

# Generative Models p.2

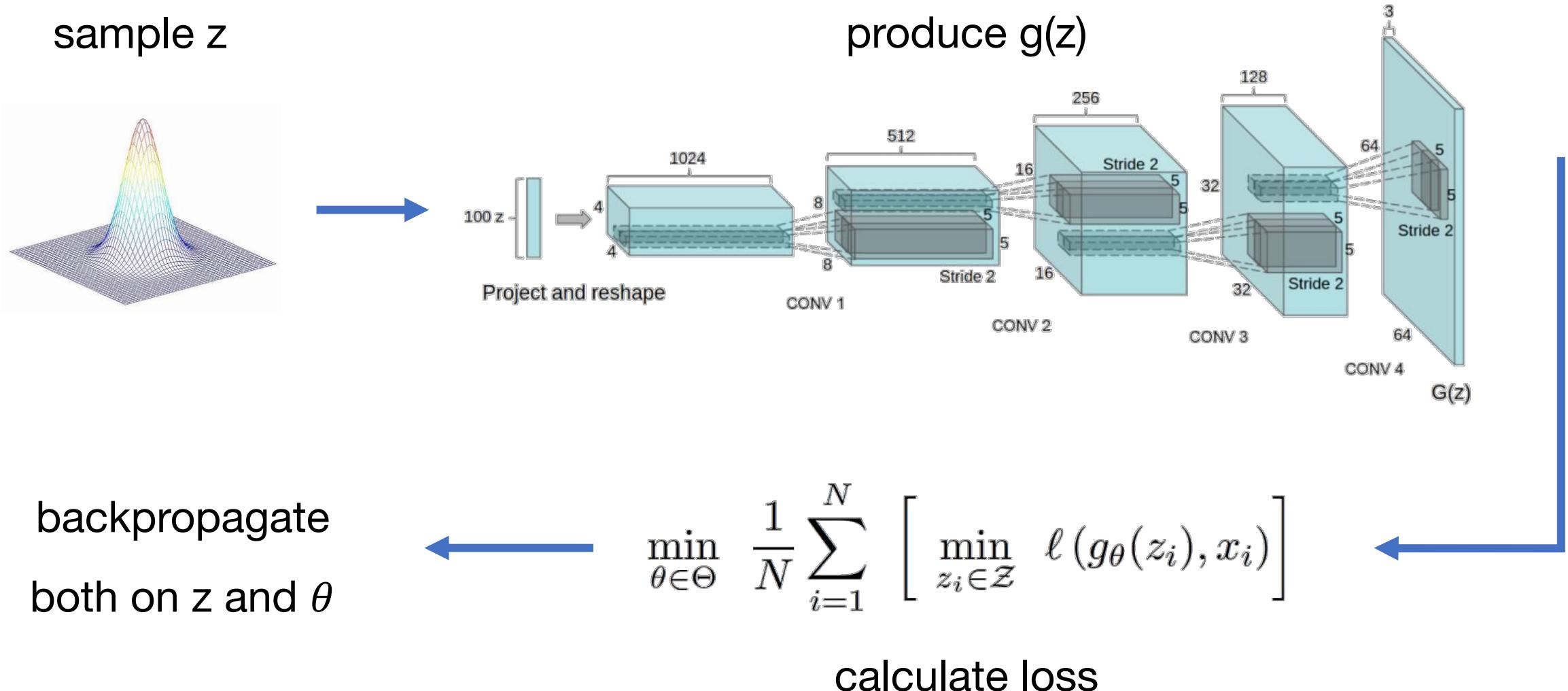
# Outline

---

- Generative Latent Optimization Model (GLO)
- Autoencoders + DeepFake
- Variational Autoencoders (VAE)
- Generative Adversarial Networks (GAN)

# Generative Latent Optimization Model (GLO)

# Generative Latent Optimization Model (GLO)



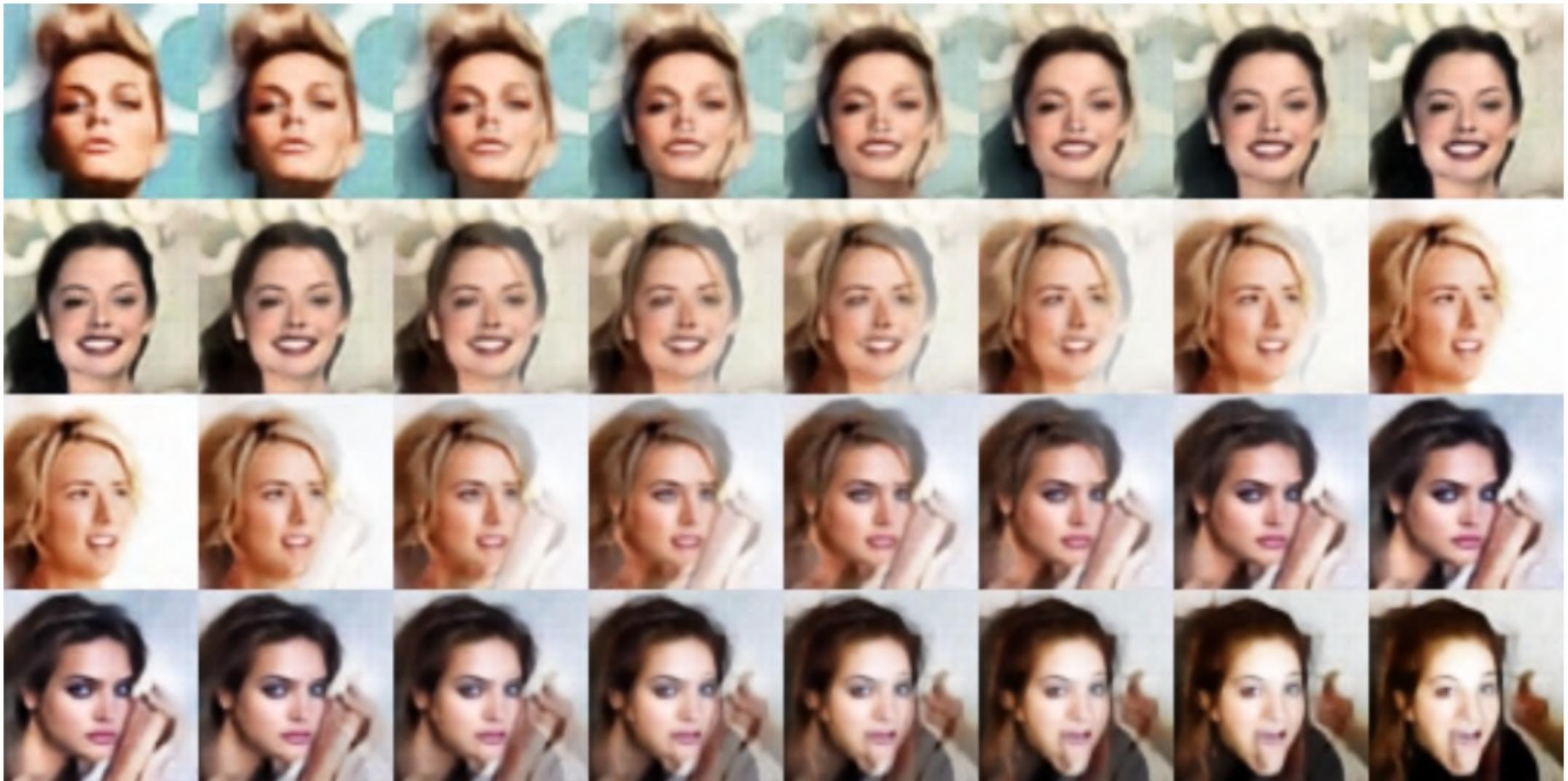
# Generative Latent Optimization Model (GLO)

---



# Generative Latent Optimization Model (GLO)

---



# Generative Latent Optimization Model (GLO)

---

Какие проблемы есть у такого подхода?

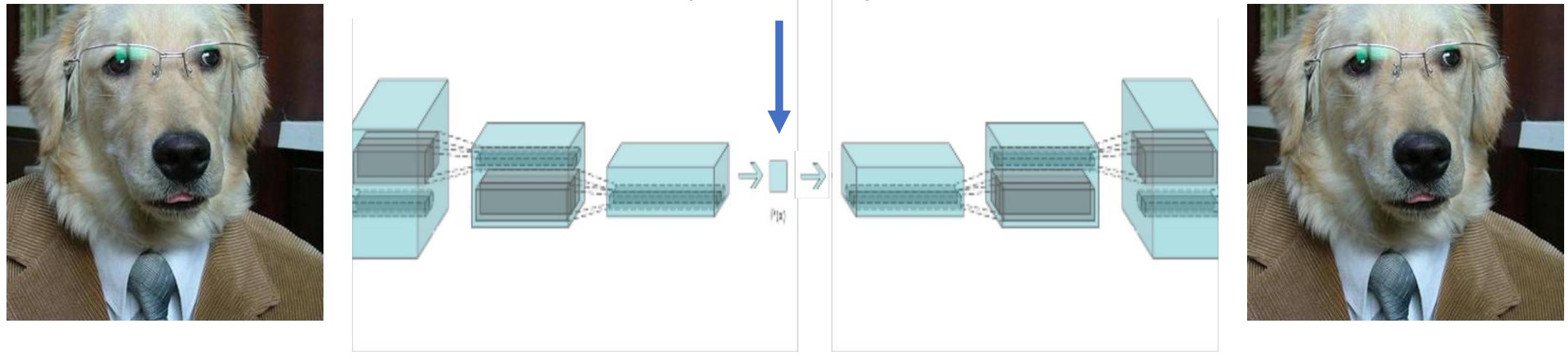
Чтобы получить вектора для новых картинок нам понадобится оптимизация (долго).

Нет гарантии, что распределение останется таким же, как было вначале (не получится качественной генерации).

# Autoencoders

# Autoencoder

---

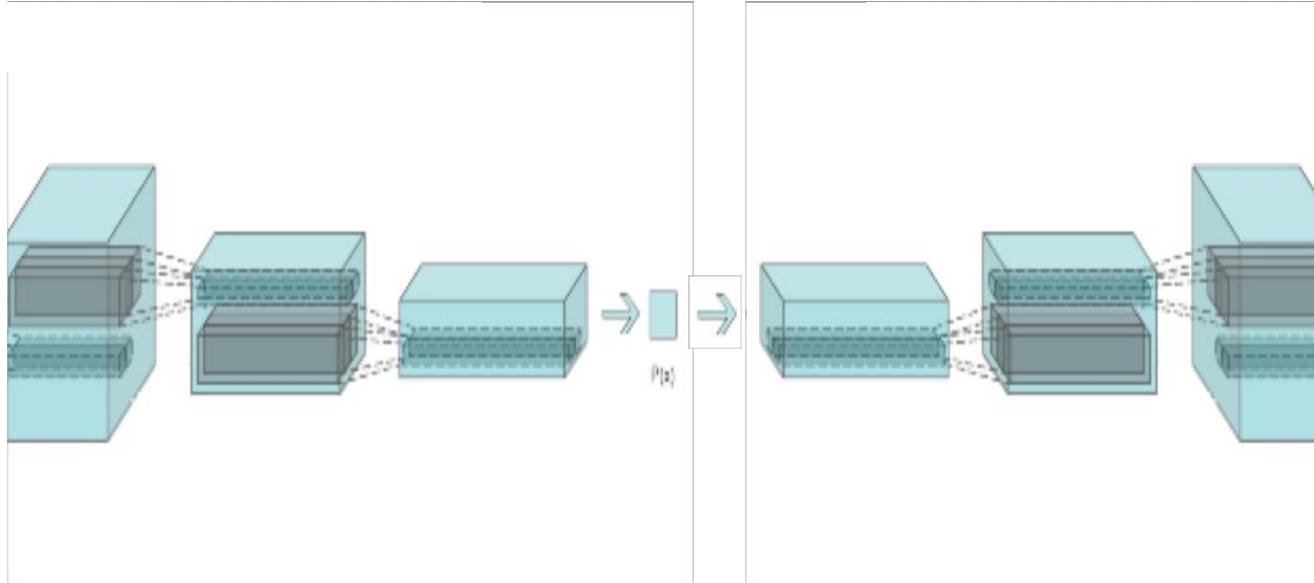


Получить вектора для новых картинок стало просто.

Мы заменили процедуру оптимизации нейронной сетью.

# Autoencoder

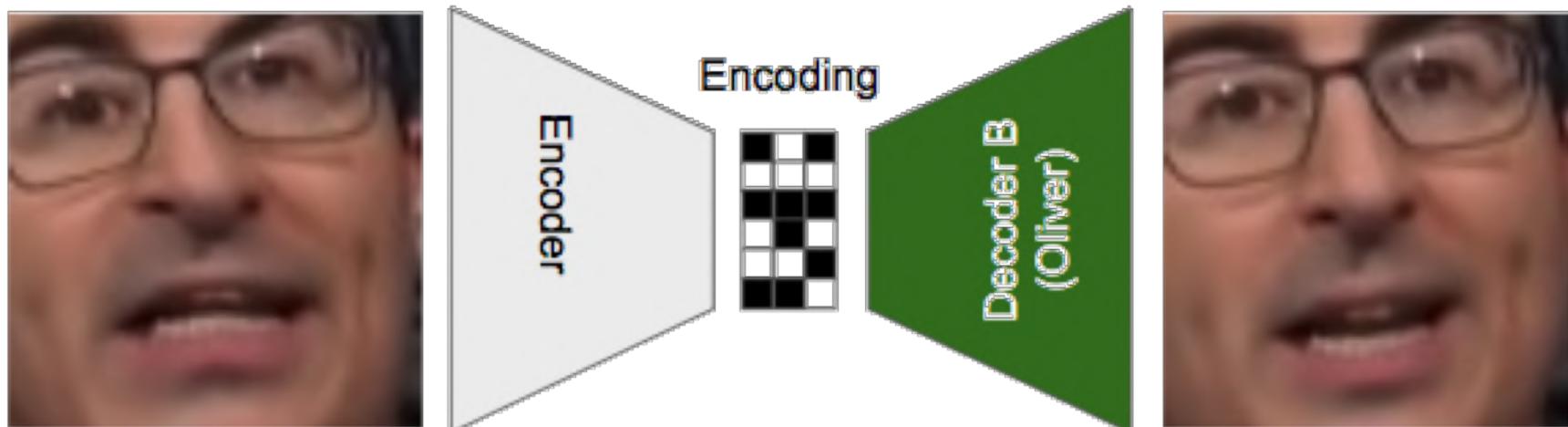
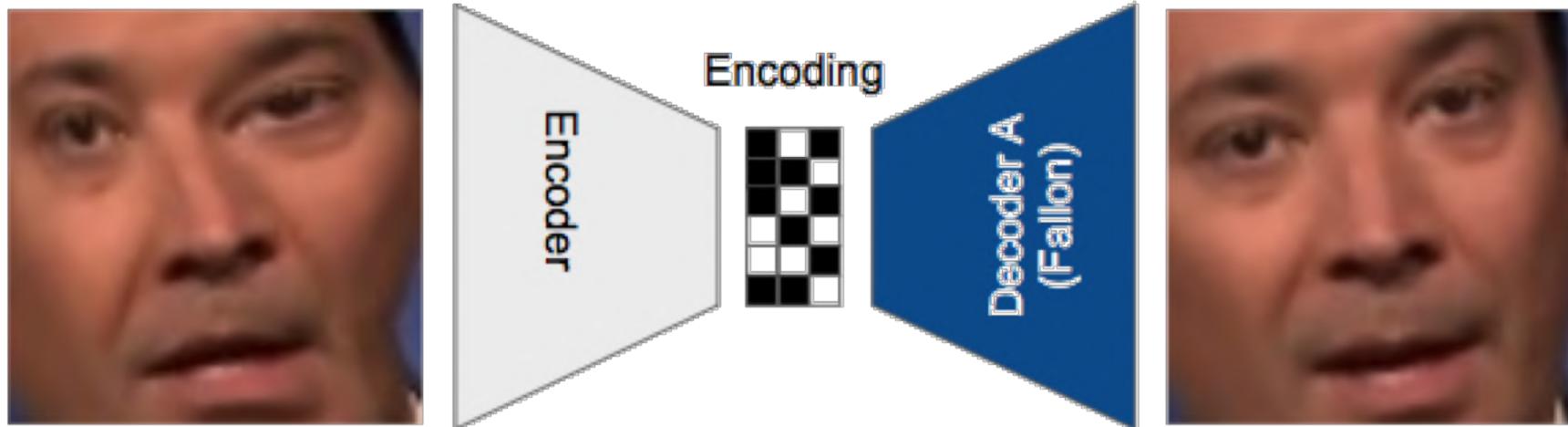
---



MSE  
MAE  
Perceptual Loss  
...

# DeepFake

---



# DeepFake

---



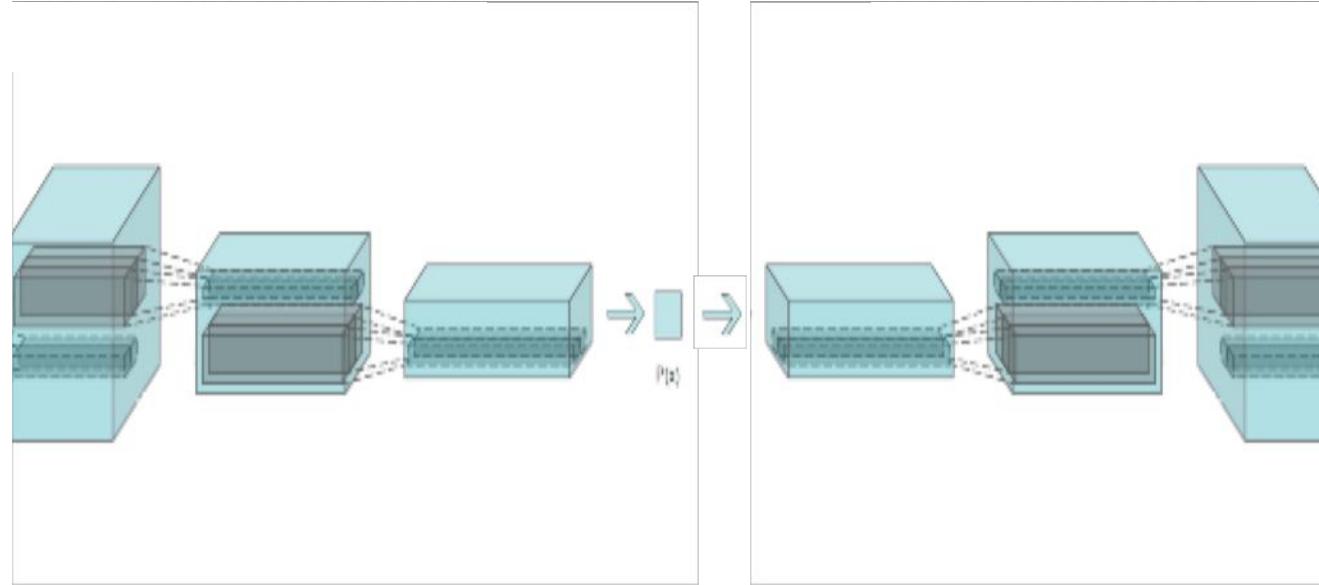
# DeepFake

---



# Autoencoder

---

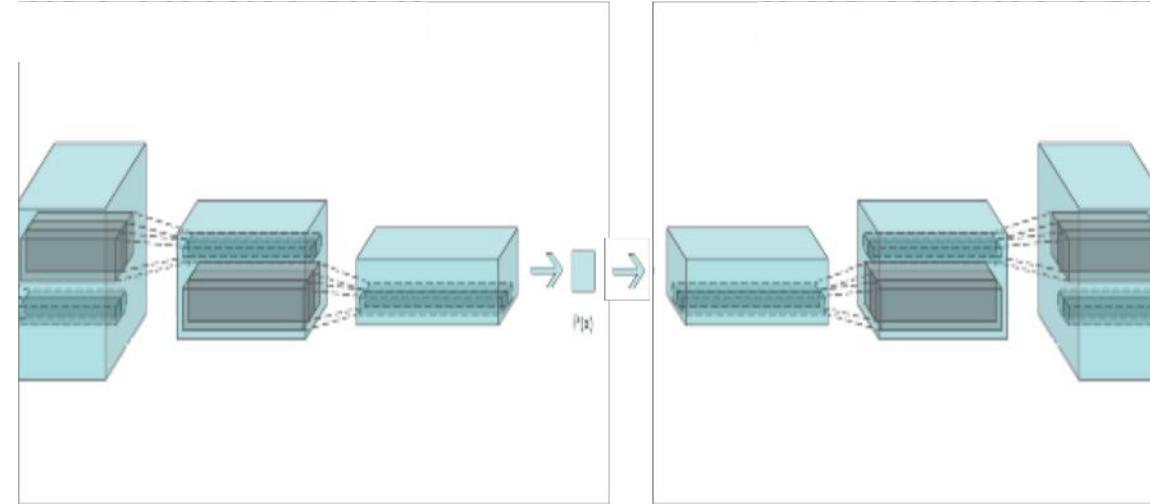


Причем тут underfitting?

Причем тут overfitting?

# Autoencoder

---



Мы смогли избавиться от процесса поиска  
(оптимизации) правильного вектора для картинки.

Каким образом мы будем брать новые вектора для  
того, чтобы генерировать новые изображения?

# Autoencoder

---

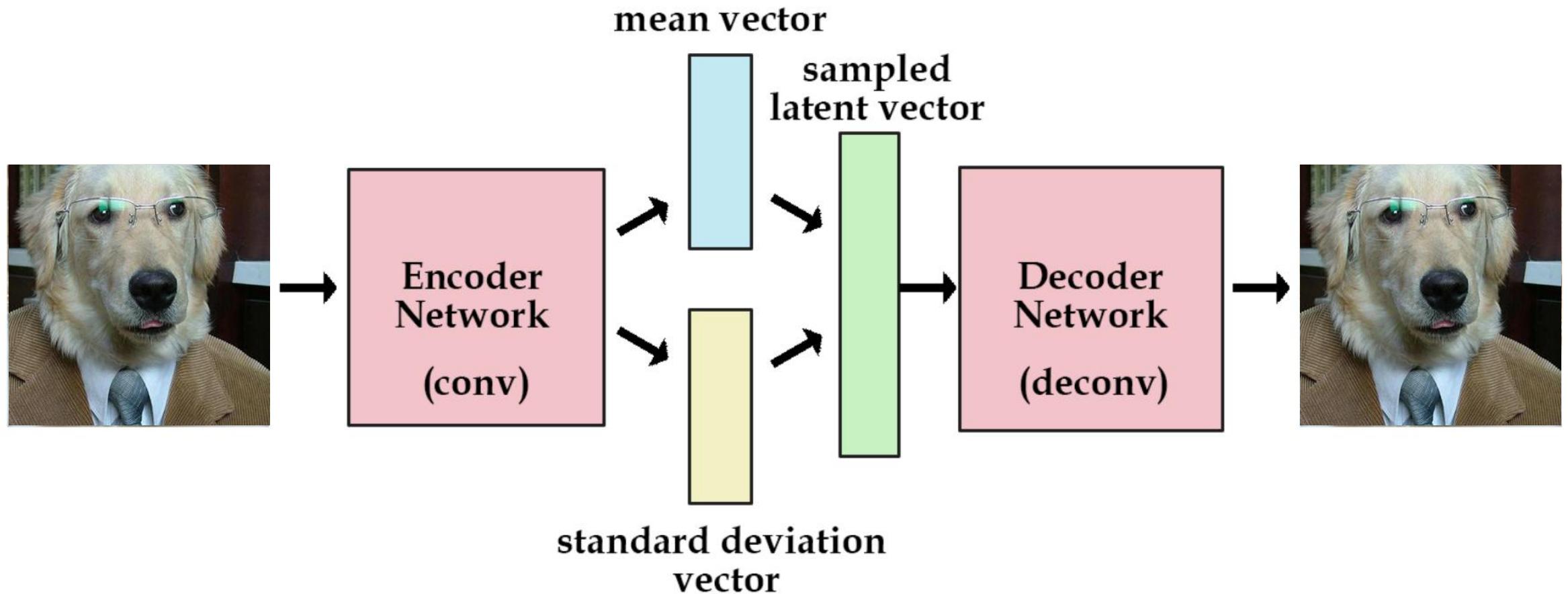
Мы можем попытаться вручную подобрать новые вектора для генерации новых картинок. Но хотелось бы иметь распределение, из которого их можно сэмплировать.

Как гарантированно получить распределение из которого можно будет сэмплировать вектора?

# Variational Autoencoders (VAE)

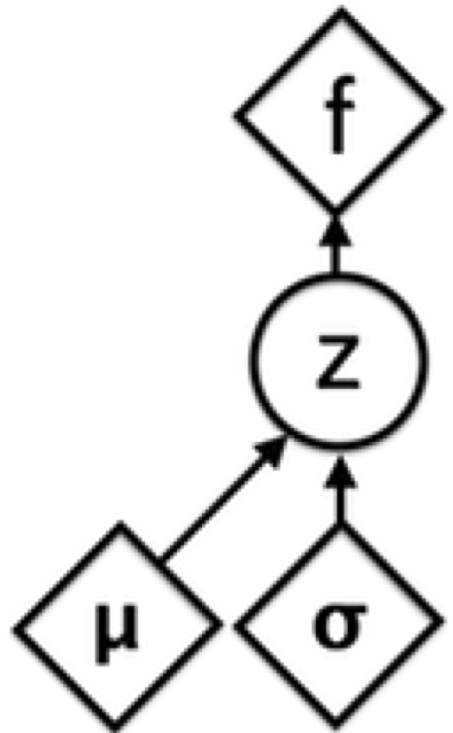
# Variational Autoencoder (VAE)

---

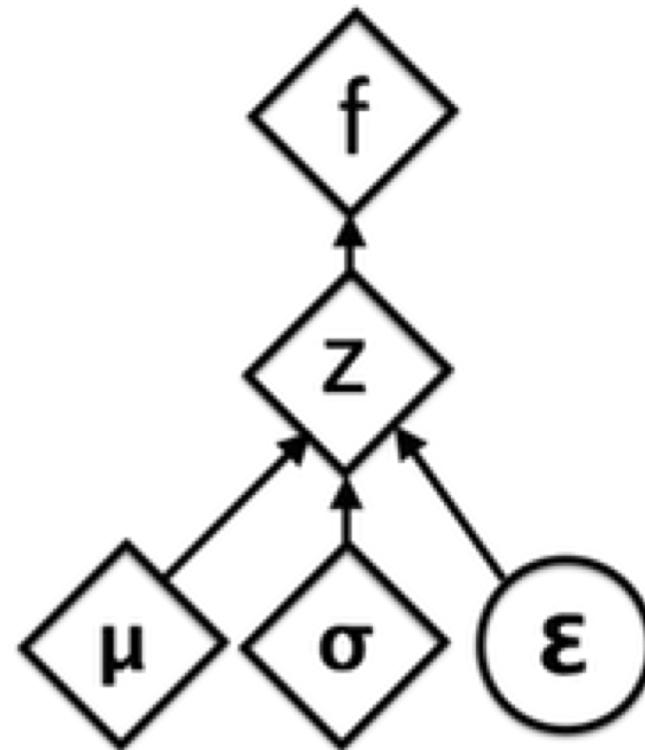


# Variational Autoencoder (VAE)

---



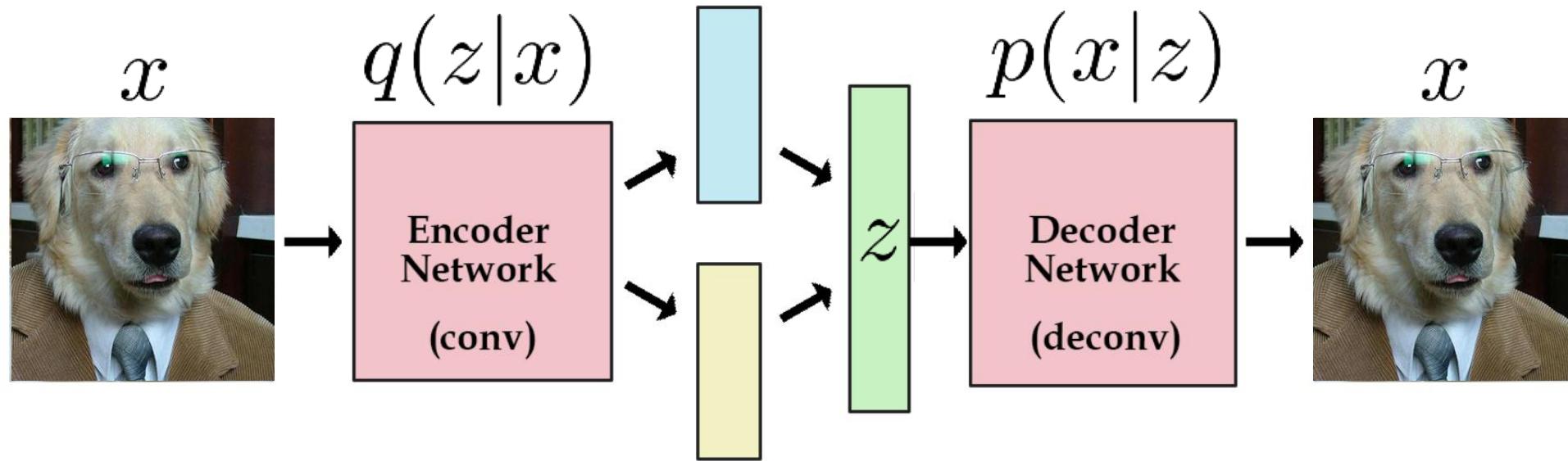
Original



Reparametrized

# Variational Autoencoder (VAE)

---



$$\mathbb{E}_{q(z|x)} \log p(x|z) - \text{KL}(q(z|x) \parallel p(z))$$

# Variational Autoencoder (VAE)

---

$$\mathbb{E}_{q(z|x)} \log p(x|z) - \text{KL}(q(z|x) \| p(z))$$

$$\mathbb{E}_{q_\phi(z|x)} \|d_\theta(z) - x\|^2$$

objective

$$\frac{1}{2} \sum_i (\mu_i^2 + \sigma_j^2 - 1 - \log \sigma_j^2)$$

regularization

# Variational Autoencoder (VAE)

---

$$\min_{\theta, \phi} \left[ \mathbb{E}_{q_\phi(z|x)} \|d_\theta(z) - x\|^2 + \frac{1}{2} \sum_i (\mu_i^2 + \sigma_j^2 - 1 - \log \sigma_j^2) \right]$$

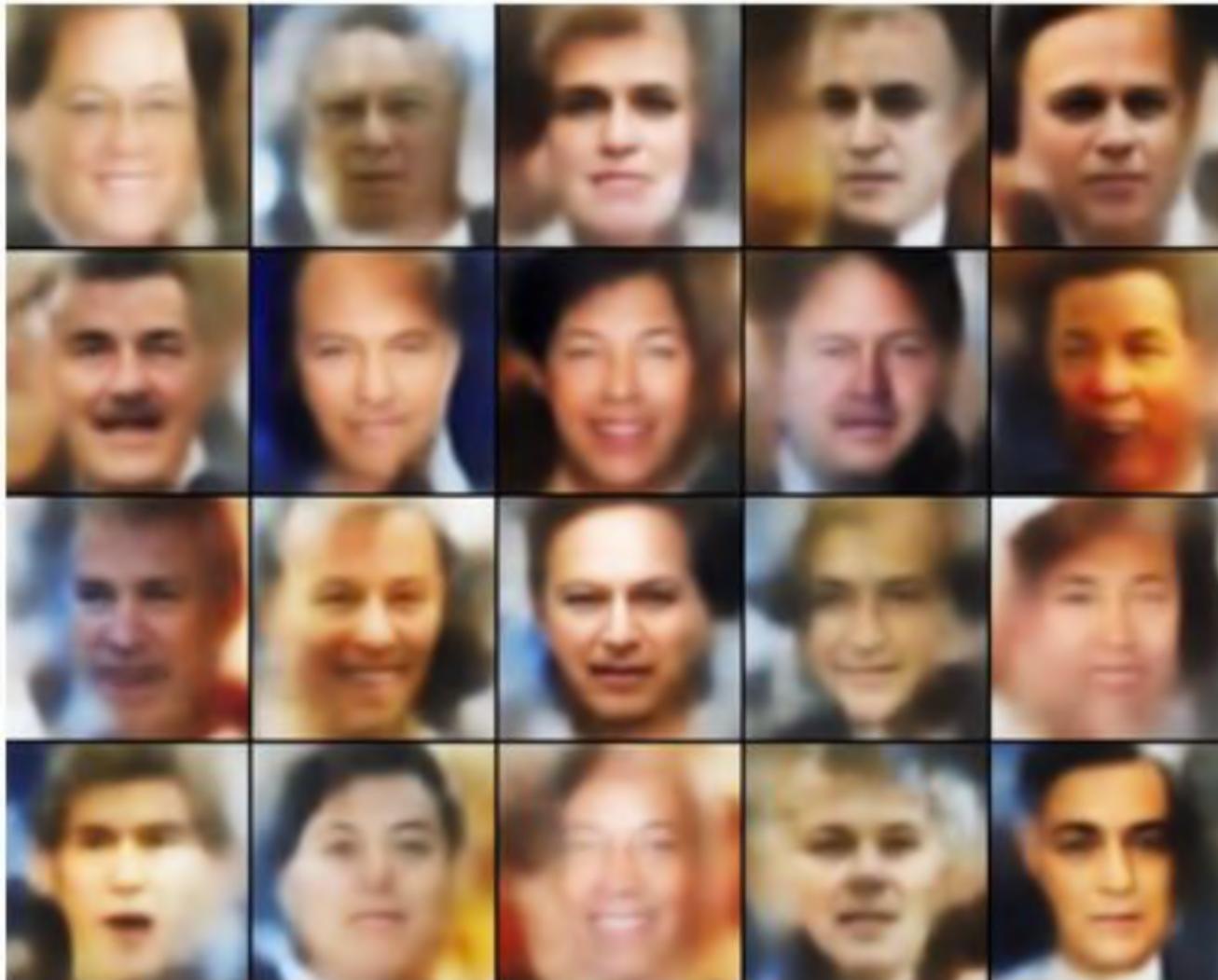
# Variational Autoencoder (VAE)

---

6 6 6 6 6 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
4 4 4 4 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
4 4 2 2 2 2 2 2 3 5 5 5 0 0 0 0 0 0 0 0 0 0 0 0  
4 4 2 2 2 2 2 2 3 3 5 5 5 5 0 0 0 0 0 0 0 0 0 0  
4 4 2 2 2 2 2 2 3 3 3 5 5 5 5 5 0 0 0 0 0 0 0 0  
4 4 2 2 2 2 2 2 3 3 3 3 5 5 5 5 5 5 5 5 5 5 5 5  
4 4 4 4 2 2 2 2 3 3 3 3 3 5 5 5 5 5 5 5 5 5 5 5  
4 4 4 4 4 2 2 2 3 3 3 3 3 3 5 5 5 5 5 5 5 5 5 5  
4 4 4 4 4 4 2 2 3 3 3 3 3 3 3 5 5 5 5 5 5 5 5 5  
4 4 4 4 4 4 4 2 3 3 3 3 3 3 3 3 5 5 5 5 5 5 5 5  
4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 5 5 5 5 5 5 5  
4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 5 5 5 5 5 5  
4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 5 5 5 5 5  
4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 5 5 5 5  
4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 5 5 5  
4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 5 5  
4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 5  
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 5

# Variational Autoencoder (VAE)

---



# Variational Autoencoder (VAE)

---

Нам по прежнему не нужно искать вектор для каждой картинки путем оптимизации.

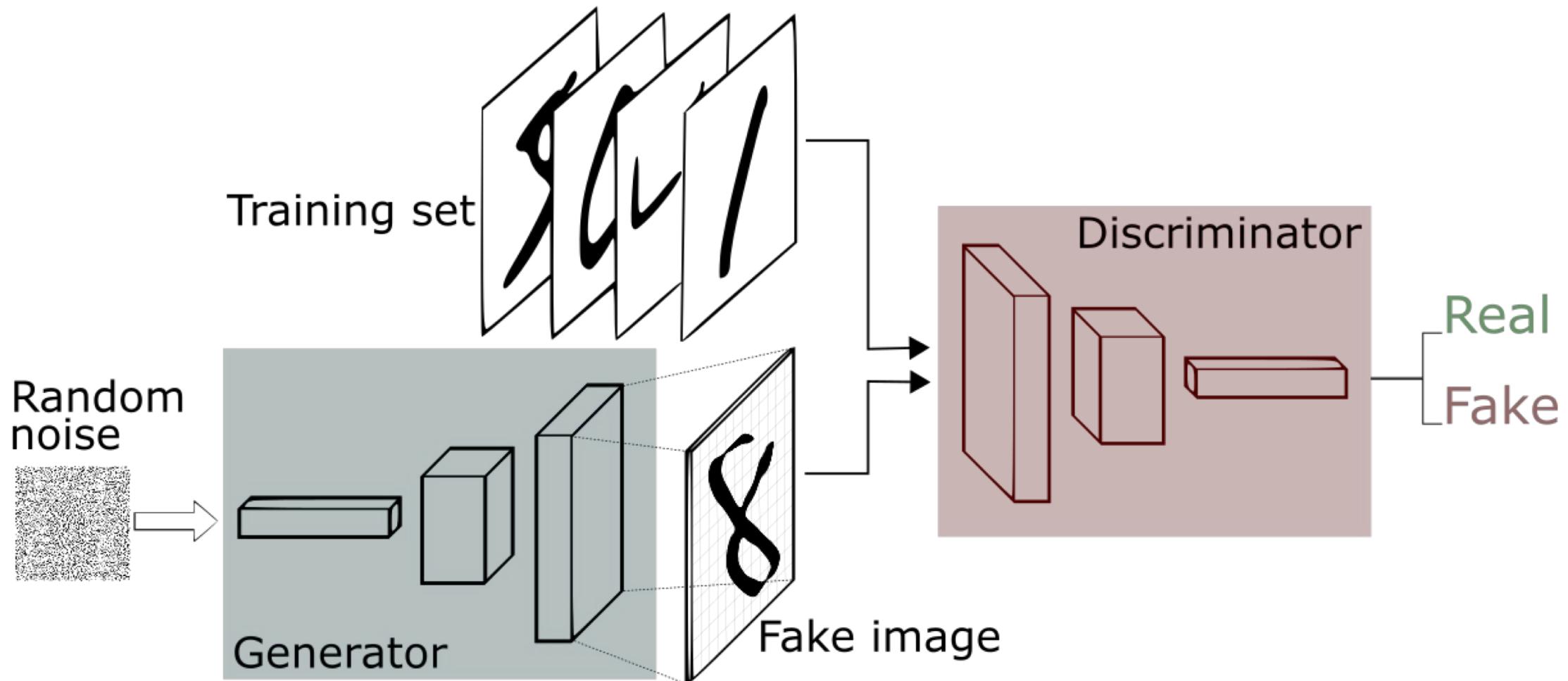
Теперь, чтобы генерировать картинки мы можем просто сэмплировать вектора из нашего априорного распределения  $p(z)$ .

Что, если мы попробуем решить задачу не напрямую, а опосредованно?

# Generative Adversarial Networks

# Generative Adversarial Network

---



# Generative Adversarial Network

---

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



Решение  
дискриминатора для  
реальной картинки

Решение  
дискриминатора для  
картинки, которую  
сделал генератор

Проблема: если дискриминатор сильный, то генератор ничему не учится.

# Generative Adversarial Network

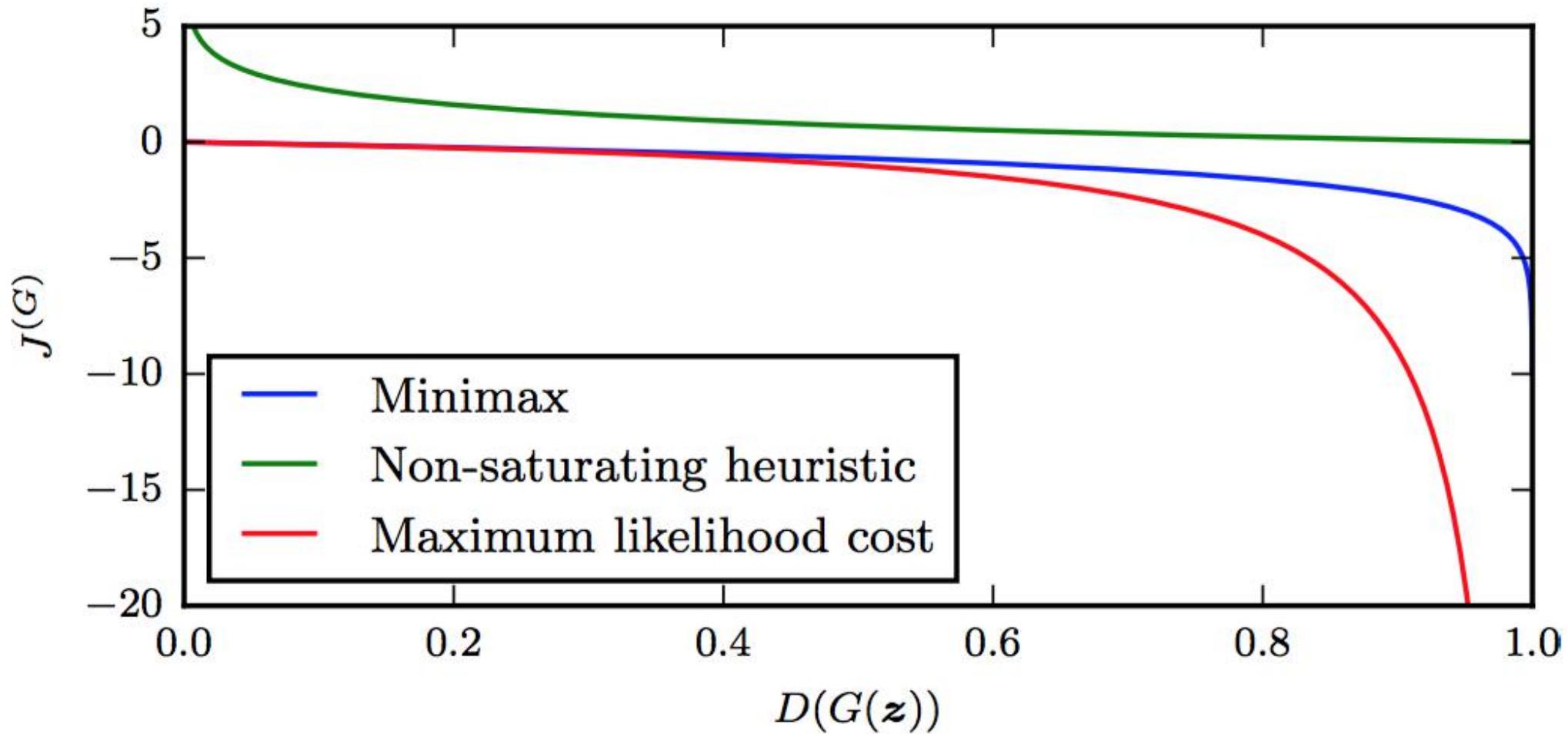
---

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

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

# Generative Adversarial Network

---



# Generative Adversarial Network

---

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

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

Более стабильный вариант обучения.

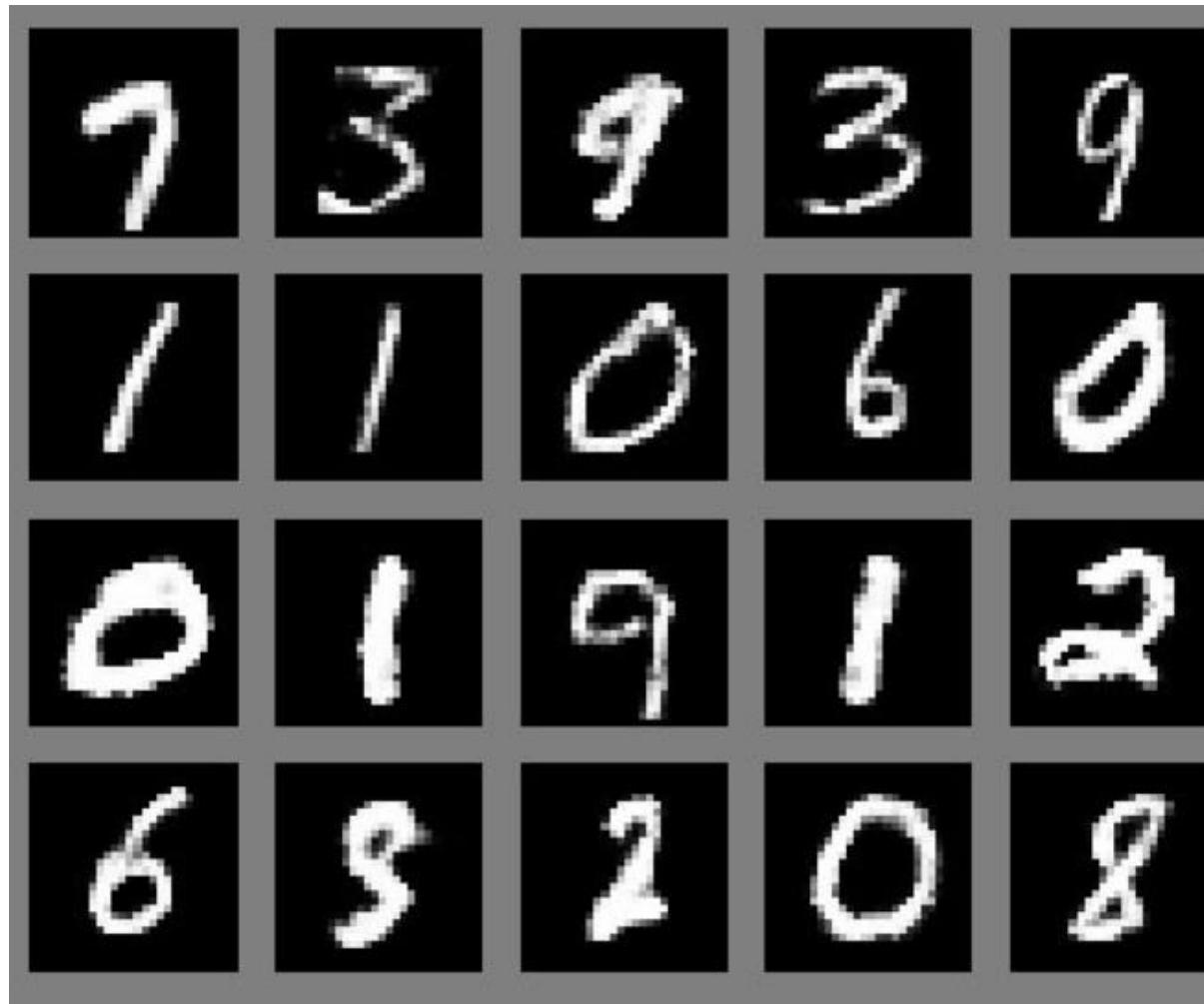
# Generative Adversarial Network

---



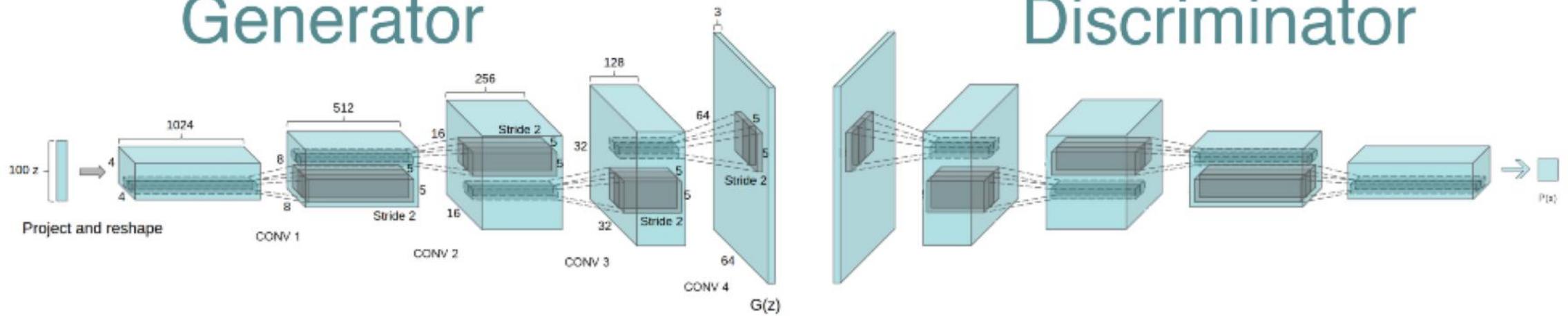
# Generative Adversarial Network

---



# DCGAN

## Generator



## Discriminator

# DCGAN

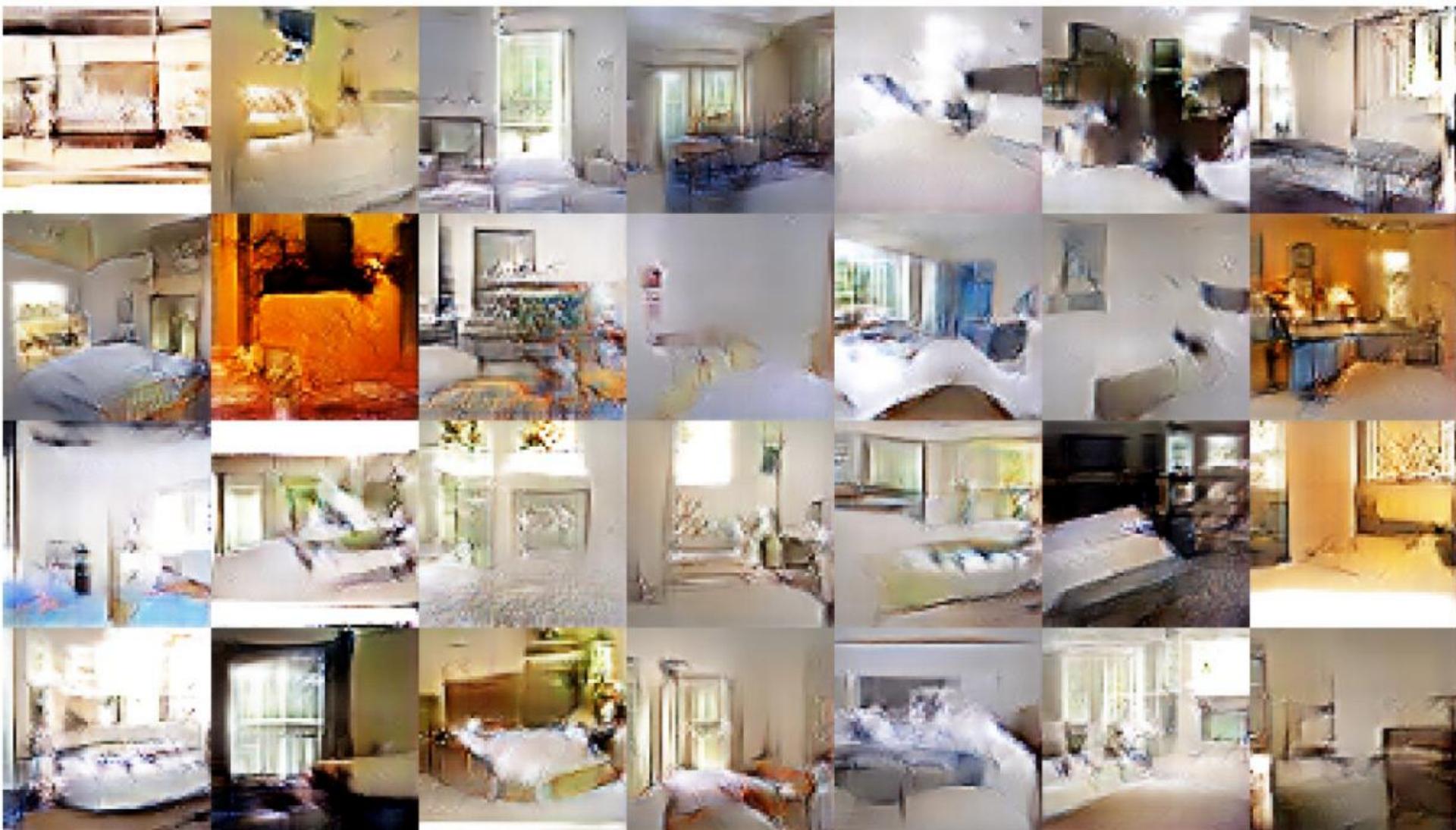
---

## Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

# DCGAN

---



# DCGAN

---



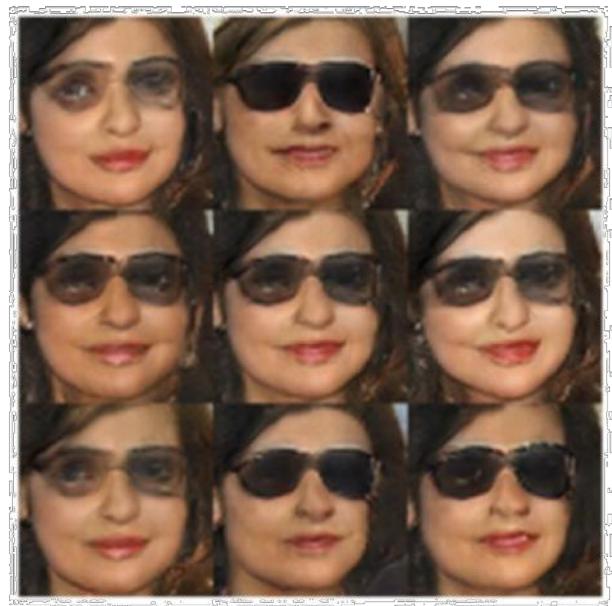
man  
with glasses



man  
without glasses

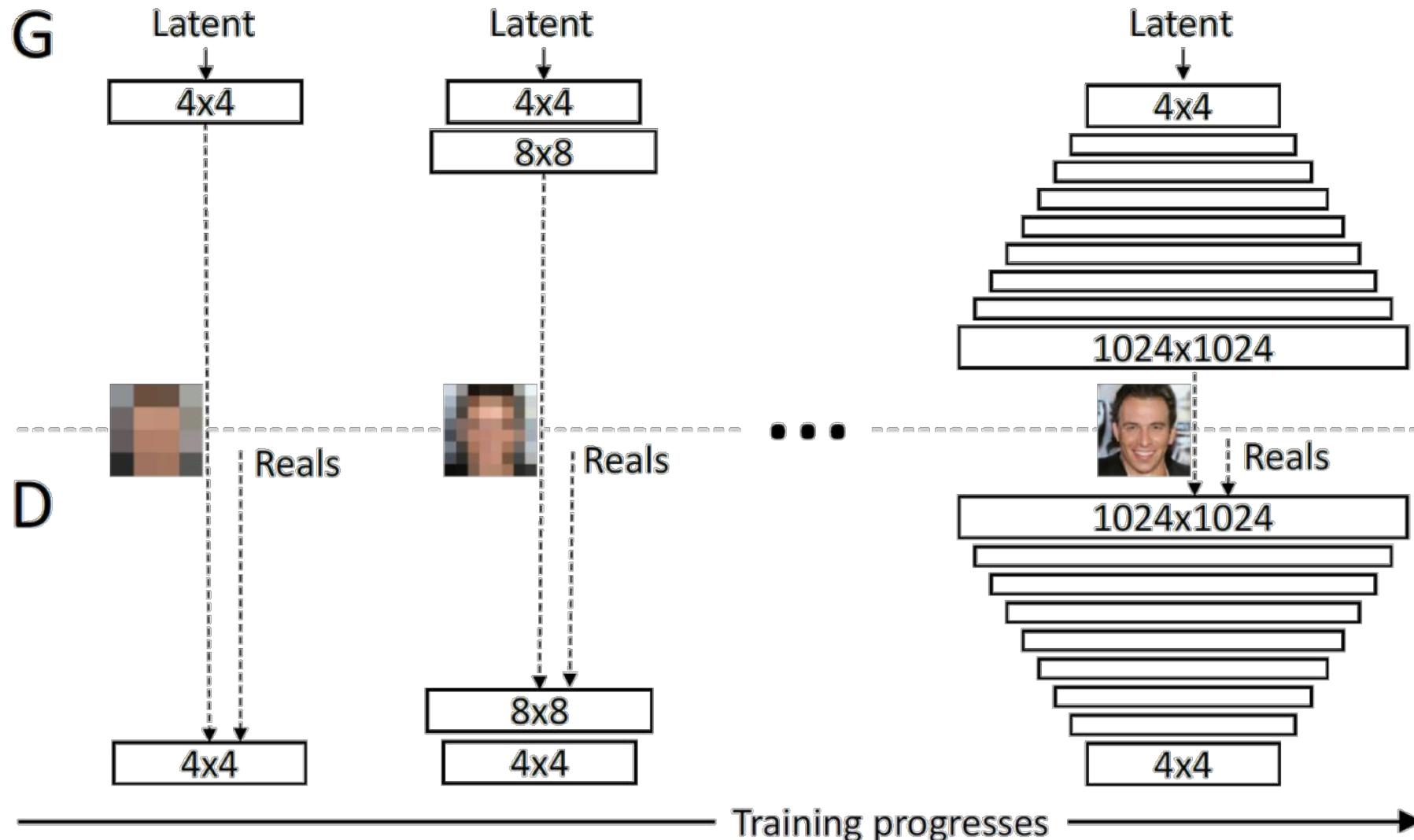


woman  
without glasses



woman  
with glasses

# Progressive Growing of GANs

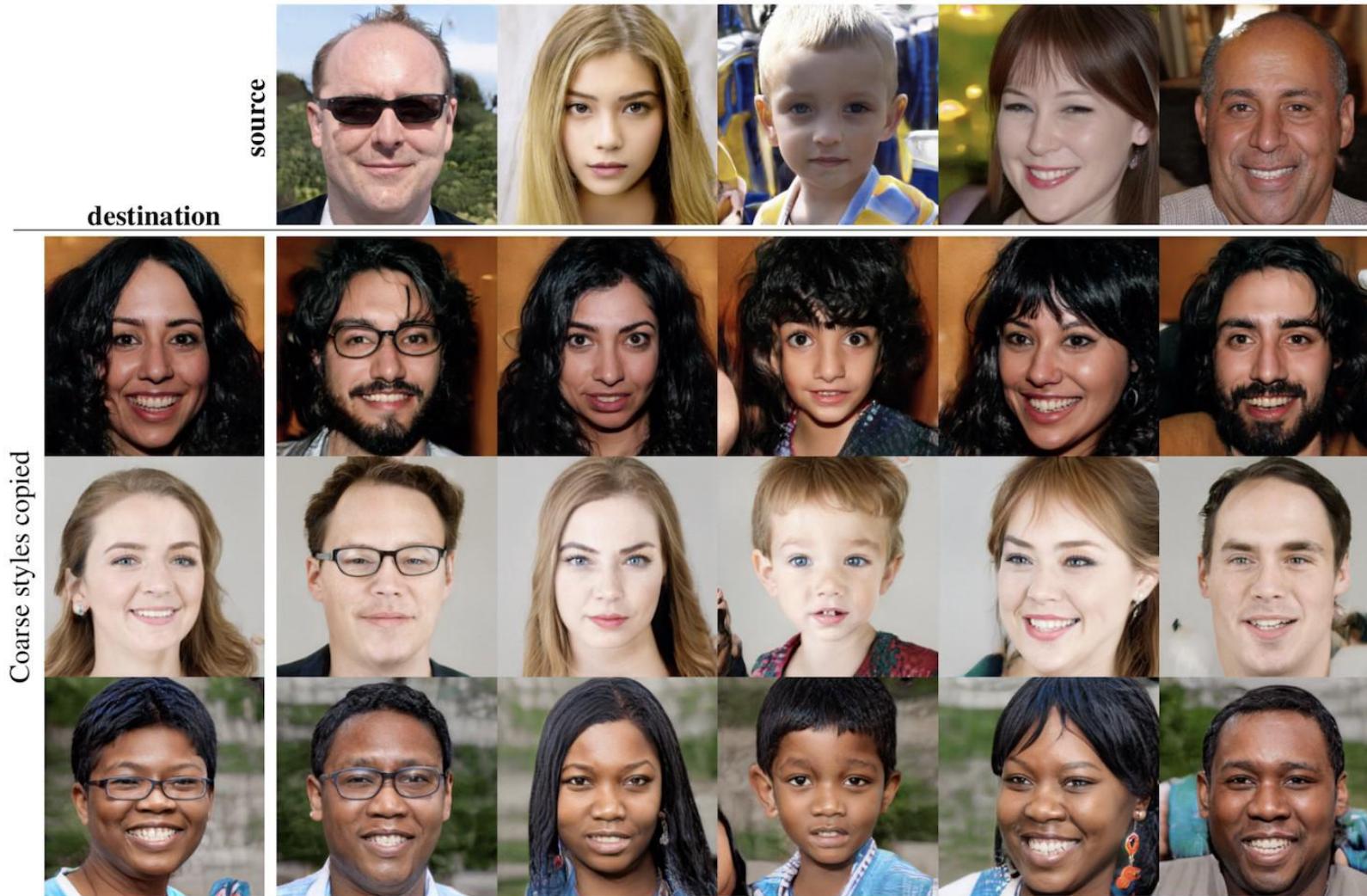


# Progressive Growing of GANs

---

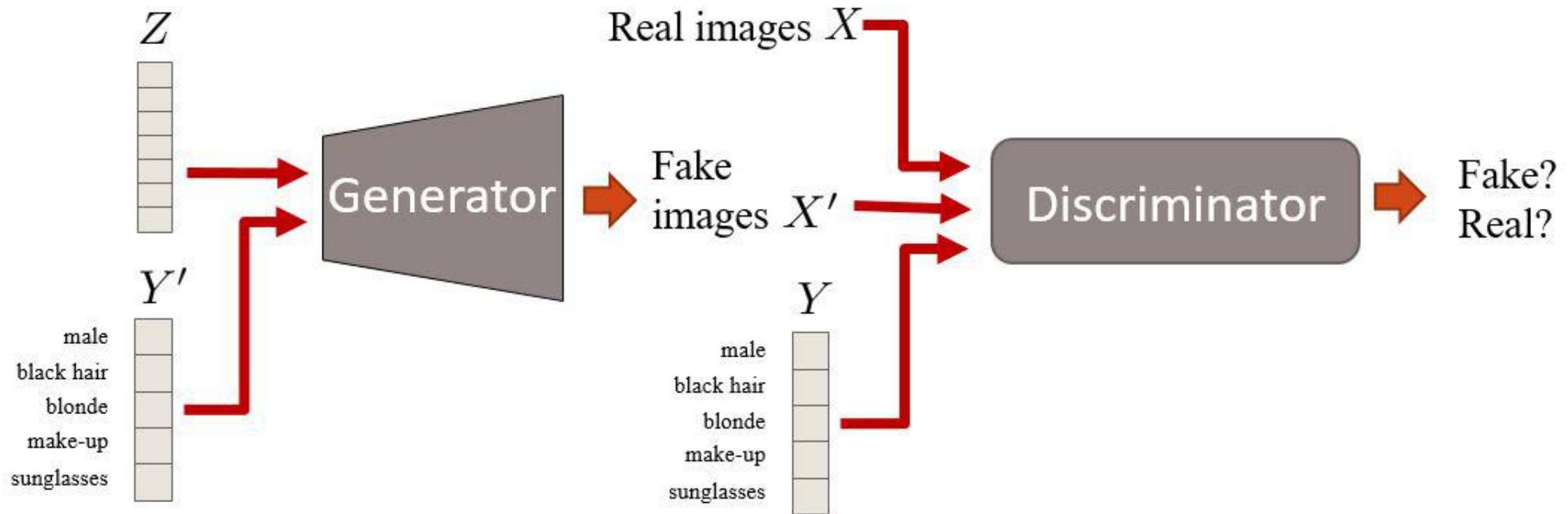


# StyleGAN



# Conditional GAN

---



# Conditional GAN: pix2pix

---



# Conditional GAN: pix2pix

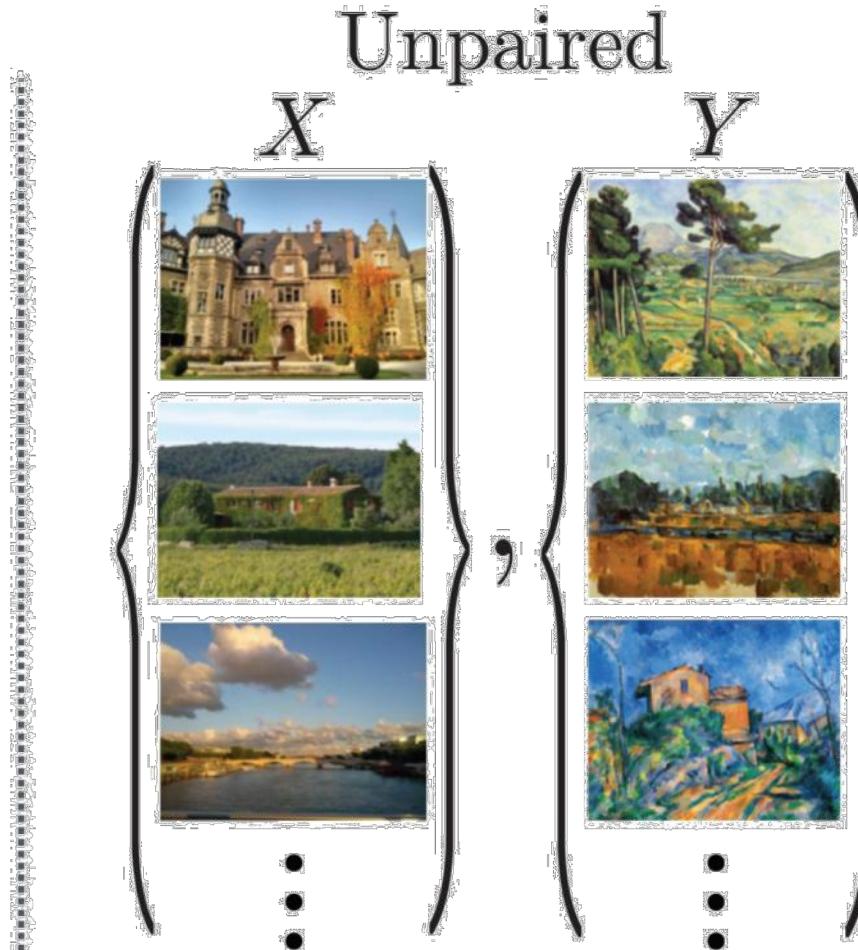
---



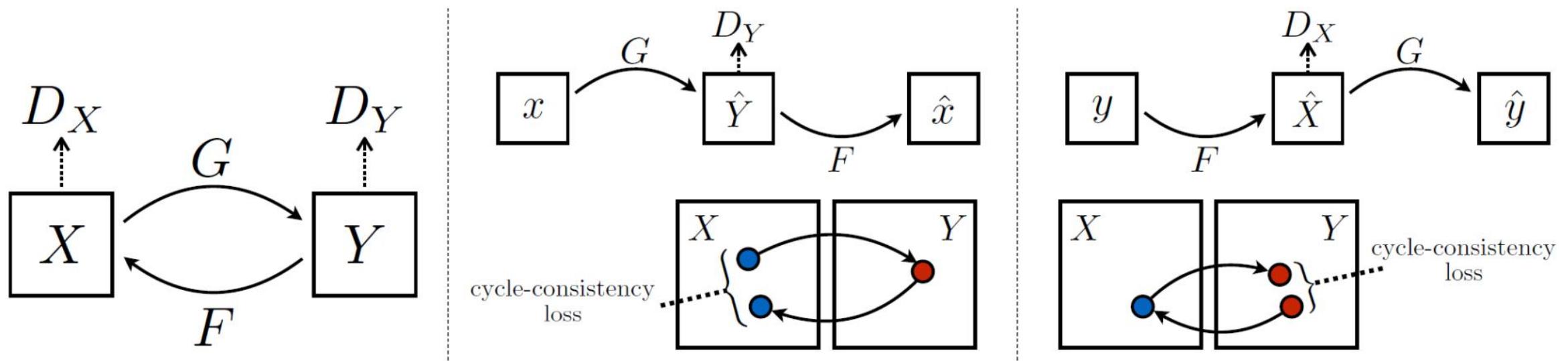
# Conditional GAN: pix2pix



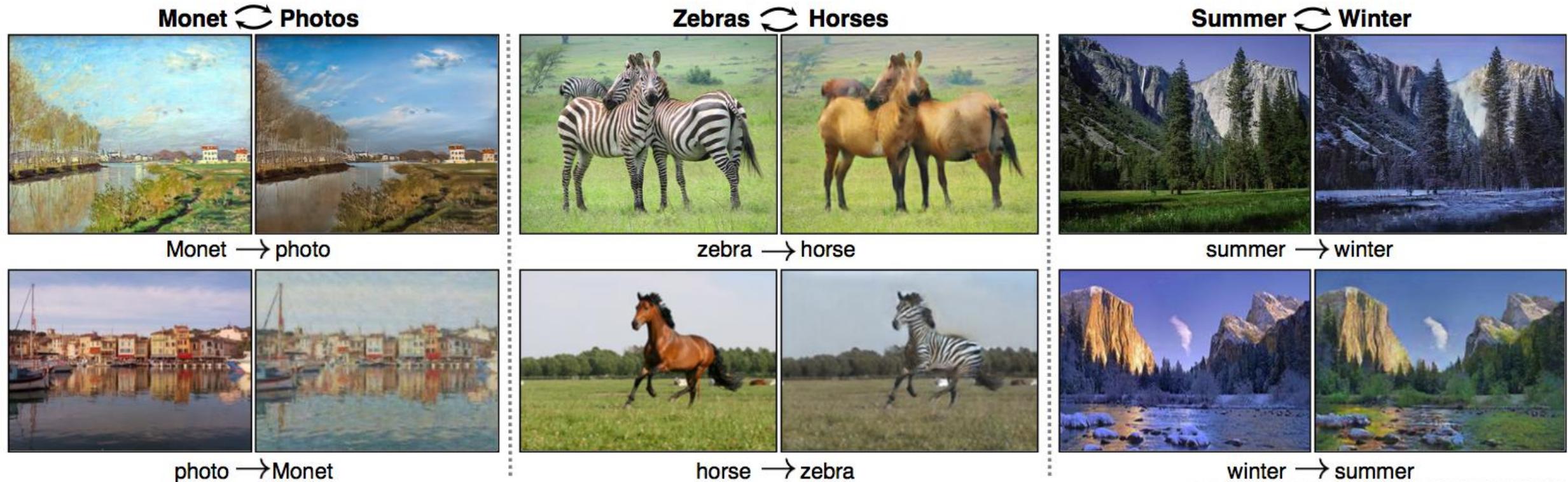
# Paired vs Unpaired Data



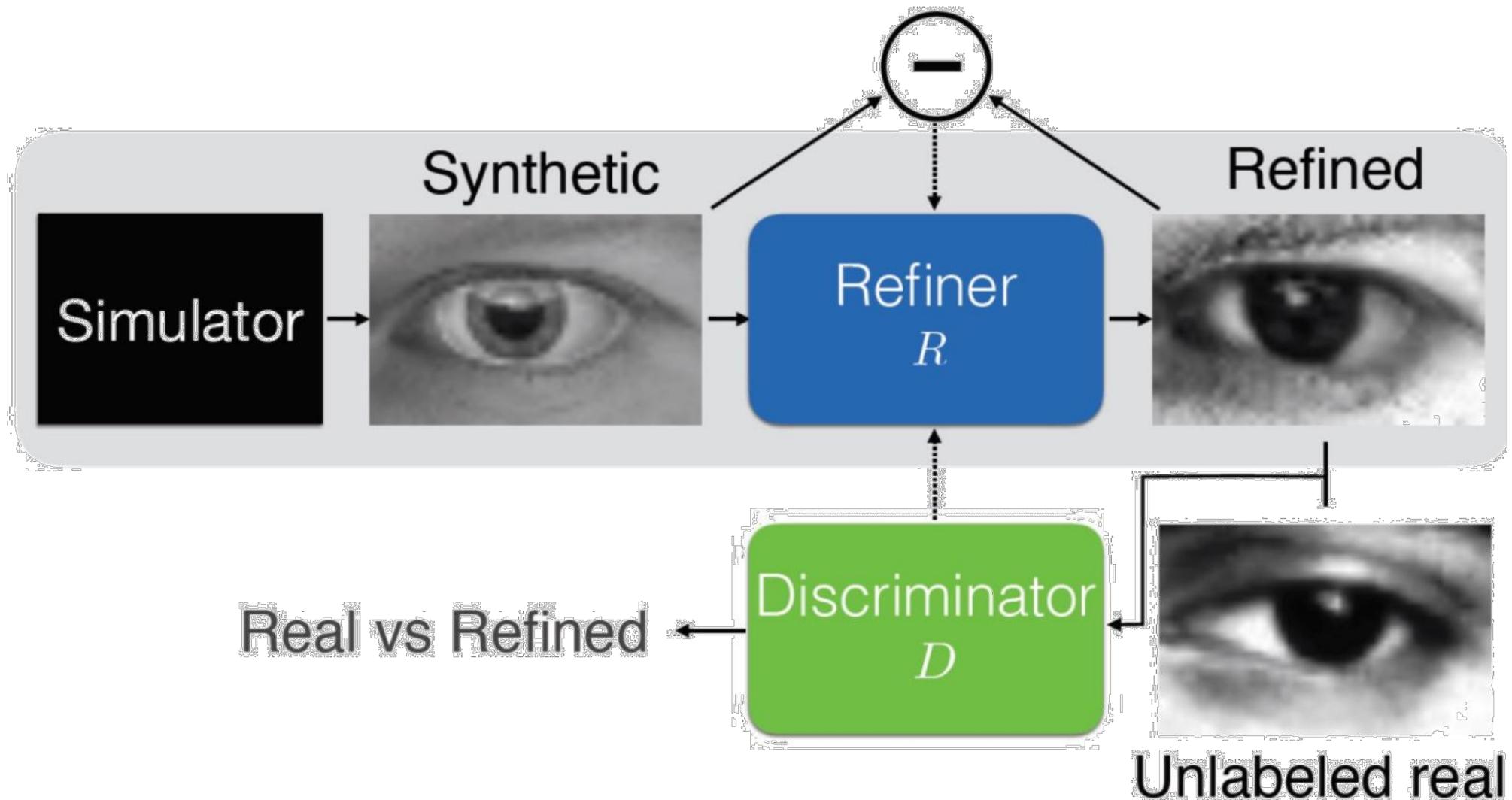
# CycleGAN



# CycleGAN



# GANs for Synthetic Data Generation



# GAN Zoo

---

- GAN - Generative Adversarial Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- AdaGAN - AdaGAN: Boosting Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference
- AM-GAN - Generative Adversarial Nets with Labeled Data by Activation Maximization
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorical GANs
- b-GAN - b-GAN: Unified Framework of Generative Adversarial Networks
- Bayesian GAN - Deep and Hierarchical Implicit Models
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BiGAN - Adversarial Feature Learning
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- CGAN - Conditional Generative Adversarial Nets
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks
- CCGAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CoGAN - Coupled Generative Adversarial Networks
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- DTN - Unsupervised Cross-Domain Image Generation
- DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
- DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DR-GAN - Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
- DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- EBGAN - Energy-based Generative Adversarial Network
- f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN - Towards Large-Pose Face Frontalization in the Wild
- GAWWN - Learning What and Where to Draw
- GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data
- Geometric GAN - Geometric GAN
- GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending
- IAN - Neural Photo Editing with Introspective Adversarial Networks
- iGAN - Generative Visual Manipulation on the Natural Image Manifold
- IcGAN - Invertible Conditional GANs for image editing
- ID-CGAN - Image De-raining Using a Conditional Generative Adversarial Network
- Improved GAN - Improved Techniques for Training GANs
- InfoGAN - InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
- LAGAN - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
- LAPGAN - Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

# Final Remarks

---

- GAN и VAE – до сих пор очень горячие темы в глубоком обучении.
- VAE – очень красавая с теоретической точки зрения идея, которую можно использовать не только в качестве генеративной модели.
- GAN – не менее красавая идея, но менее теоретически обоснованная. GAN тоже можно использовать не только для целей генерации примеров.

# Useful Materials

---

Advances in Generative Adversarial Networks (GANs) (Medium):

<https://medium.com/beyondminds/advances-in-generative-adversarial-networks-7bad57028032>

ProGAN: How NVIDIA Generated Images of Unprecedented Quality (Medium):

<https://towardsdatascience.com/progan-how-nvidia-generated-images-of-unprecedented-quality-51c98ec2cbd2>

Explained: A Style-Based Generator Architecture for GANs - Generating and Tuning Realistic Artificial Faces (Medium):

<https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431>