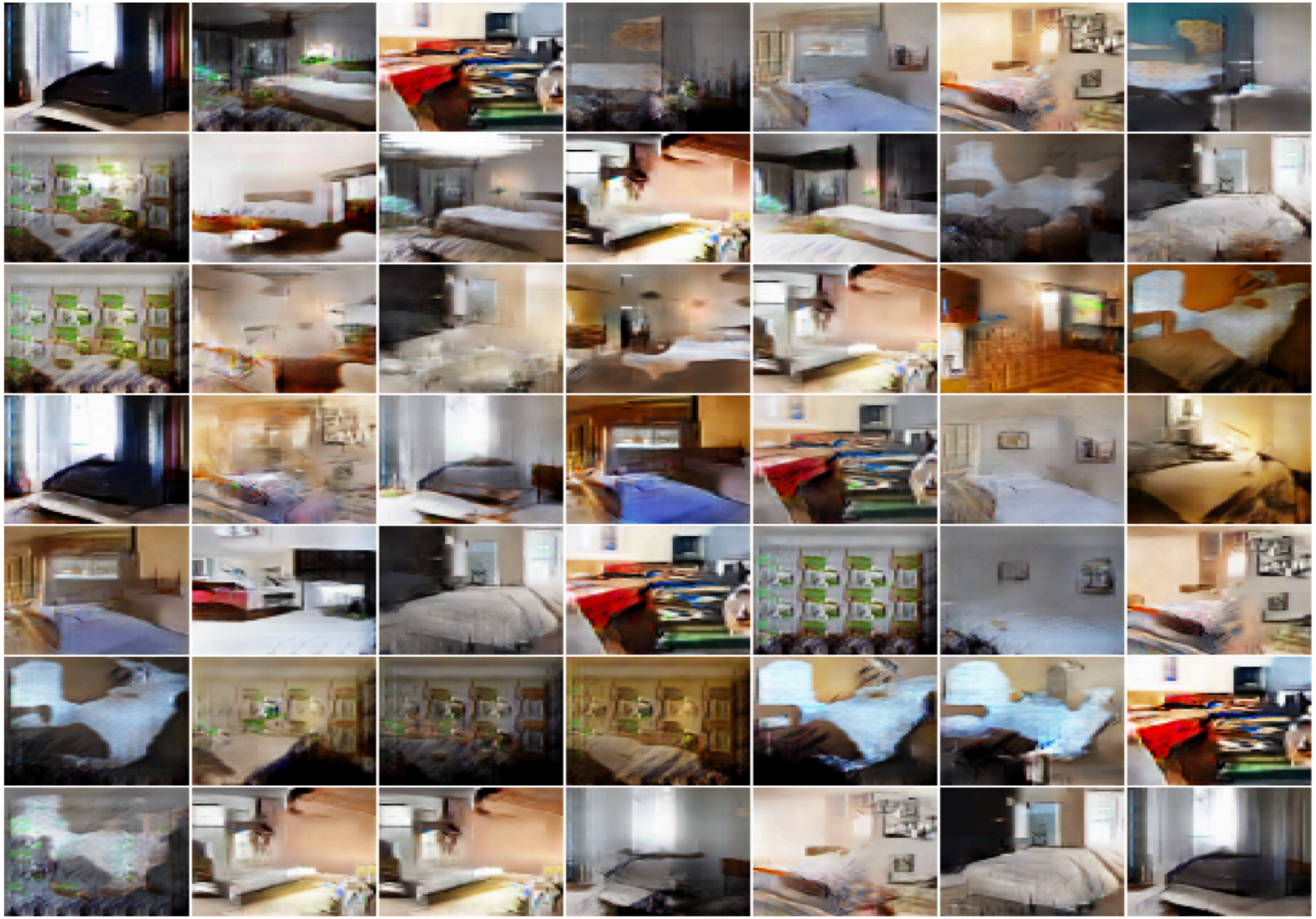
$$D(x) = \ ||Dec\big(Enc(x)\big) - x||_{MSE}$$

- **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



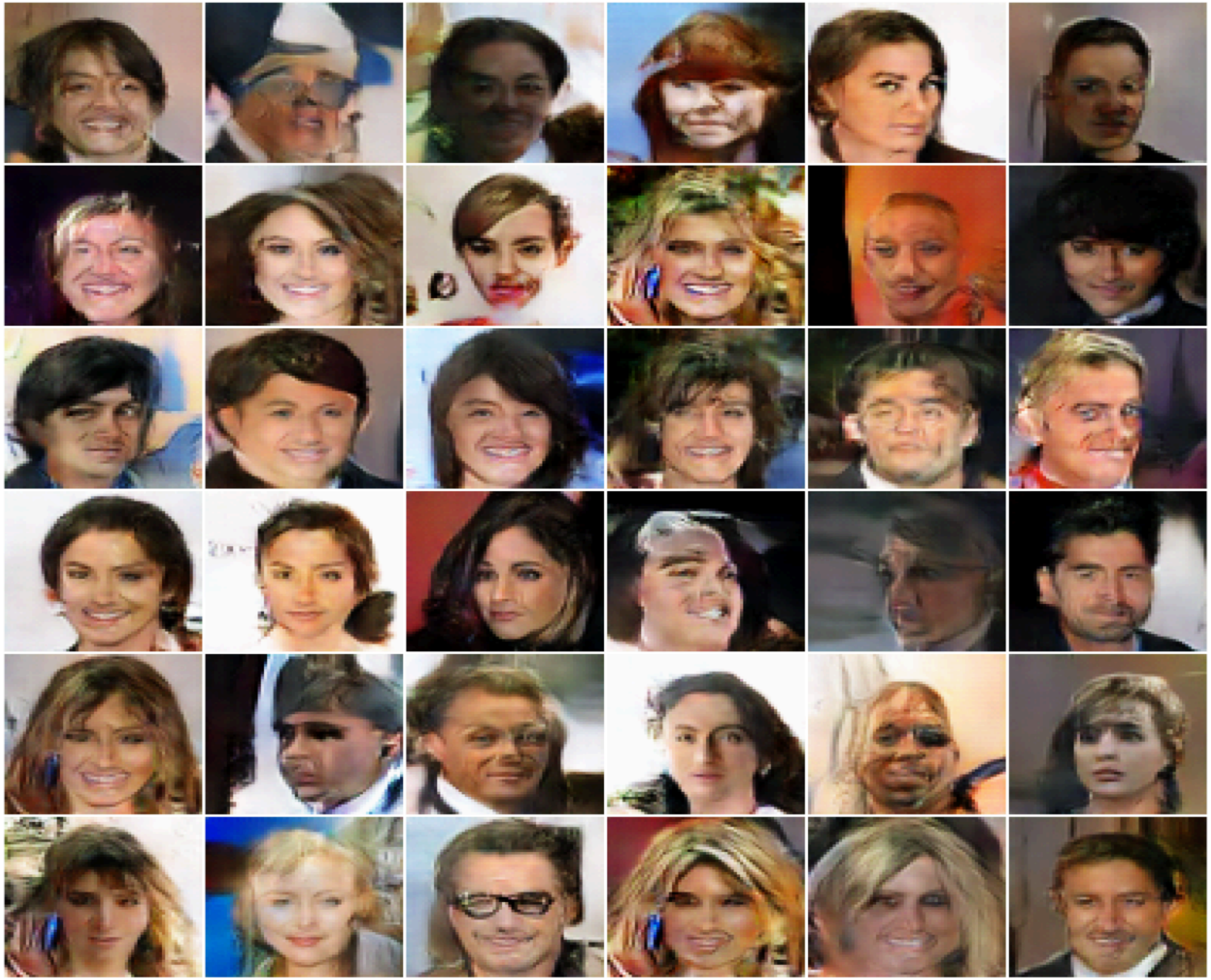
Zhao, Junbo, Michael Mathieu, and Yann LeCun. **"Energy-based generative adversarial network."** arXiv preprint arXiv:1609.03126 (2016)

More Bedrooms…



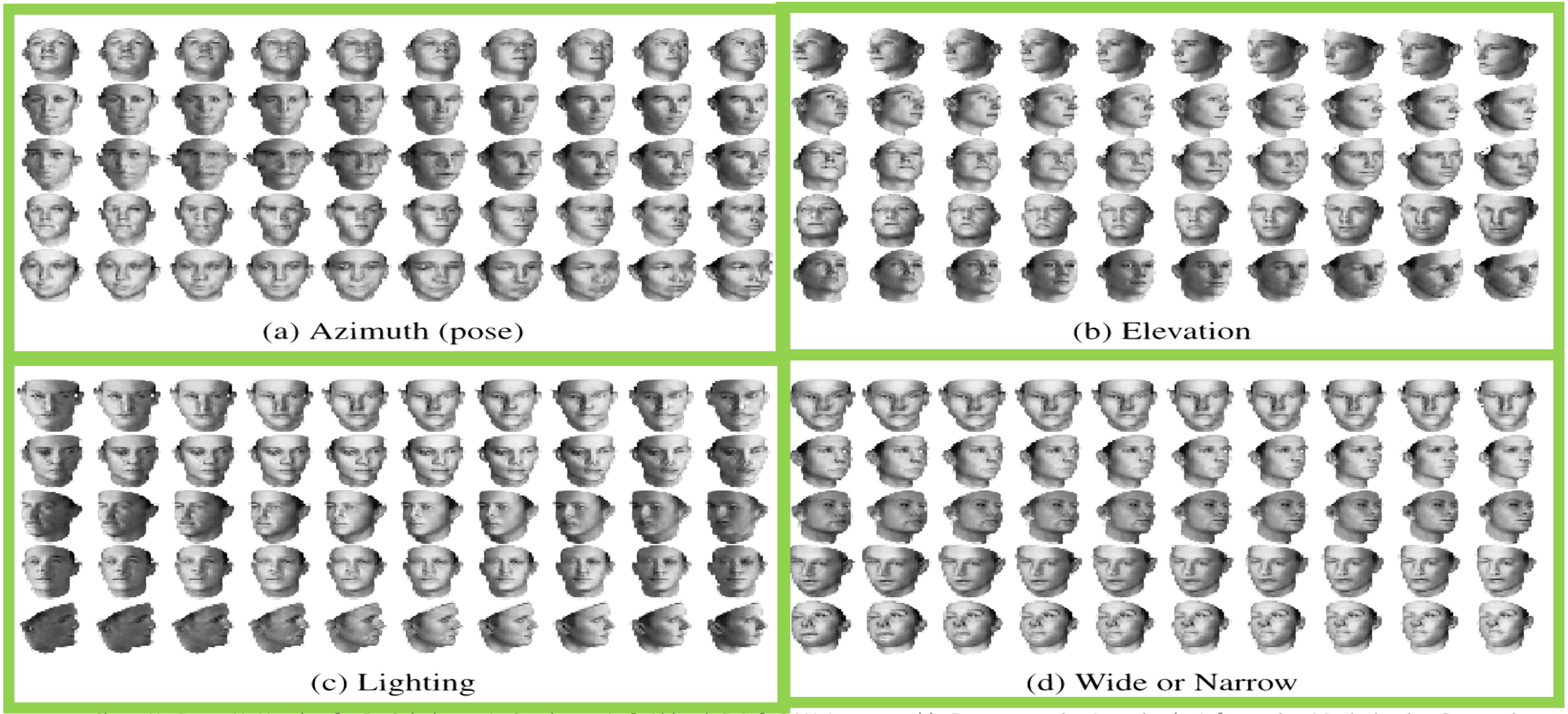
Zhao, Junbo, Michael Mathieu, and Yann LeCun. **"Energy-based generative adversarial network."** arXiv preprint arXiv:1609.03126 (2016)

Celebs...



The Cool Stuff…

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).

Cool Stuff (contd.)

3D Chairs

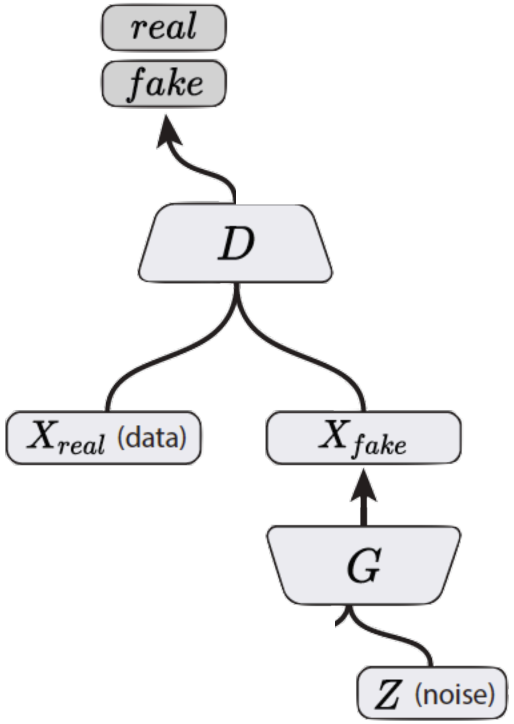
(a) Rotation (b) Width

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?

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

Recap: Mutual Information

- **Mutual Information captures the mutual dependence between two variables**

- **Mutual information between two variables $\boldsymbol{X, Y}$ is defined as:**

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

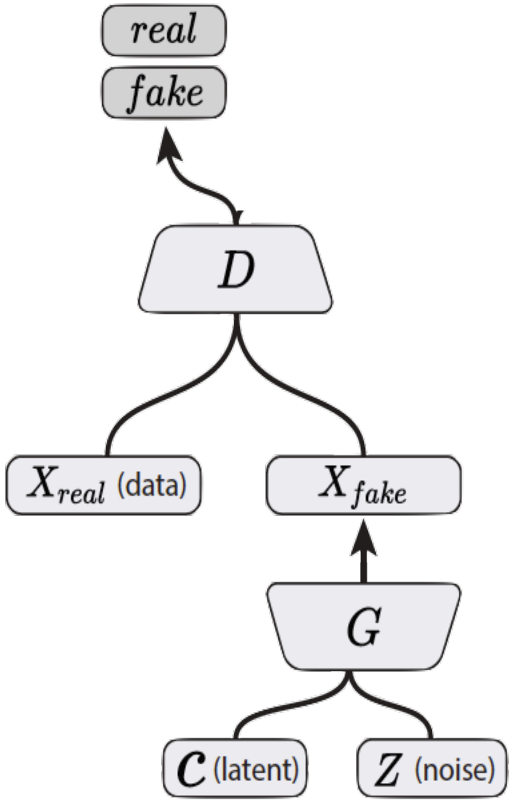
$$I(X;Y) = H(X) - H(X|Y) = \mathrm{H(Y)} - \mathrm{H(Y|X)}$$

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.

$$\min_{G} \max_{D} V_I(D, G) = V(D, G) \boxed{- \lambda\, I\big(c; G(z, c)\big)}$$



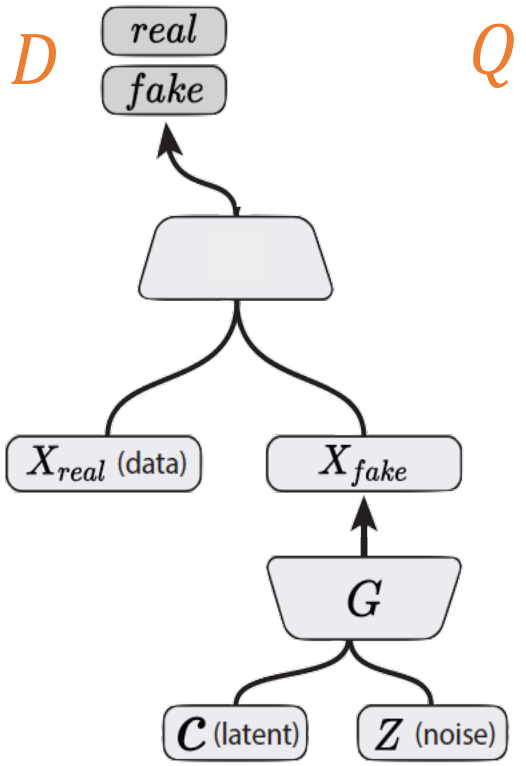
Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets, NIPS (2016).

InfoGAN

Mutual Information's Variational Lower bound

$$I\big(c; G(z,c)\big) = H(c) - H(c|G(z,c))$$
$$= \mathbb{E}_{x \sim G(z,c)}\Big[\mathbb{E}_{c' \sim P(c|x)}[\log P(c'|x)]\Big] + H(c)$$
$$= \mathbb{E}_{x \sim G(z,c)}\Big[D_{KL}(P||Q) + \mathbb{E}_{c' \sim P(c|x)}[\log Q(c'|x)]\Big] + H(c)$$
$$\geq \mathbb{E}_{x \sim G(z,c)}\Big[\mathbb{E}_{c' \sim P(c|x)}[\log Q(c'|x)]\Big] + H(c)$$

$$\geq \mathbb{E}_{c \sim P(c),\ x \sim G(z,c)}\boldsymbol{[\log Q(c|x)]} + H(c)$$



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

Part 3

- **Conditional GANs**
- **Applications**
  - Image-to-Image Translation
  - Text-to-Image Synthesis
  - Face Aging
- **Advanced GAN Extensions**
  - Coupled GAN
  - LAPGAN – Laplacian Pyramid of Adversarial Networks
  - Adversarially Learned Inference
- **Summary**

Conditional GANs

MNIST digits generated conditioned on their class label.

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

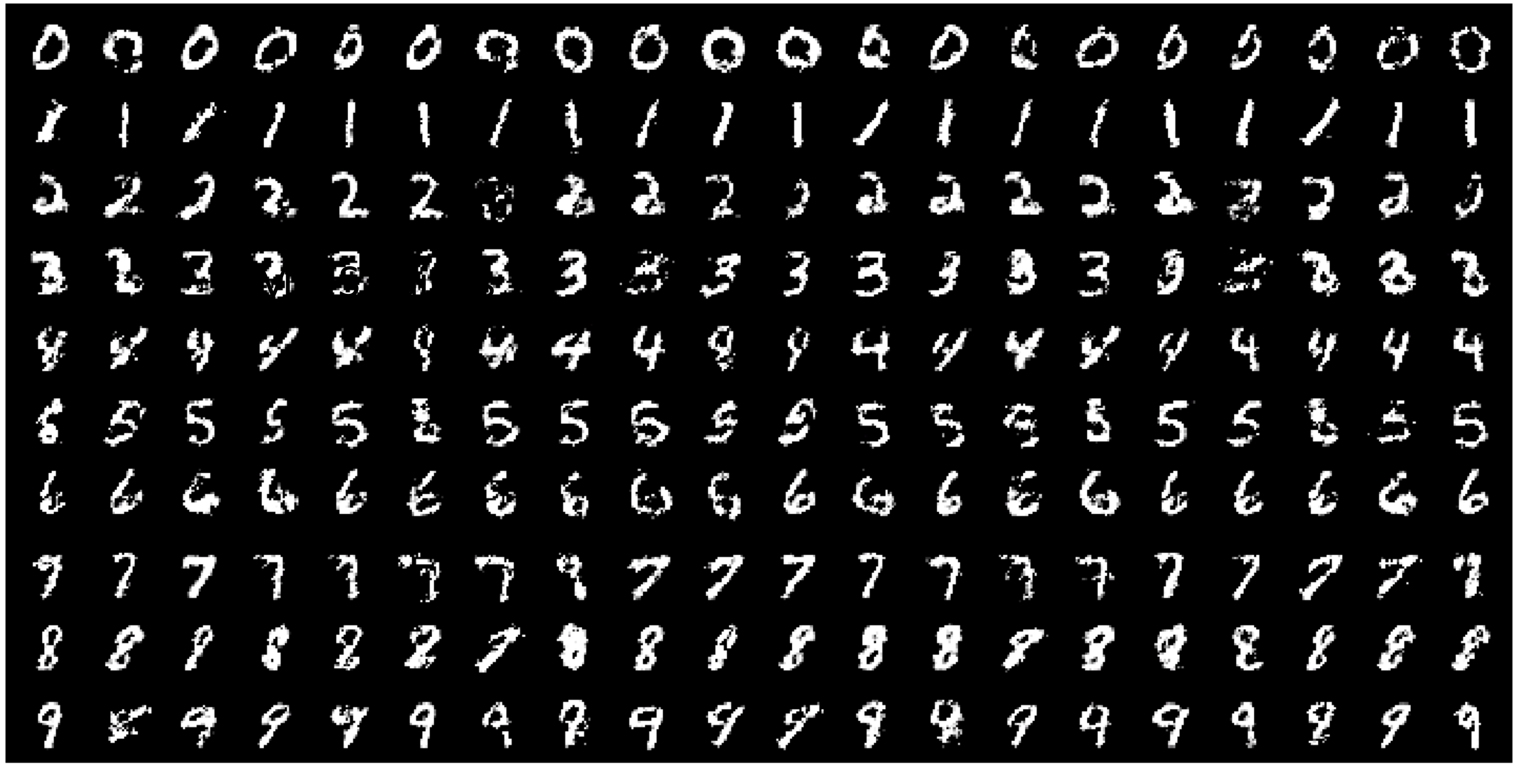


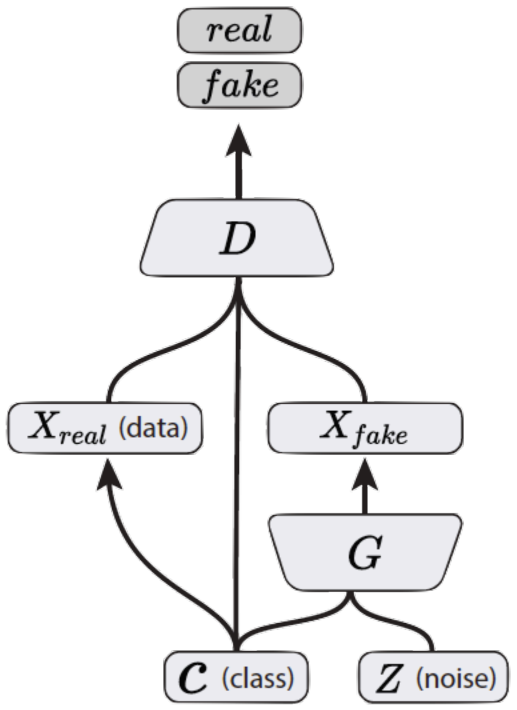
Figure 2 in the original paper.

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

Conditional GANs

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

- Lends to many practical applications of GANs when we have explicit *supervision* available.



Conditional GAN
(Mirza & Osindero, 2014)

Image Credit: Figure 2 in Odena, A., Olah, C. and Shlens, J., 2016. Conditional image synthesis with auxiliary classifier GANs. *arXiv preprint arXiv:1610.09585*.

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

Part 3

- Conditional GANs
- **Applications**
  - Image-to-Image Translation
  - Text-to-Image Synthesis
  - Face Aging
- Advanced GAN Extensions
  - Coupled GAN
  - LAPGAN – Laplacian Pyramid of Adversarial Networks
  - Adversarially Learned Inference
- Summary

Image-to-Image Translation

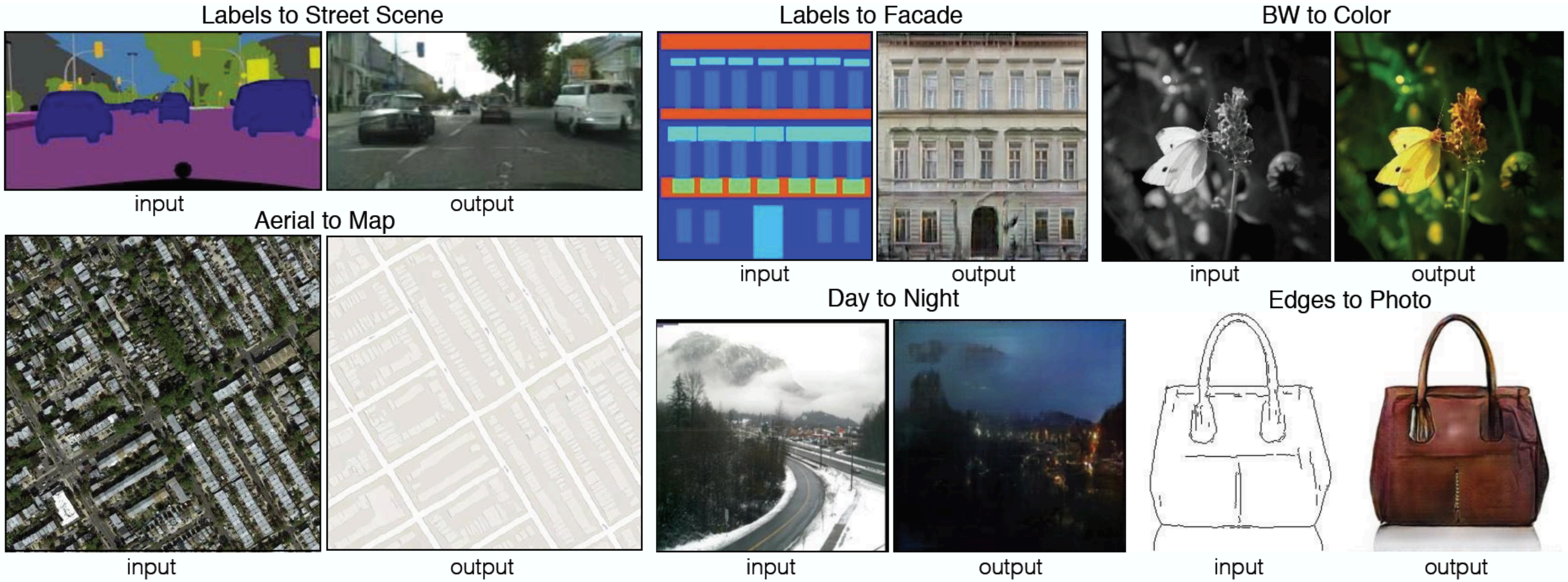


Figure 1 in the original paper.

[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.

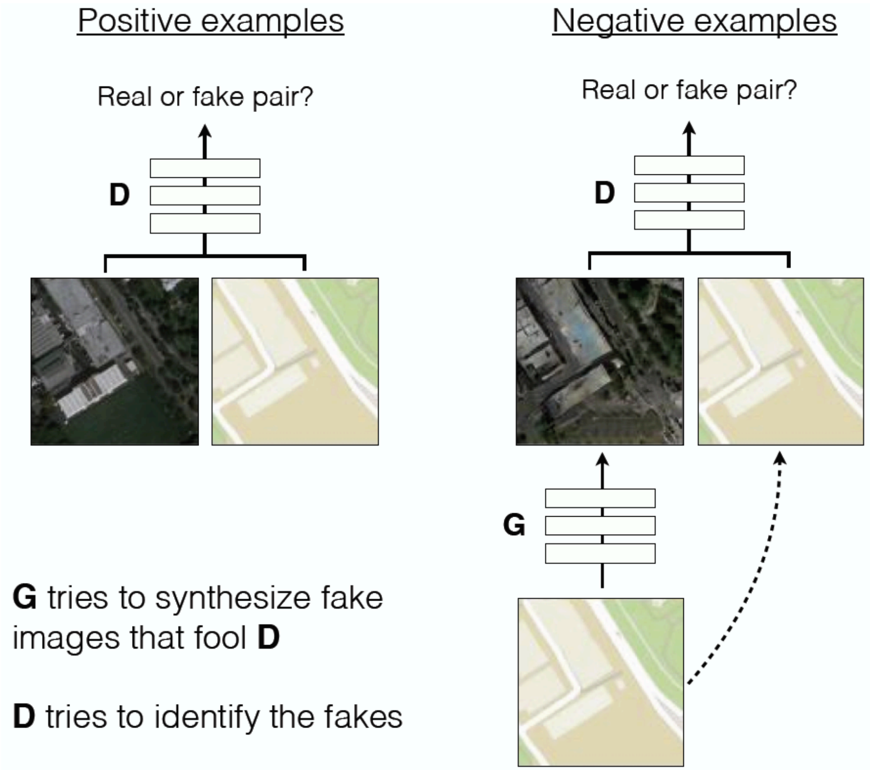


Figure 2 in the original paper.

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 condition on “dense” text embedding.

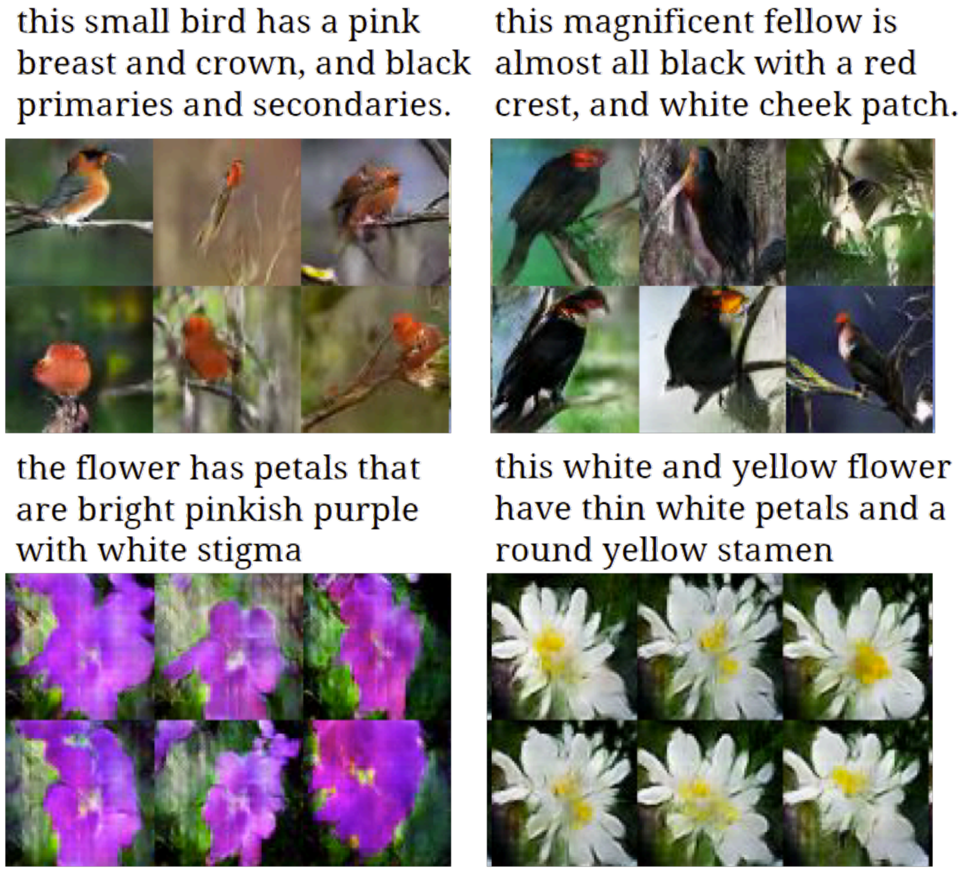


Figure 1 in the original paper.

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

Text-to-Image Synthesis

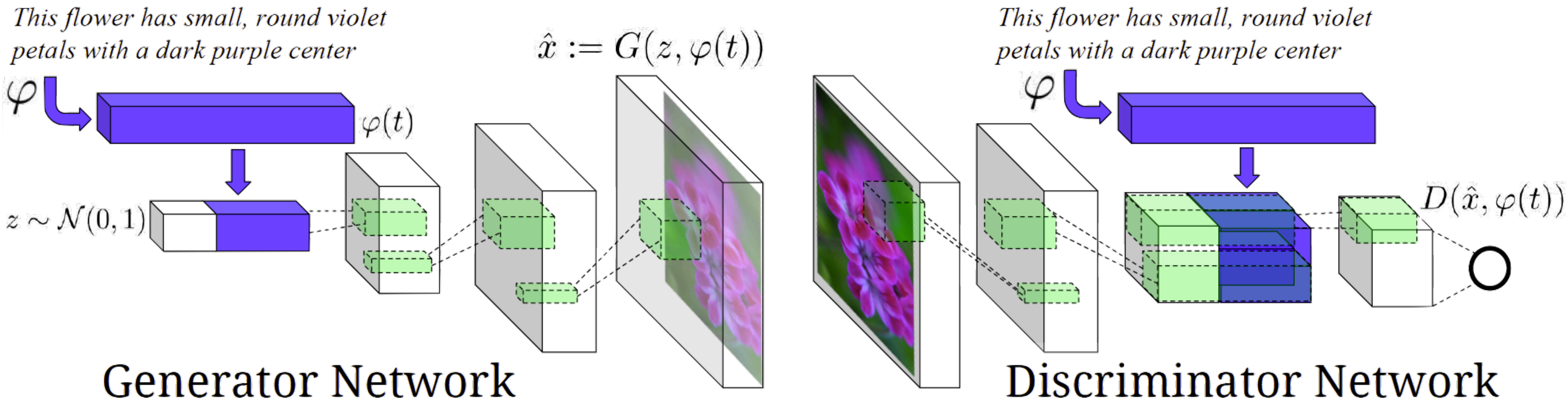


Figure 2 in the original paper.

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.

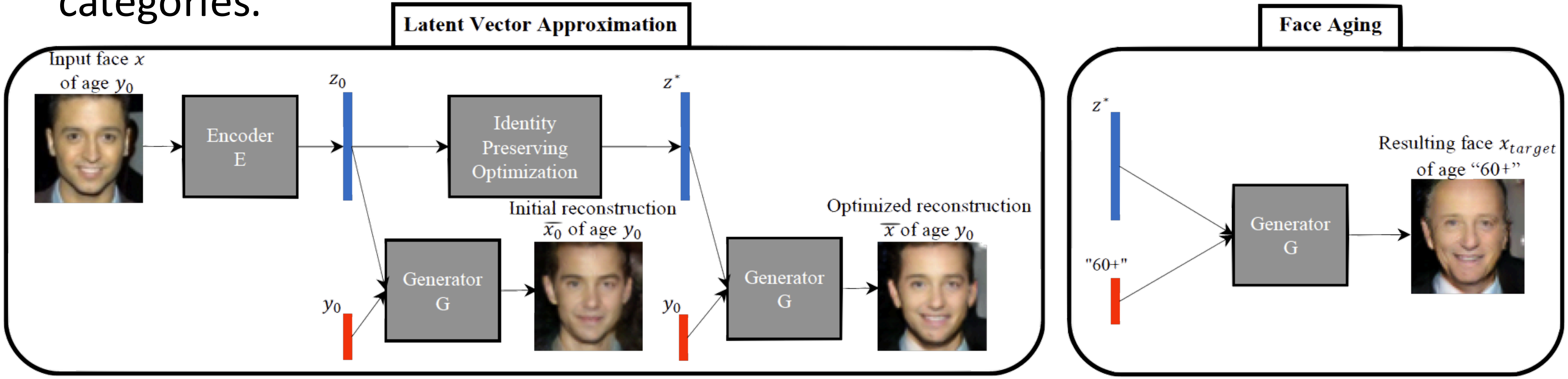


Figure 1 in the original paper.

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

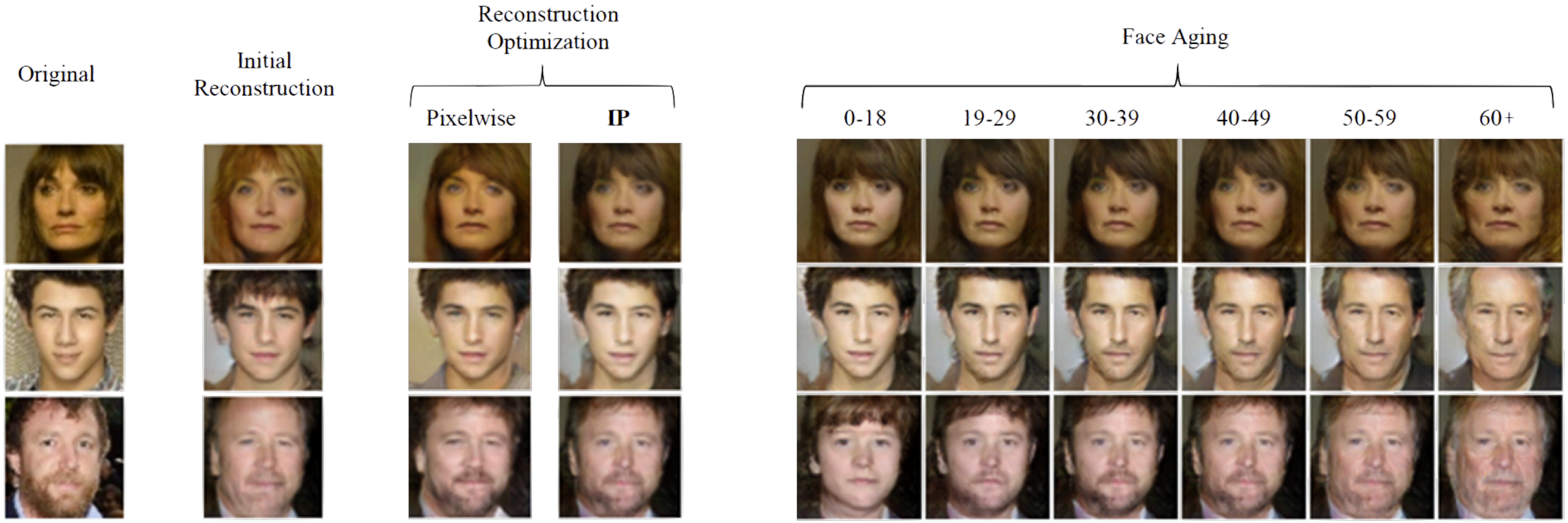


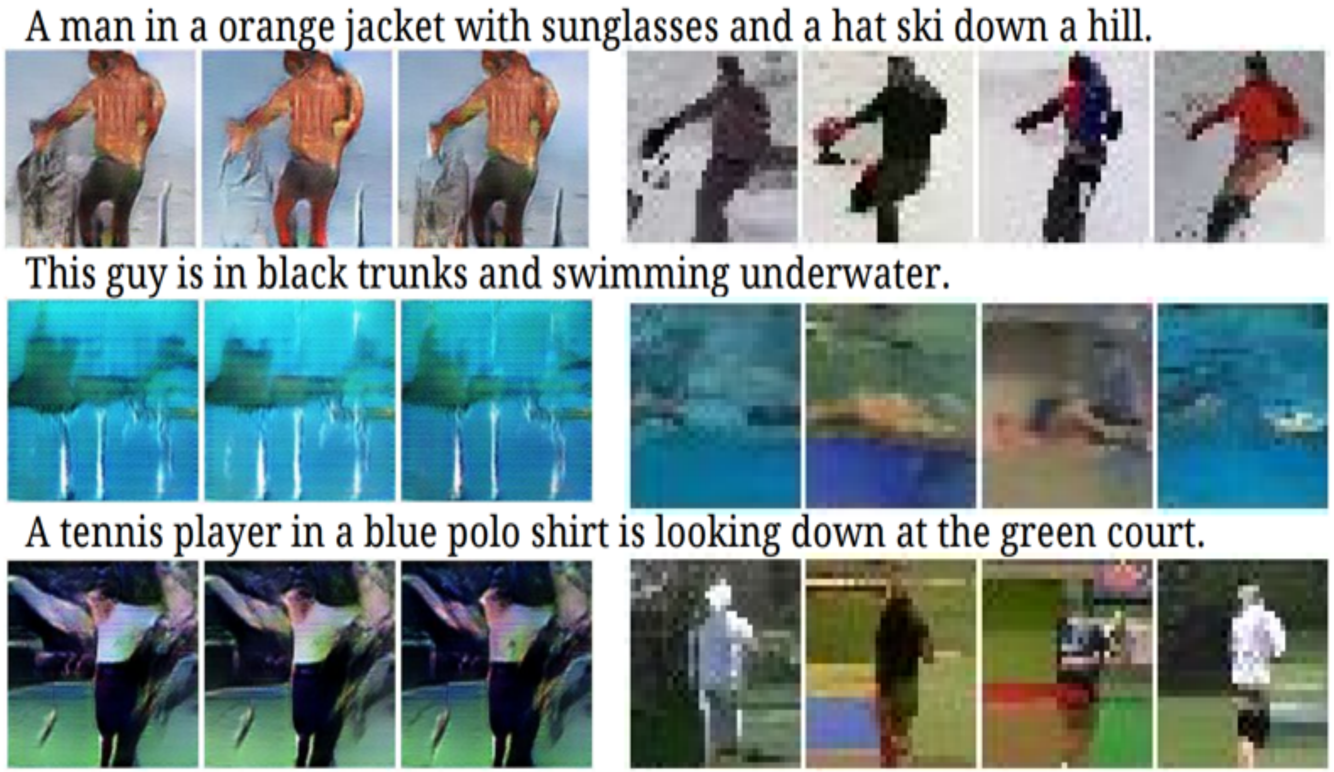
Figure 3 in the original paper.

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

Conditional GANs

Conditional Model Collapse

- Scenario observed when the Conditional GAN starts *ignoring* either the code (c) or the noise variables (z).
- This limits the diversity of images generated.



Credit?

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

Part 3

- Conditional GANs
- Applications
  - Image-to-Image Translation
  - Text-to-Image Synthesis
  - Face Aging
- **Advanced GAN Extensions**
  - Coupled GAN
  - LAPGAN – Laplacian Pyramid of Adversarial Networks
  - Adversarially Learned Inference
- Summary

Coupled GAN

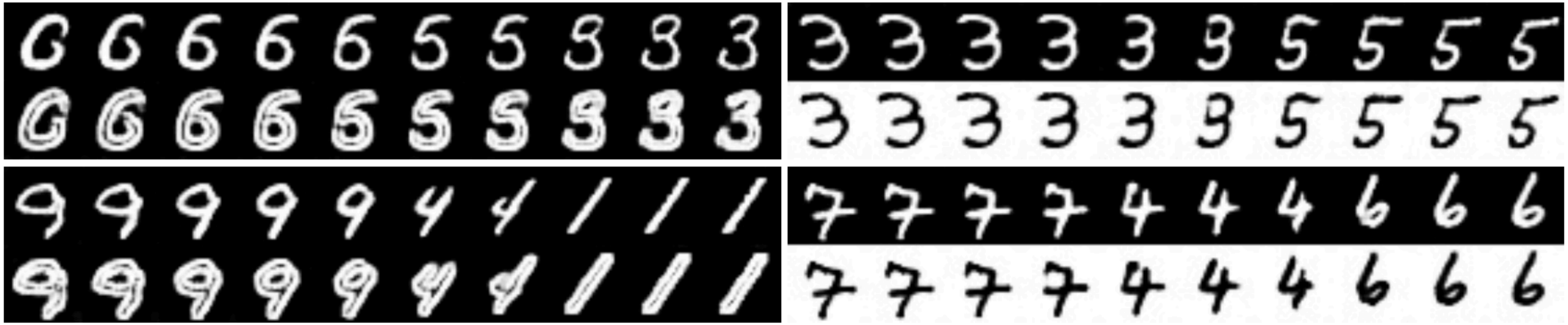


Figure 2 in the original paper.

- Learning a *joint distribution* of *multi-domain* images.
- Using GANs to learn the joint distribution with samples drawn from the marginal distributions.
- Direct applications in domain adaptation and image translation.

Liu, Ming-Yu, and Oncel Tuzel. "**Coupled generative adversarial networks**". NIPS (2016).

Coupled GANs

- Architecture

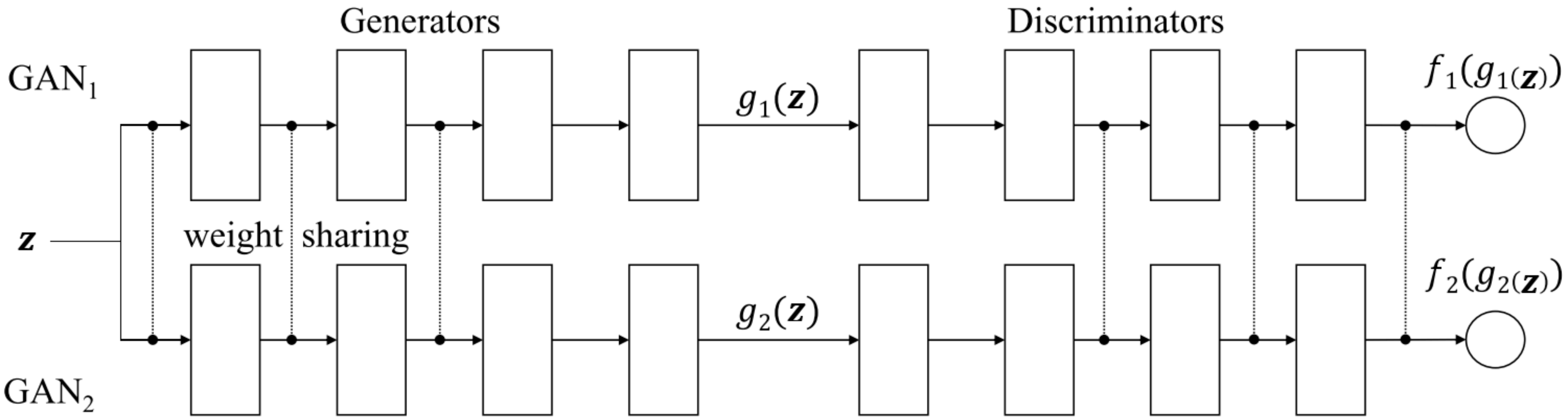


Figure 1 of the original paper.

Weight-sharing constraints the network to learn a *joint distribution* without corresponding supervision.

Liu, Ming-Yu, and Oncel Tuzel. "**Coupled generative adversarial networks**". NIPS (2016).

Coupled GANs

- Some examples of generating facial images across different feature domains.
- Corresponding images in a column are generate from the same latent code z

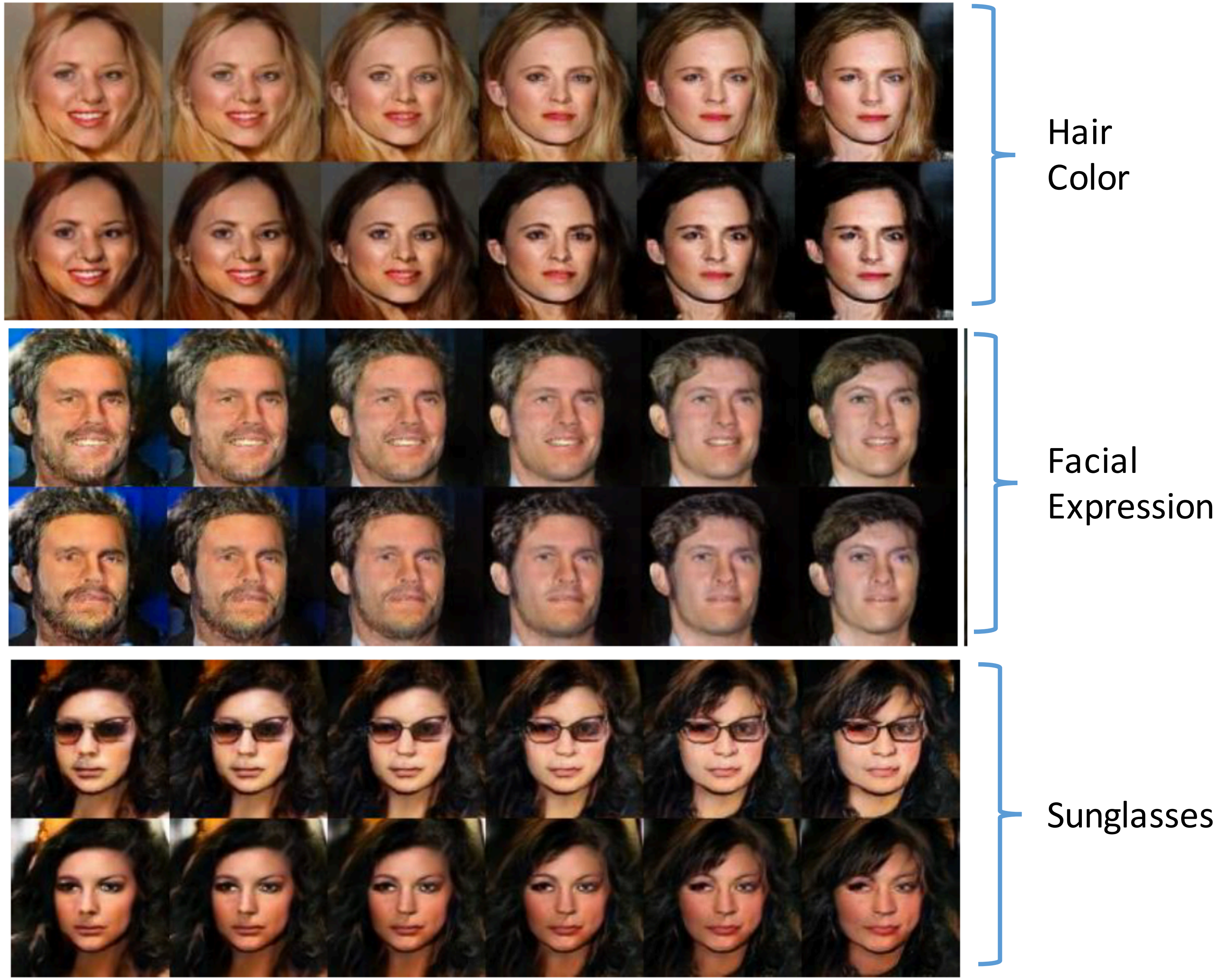


Figure 4 in the original paper.

Liu, Ming-Yu, and Oncel Tuzel. "**Coupled generative adversarial networks**". NIPS (2016).

Laplacian Pyramid of Adversarial Networks

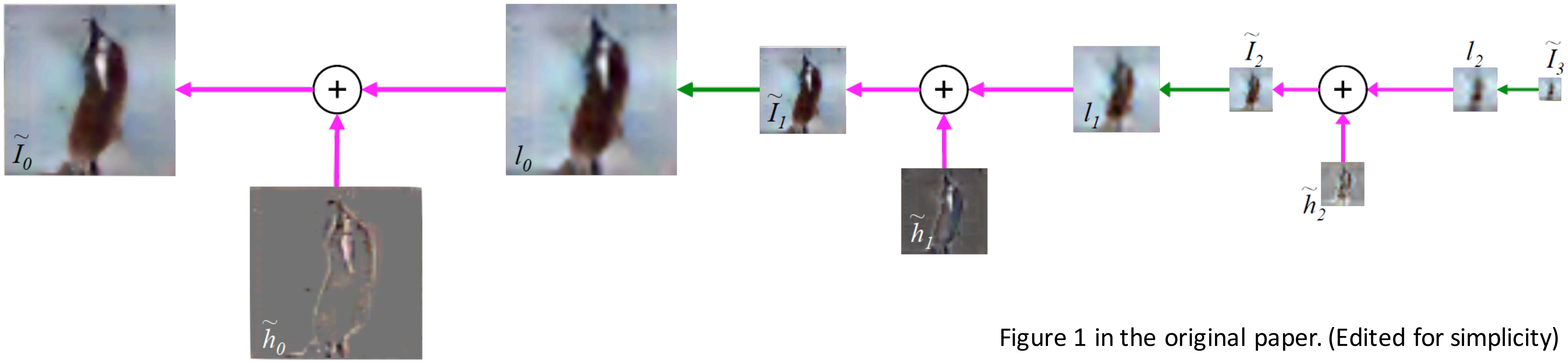


Figure 1 in the original paper. (Edited for simplicity)

- Based on the Laplacian Pyramid representation of images. (1983)
- Generate high resolution (dimension) images by using a hierarchical system of GANs
- Iteratively increase image resolution and quality.

Denton, E.L., Chintala, S. and Fergus, R., 2015. "**Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks**". NIPS (2015)

Laplacian Pyramid of Adversarial Networks

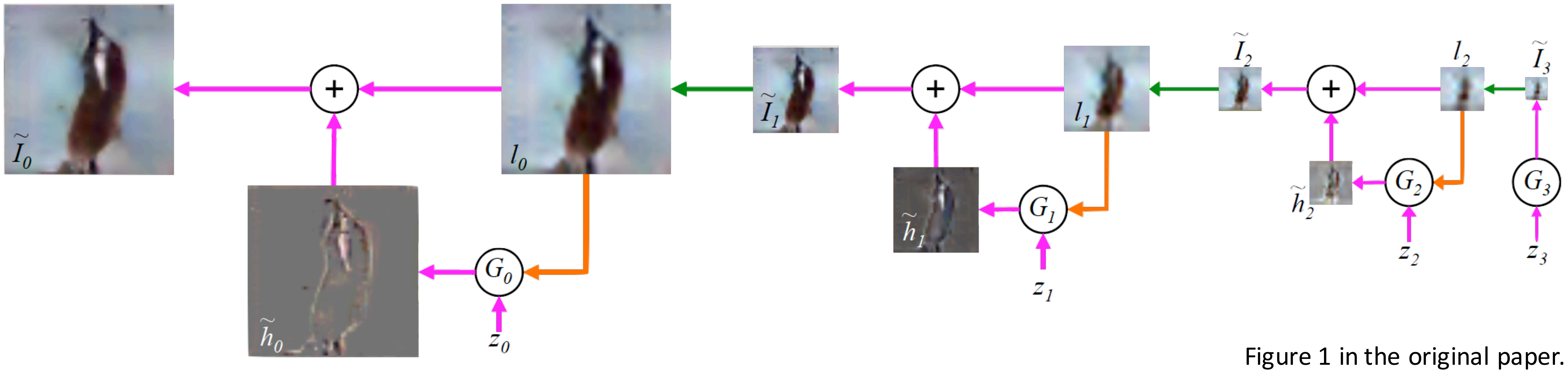


Figure 1 in the original paper.

Image Generation using a LAPGAN

- Generator G_3 generates the base image $\tilde{I}_3$ from random noise input z_3.
- Generators (G_2, G_1, G_0) iteratively generate the *difference image* $(\hat{h})$ conditioned on previous small image (l).
- This *difference image* is added to an up-scaled version of previous smaller image.

Denton, E.L., Chintala, S. and Fergus, R., 2015. "**Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks**". NIPS (2015)

Laplacian Pyramid of Adversarial Networks

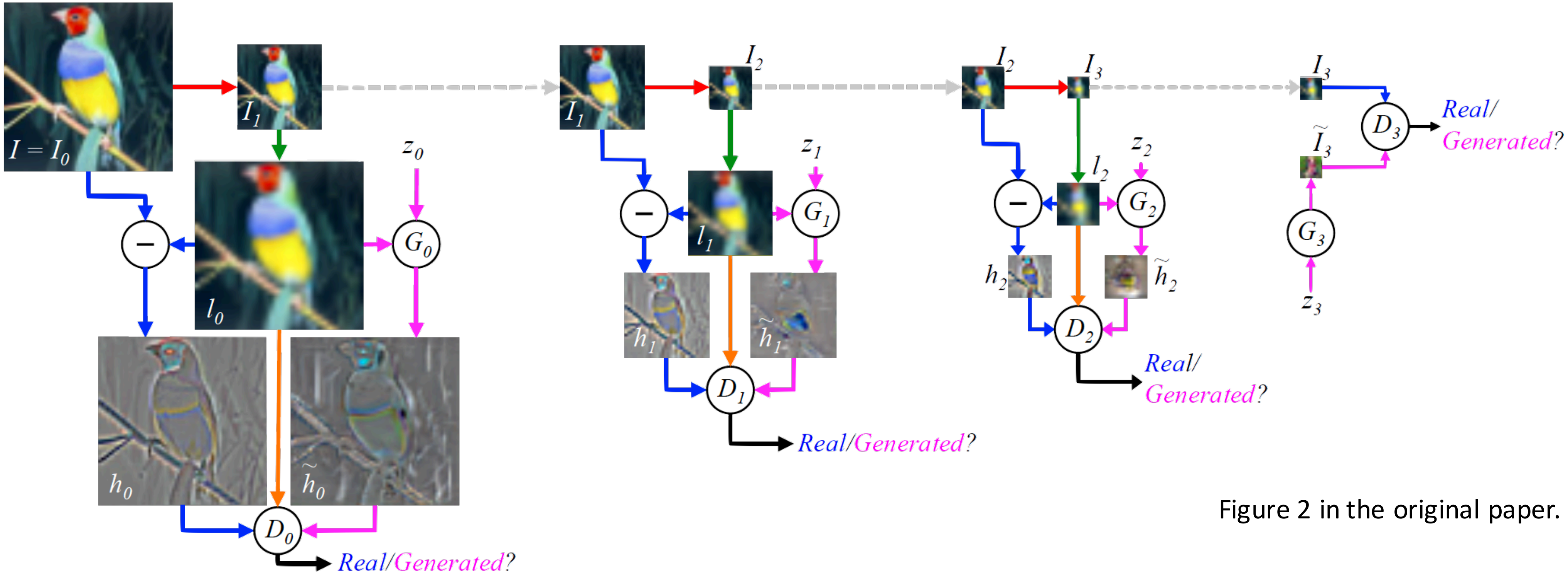


Figure 2 in the original paper.

Training Procedure:

Models at each level are trained independently to learn the required representation.

Denton, E.L., Chintala, S. and Fergus, R., 2015. "**Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks**". NIPS (2015)

Adversarially Learned Inference

- Basic idea is to learn an encoder/inference network along with the generator network.

- Consider the following joint distributions over x (image) and z (latent variables) :

$$q(x,z) = q(x)\, q(z|x)$$ encoder distribution

$$p(x,z) = p(z)\, p(x|z)$$ generator distribution

Dumoulin, Vincent, et al. "**Adversarially learned inference**". *arXiv preprint arXiv:1606.00704* (2016).

Adversarially Learned Inference

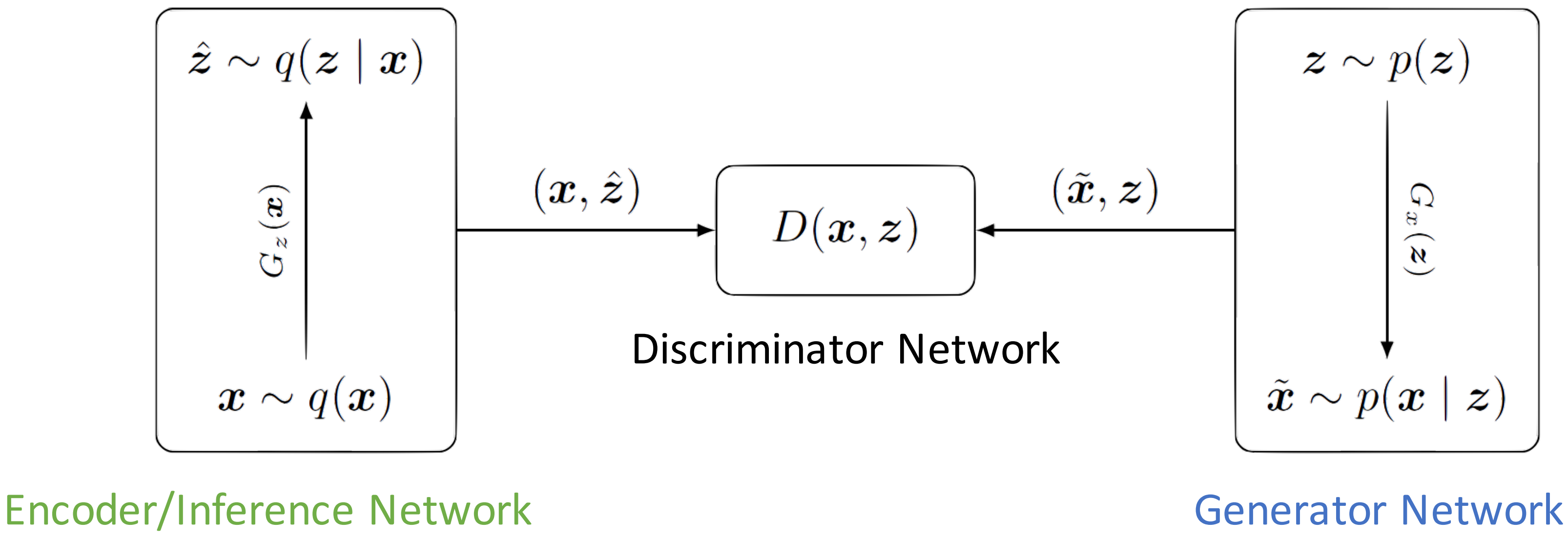


Figure 1 in the original paper.

$$\min_G \max_D \; \mathbb{E}_{q(x)}[\log(D(\underline{x, G_z(x)}))] + \mathbb{E}_{p(x)}[\log(1 - D(\underline{G_x(z), z}))]$$

Dumoulin, Vincent, et al. “**Adversarially learned inference**”. *arXiv preprint arXiv:1606.00704* (2016).

Adversarially Learned Inference

- Nash equilibrium yields

  - **Joint:** $p(x,z) \sim q(x,z)$
  - **Marginals:** $p(x) \sim q(x)$ and $p(z) \sim q(z)$
  - **Conditionals:** $p(x|z) \sim q(x|z)$ and $p(z|x) \sim q(z|x)$

- Inferred latent representation successfully reconstructed the original image.
- Representation was useful in the downstream semi-supervised task.

Dumoulin, Vincent, et al. "**Adversarially learned inference**". *arXiv preprint arXiv:1606.00704* (2016).

Summary

- GANs are generative models that are implemented using two stochastic 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.

Why use GANs for Generation?

- Can be trained using back-propagation for Neural Network based Generator/Discriminator functions.
- Sharper images can be generated.
- Faster to sample from the model distribution: *single* forward pass generates a *single* sample.

Reading List

- 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.

Questions?