2019
怪兽
学堂

GAN

虾米 2019-04

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?

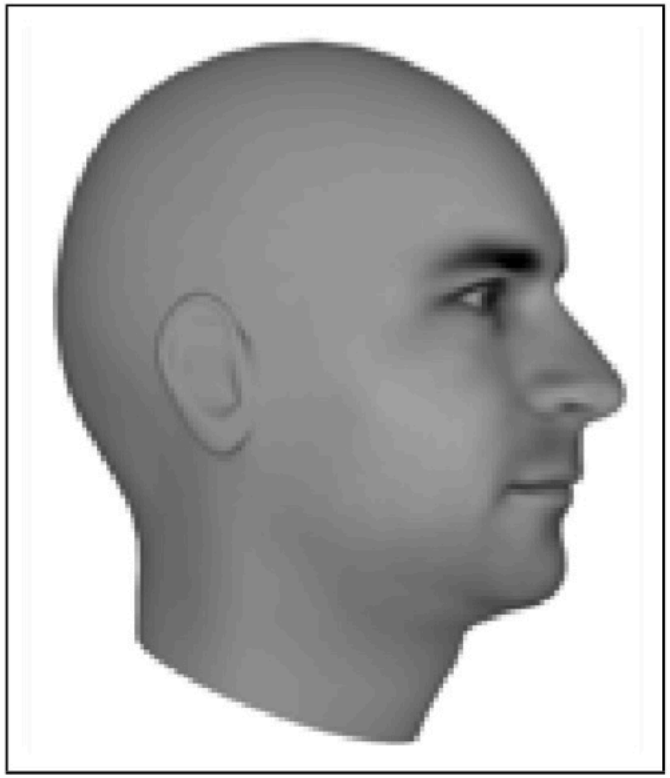
- **We've only seen discriminative models so far**
  - Given an image **X**, predict a label **Y**
  - Estimates **P(Y|X)**

- **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**

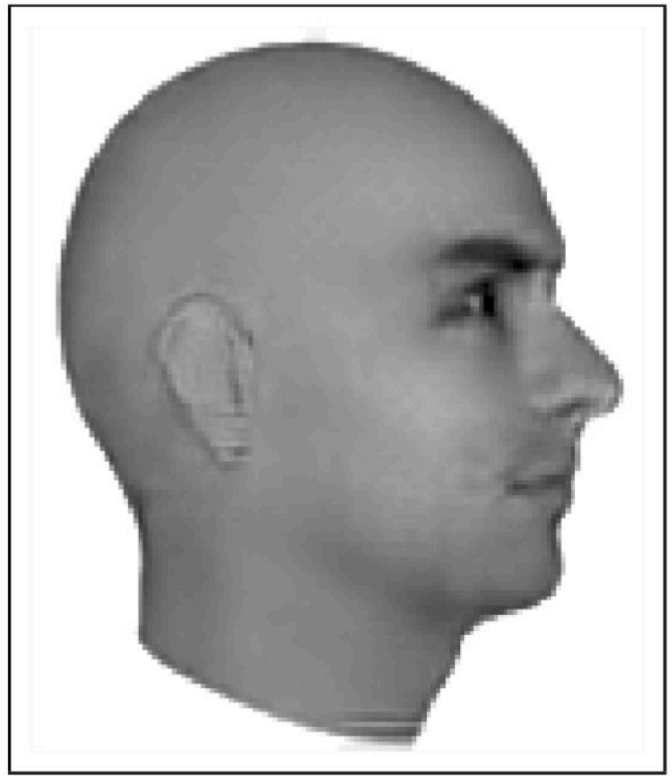
- **Generative models (in general) cope with all of above**
  - Can model **P(X)**
  - Can generate new images

Magic of GANs…

Ground Truth



Adversarial



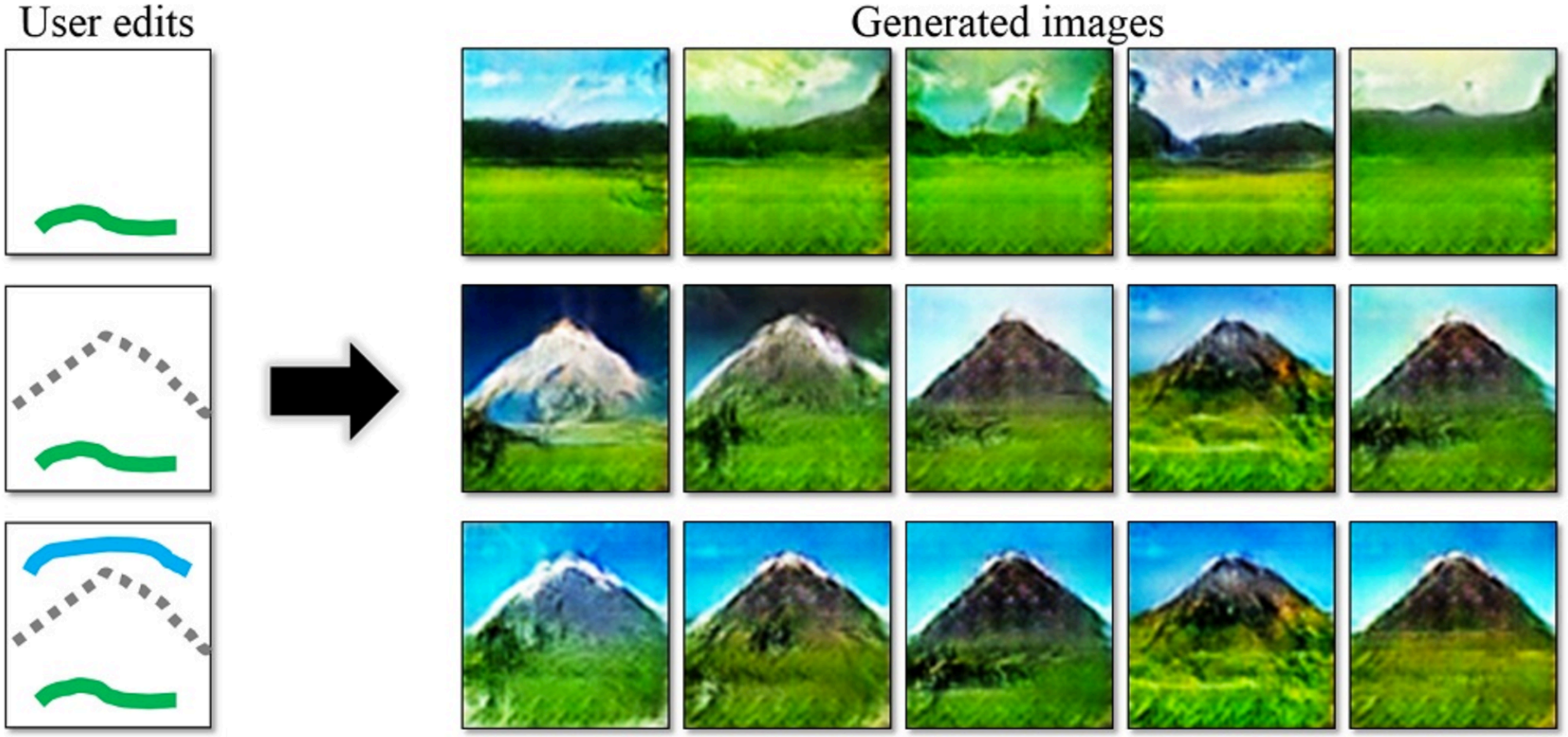
Lotter, William, Gabriel Kreiman, and David Cox. "Unsupervised learning of visual structure using predictive generative networks." *arXiv preprint arXiv:1511.06380* (2015).

Magic of GANs…

Which one is Computer generated?

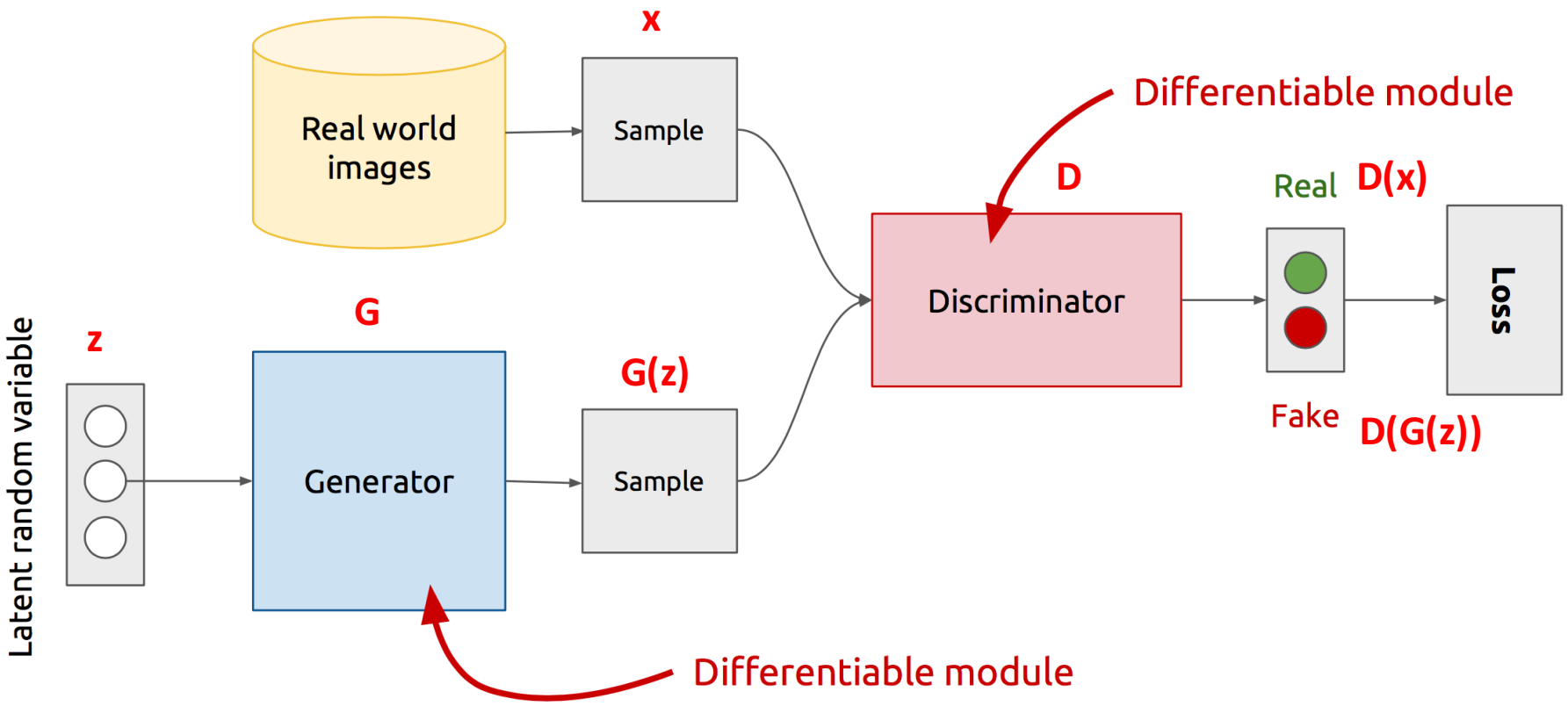
Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *arXiv preprint arXiv:1609.04802* (2016).

Magic of GANs…



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

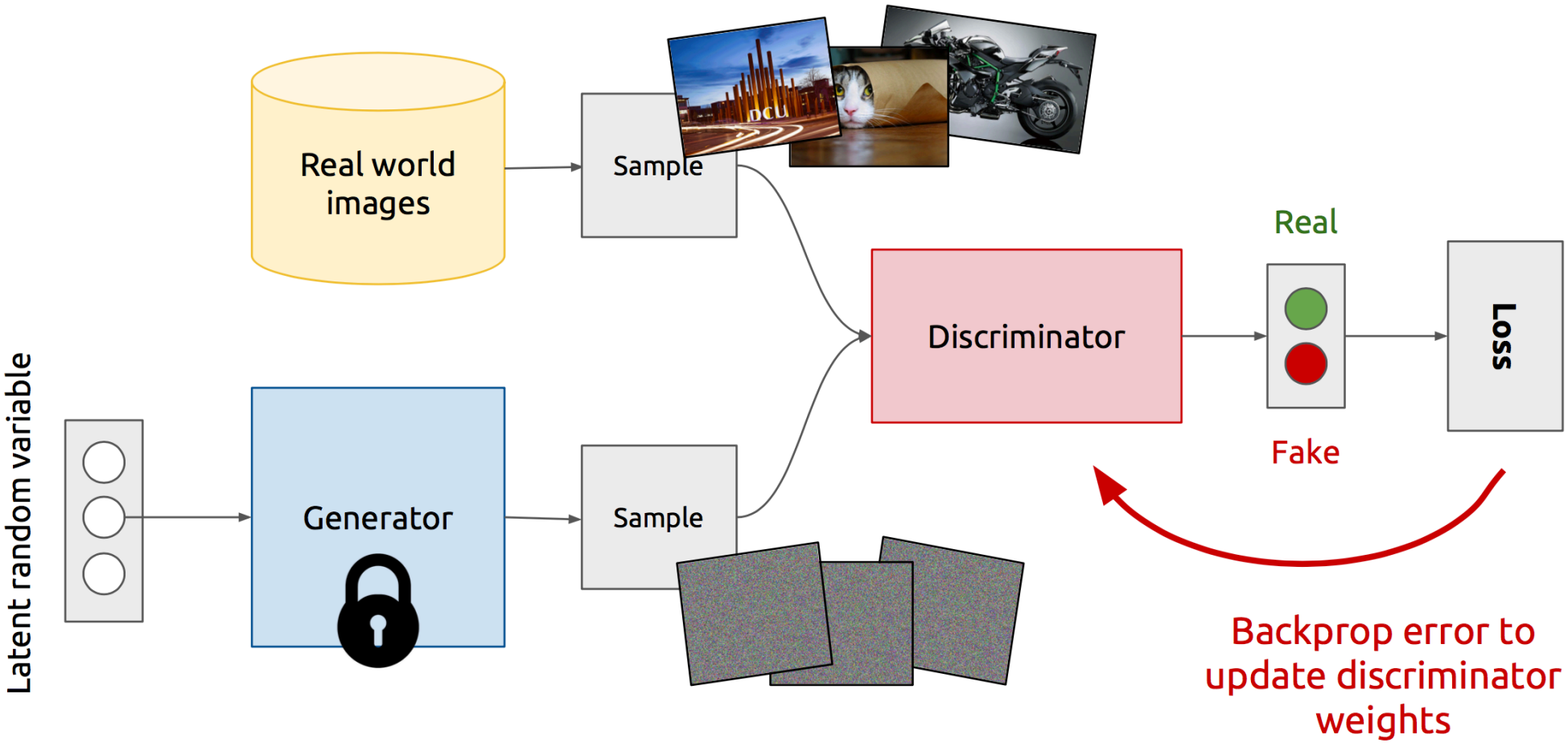
GAN's Architecture



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

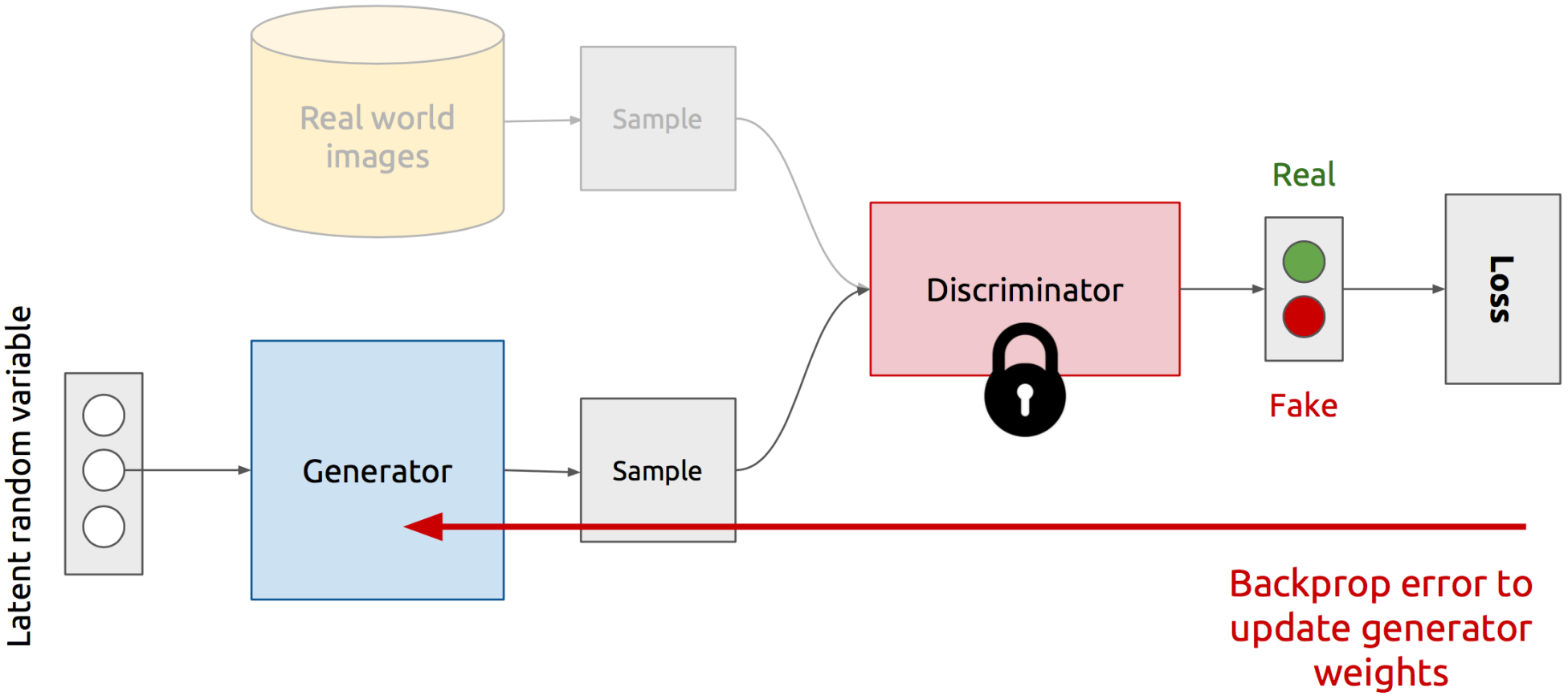
https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

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

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:
  - $P_{data}(x) = P_{gen}(x) \;\; \forall x$
  - $\mathrm{D}(x) = \frac{1}{2} \;\; \forall x$

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 strikes back again…

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

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

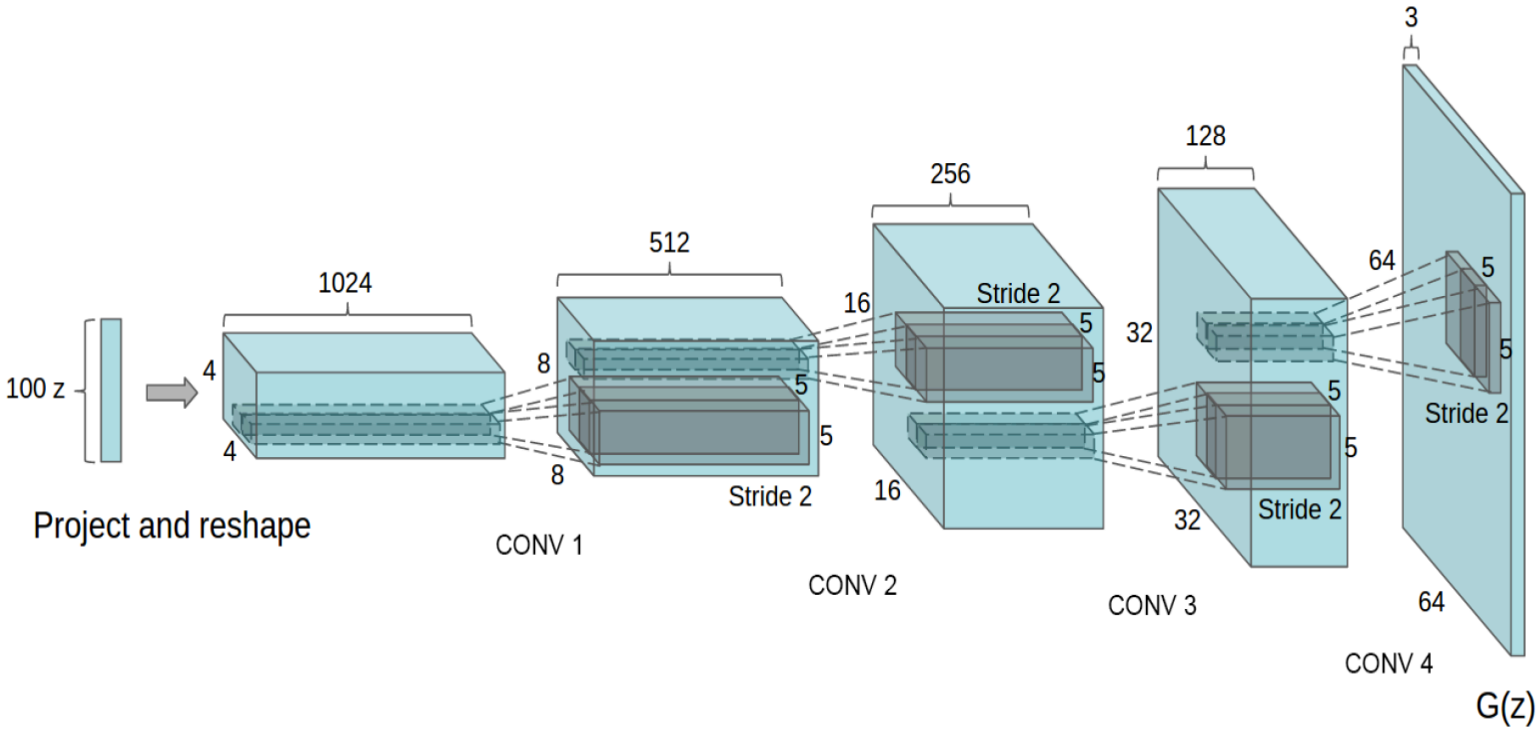
$$\nabla_{\theta_G} V(D,G) = \nabla_{\theta_G} \mathbb{E}_{z\sim q(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

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

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

Deep Convolutional GANs (DCGANs)

Generator Architecture



Key ideas:

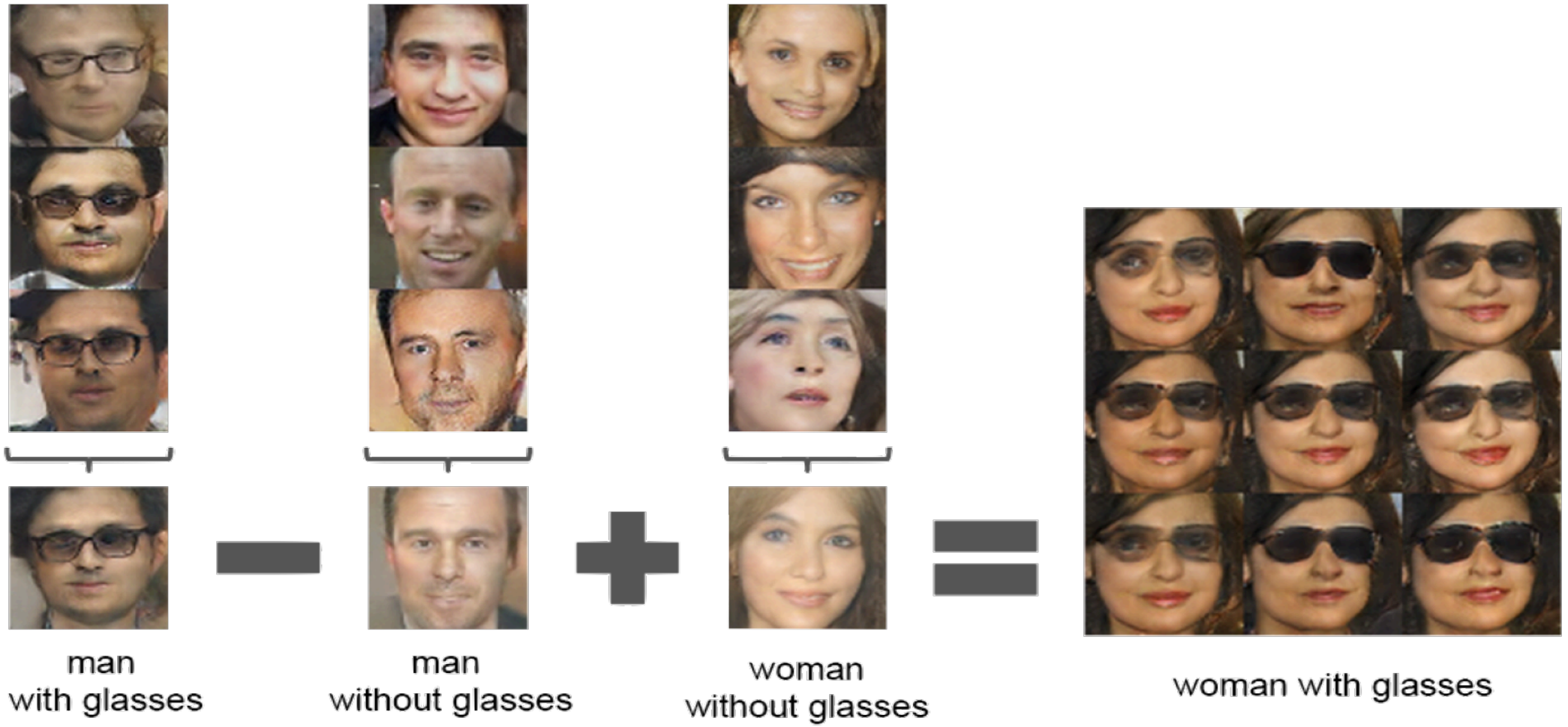
- Replace FC hidden layers with Convolutions
  - **Generator:** Fractional-Strided convolutions
- Use Batch Normalization after each layer
- **Inside Generator**
  - Use ReLU for hidden layers
  - Use Tanh for the output layer

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

DCGAN: Bedroom images

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.

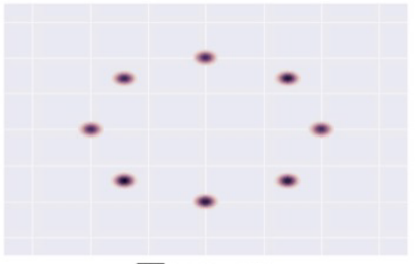
Problems with GANs

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

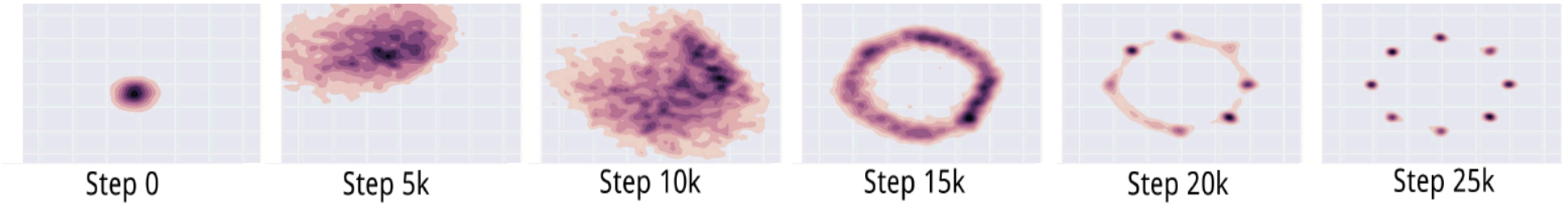
Mode-Collapse

- **Generator fails to output diverse samples**

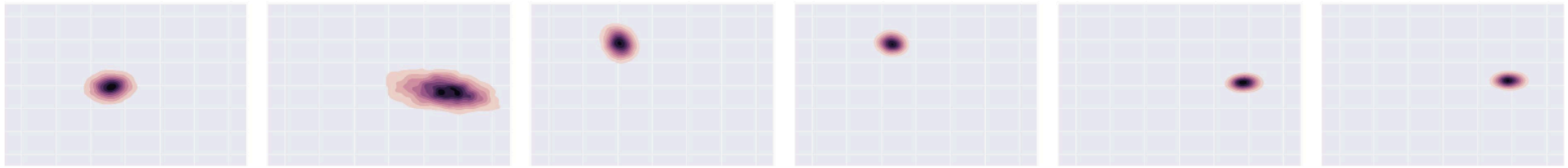
Target



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

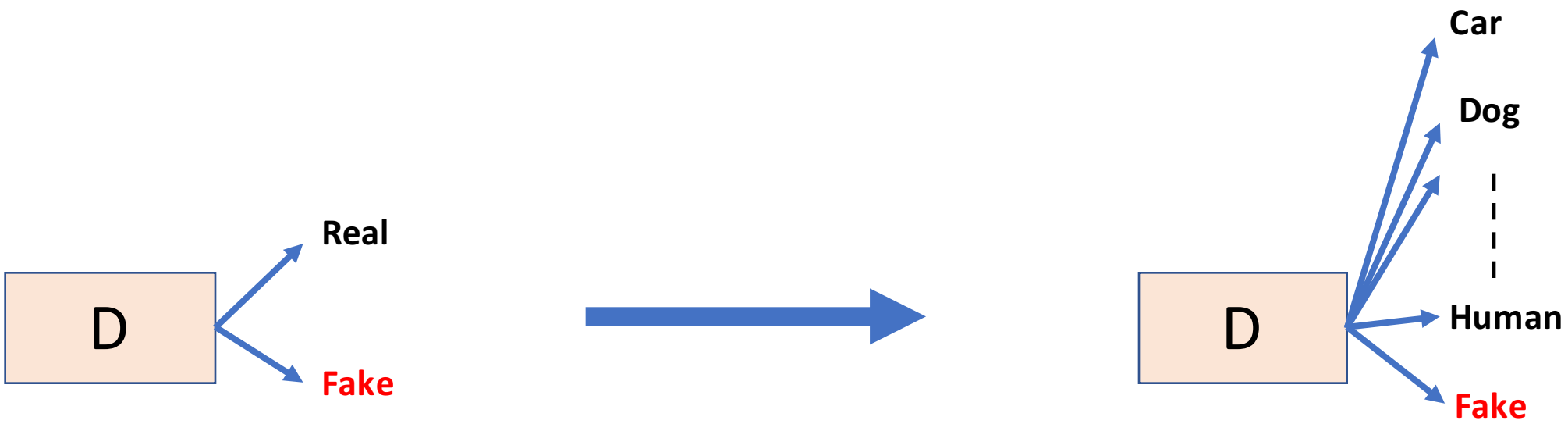
Salimans, Tim, et al. "Improved techniques for training gans." *Advances in Neural Information Processing Systems*. 2016.

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.

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.

$[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ →
$[0, 1, 0, 0, 0, 0, 0, 0, 0, 0]$ →
$[0, 0, 1, 0, 0, 0, 0, 0, 0, 0]$ →
$[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]$ →
$[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]$ →
$[0, 0, 0, 0, 0, 1, 0, 0, 0, 0]$ →
$[0, 0, 0, 0, 0, 0, 1, 0, 0, 0]$ →
$[0, 0, 0, 0, 0, 0, 0, 1, 0, 0]$ →
$[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]$ →
$[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]$ →

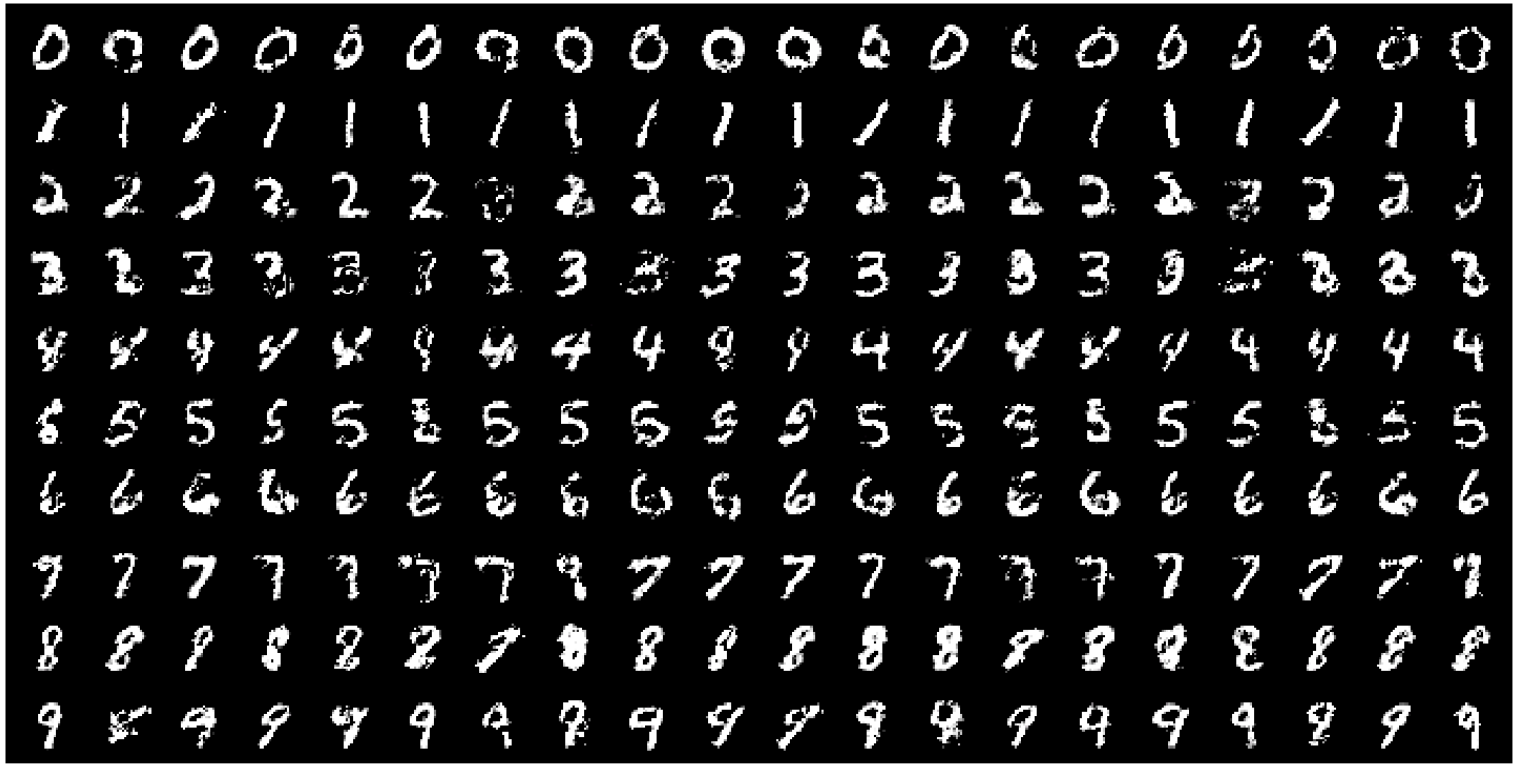


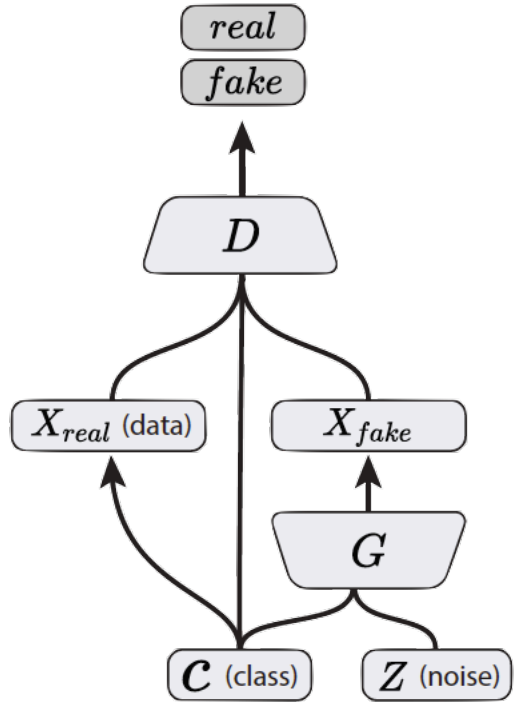
Figure 2 in the original paper.

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

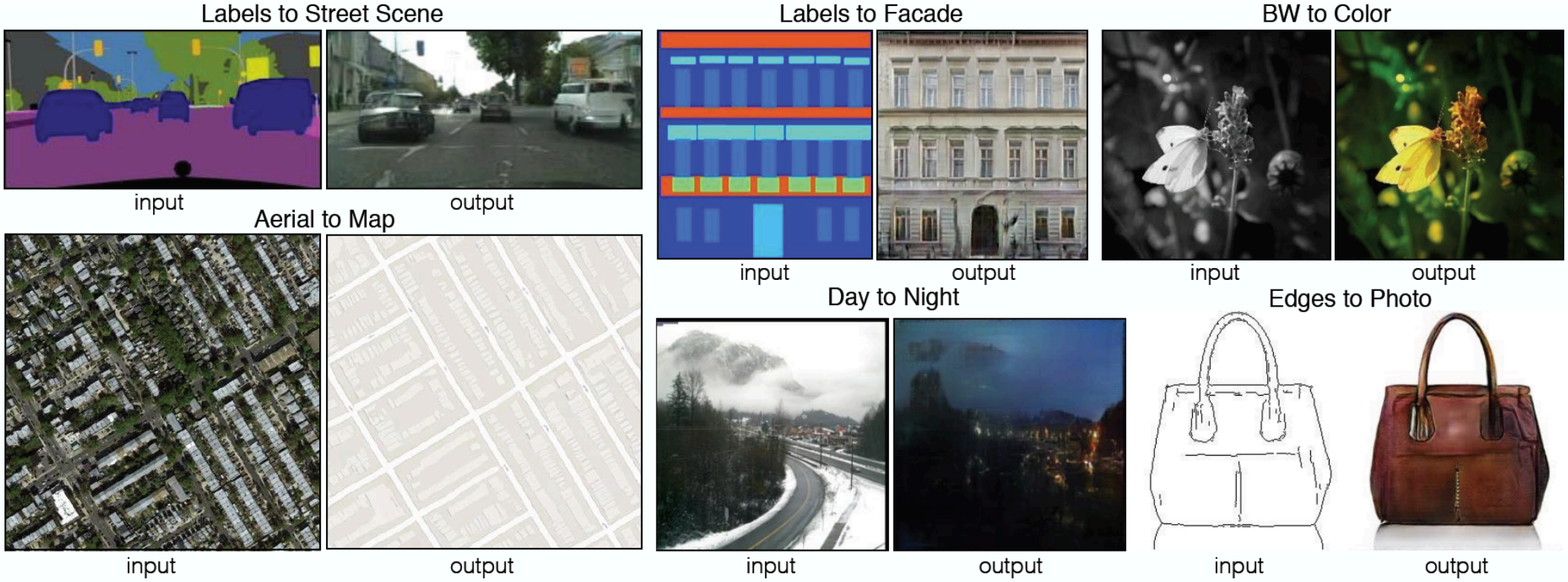


Figure 1 in the original paper.

[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.

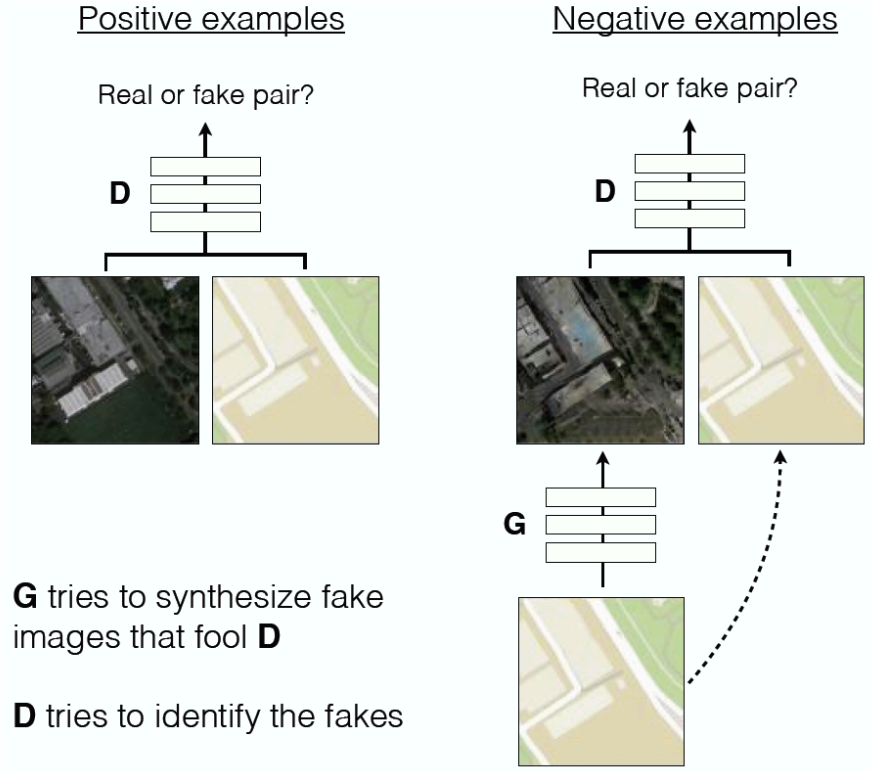


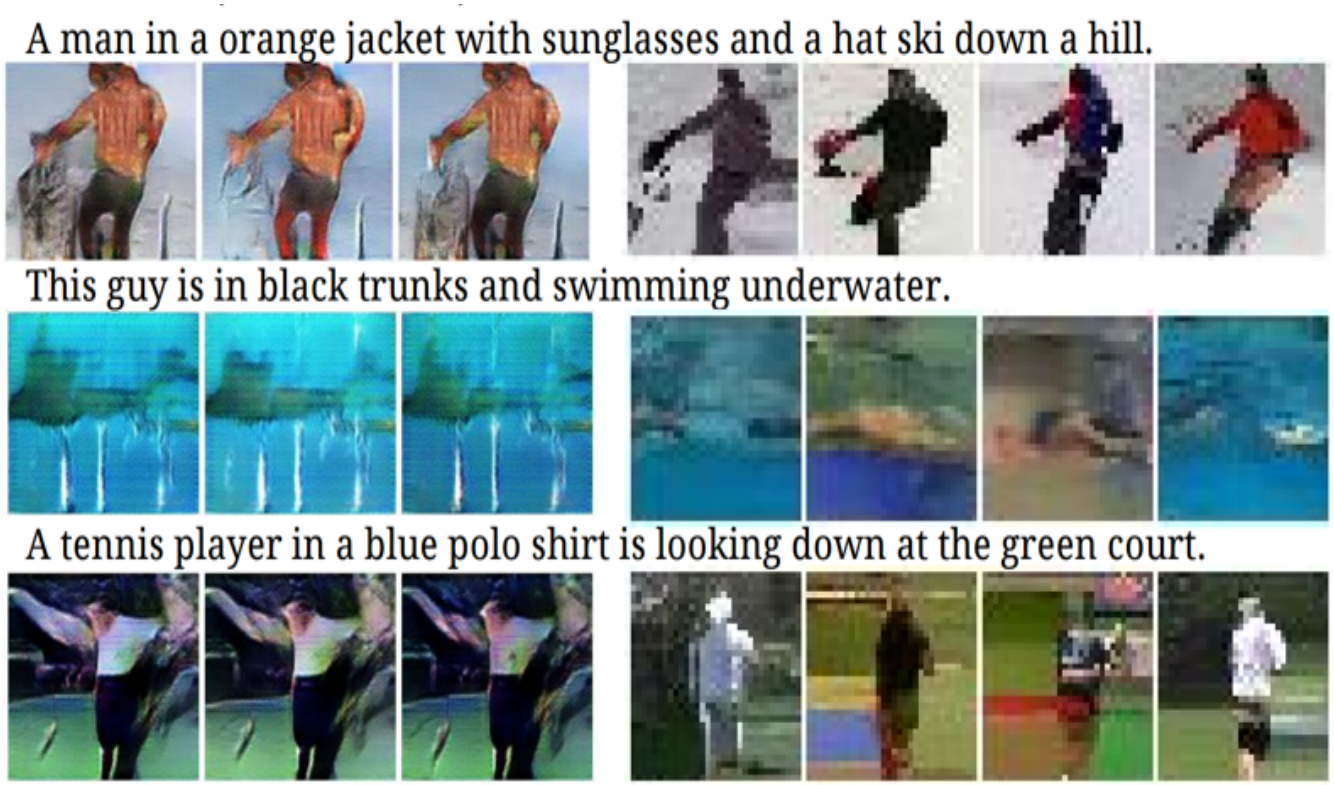
Figure 2 in the original paper.

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

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

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.

补充材料

- https://mp.weixin.qq.com/s?__biz=MzI5ODMzMjk0NA==&mid=100000106&idx=1&sn=6d2440716ad5cffa9535c785bbf71372&chksm=6ca627155bd1ae0302c02e6a34f3437b992e68221c30e83d1fc881795221519 5b4bd815adca6&mpshare=1&scene=1&srcid=#rd

2019
怪兽
学堂

THANKS