

Генеративные модели. Часть 2

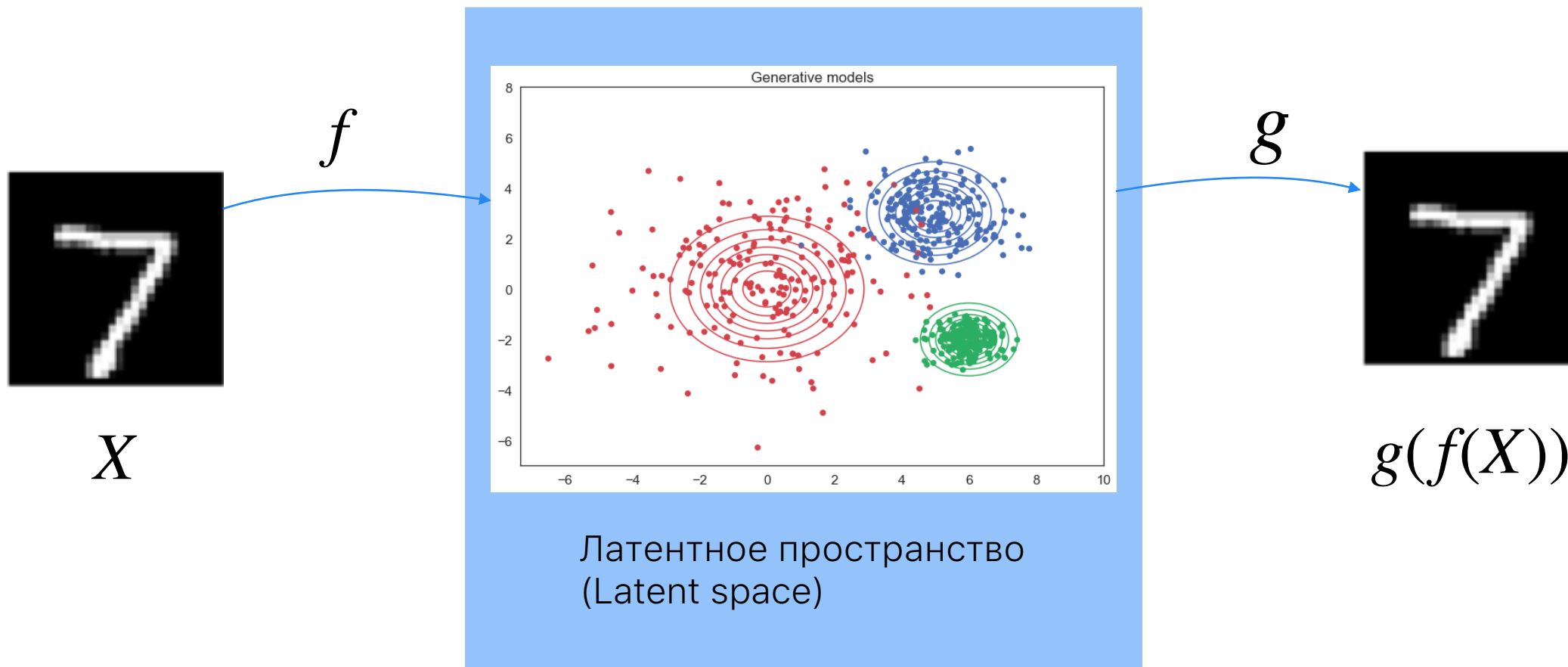
Никита Балаганский

План

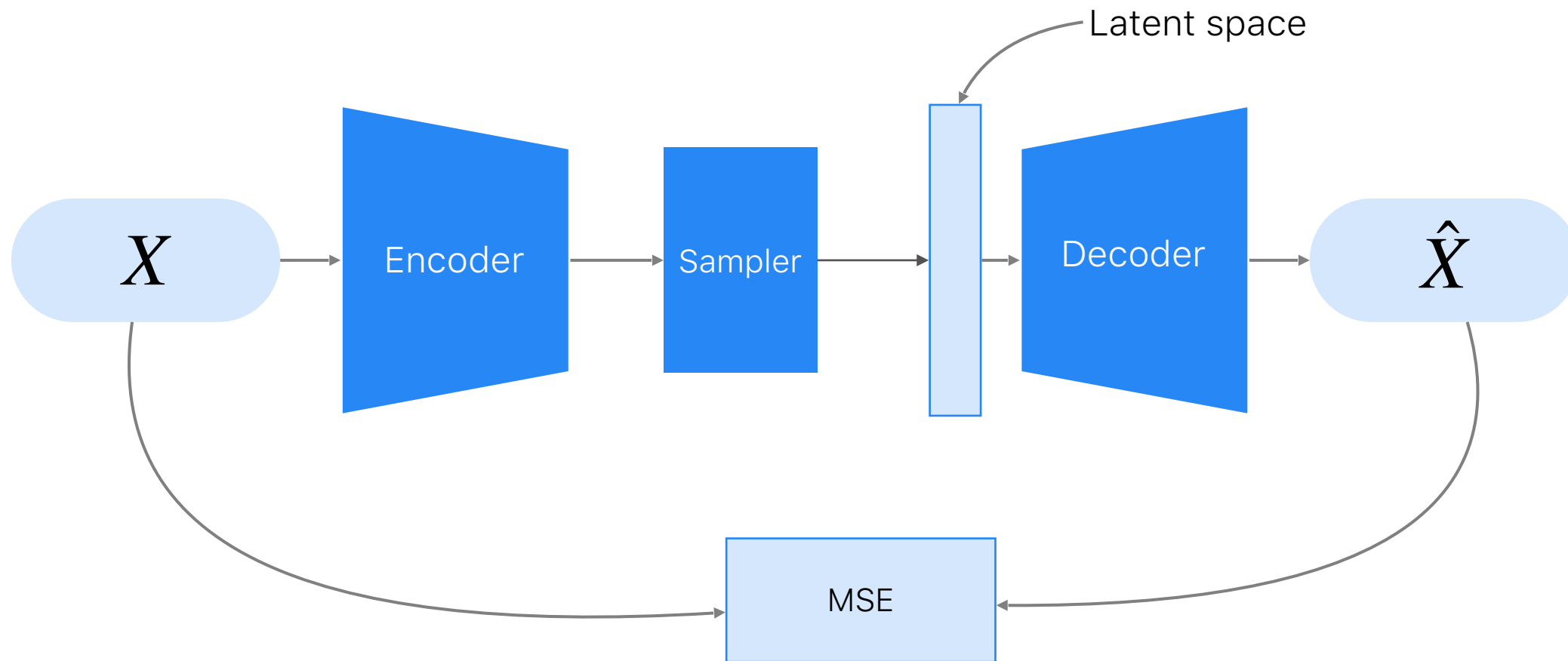
- Идея генеративных моделей
- Что не так с VAE?
- Идея GAN
- Трюки при обучении
- Архитектуры GAN

На чем мы остановились?

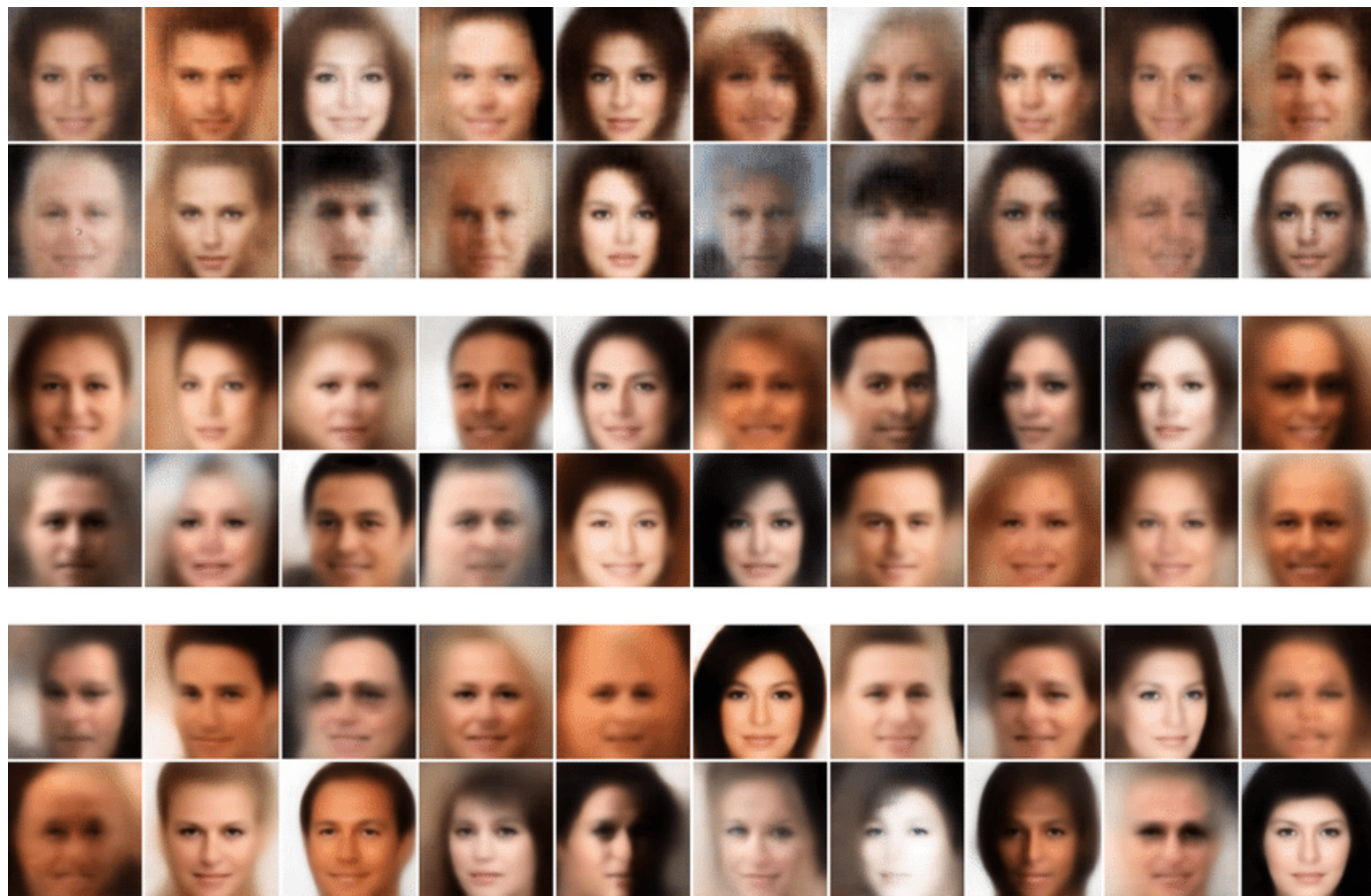
Generative models



VAE

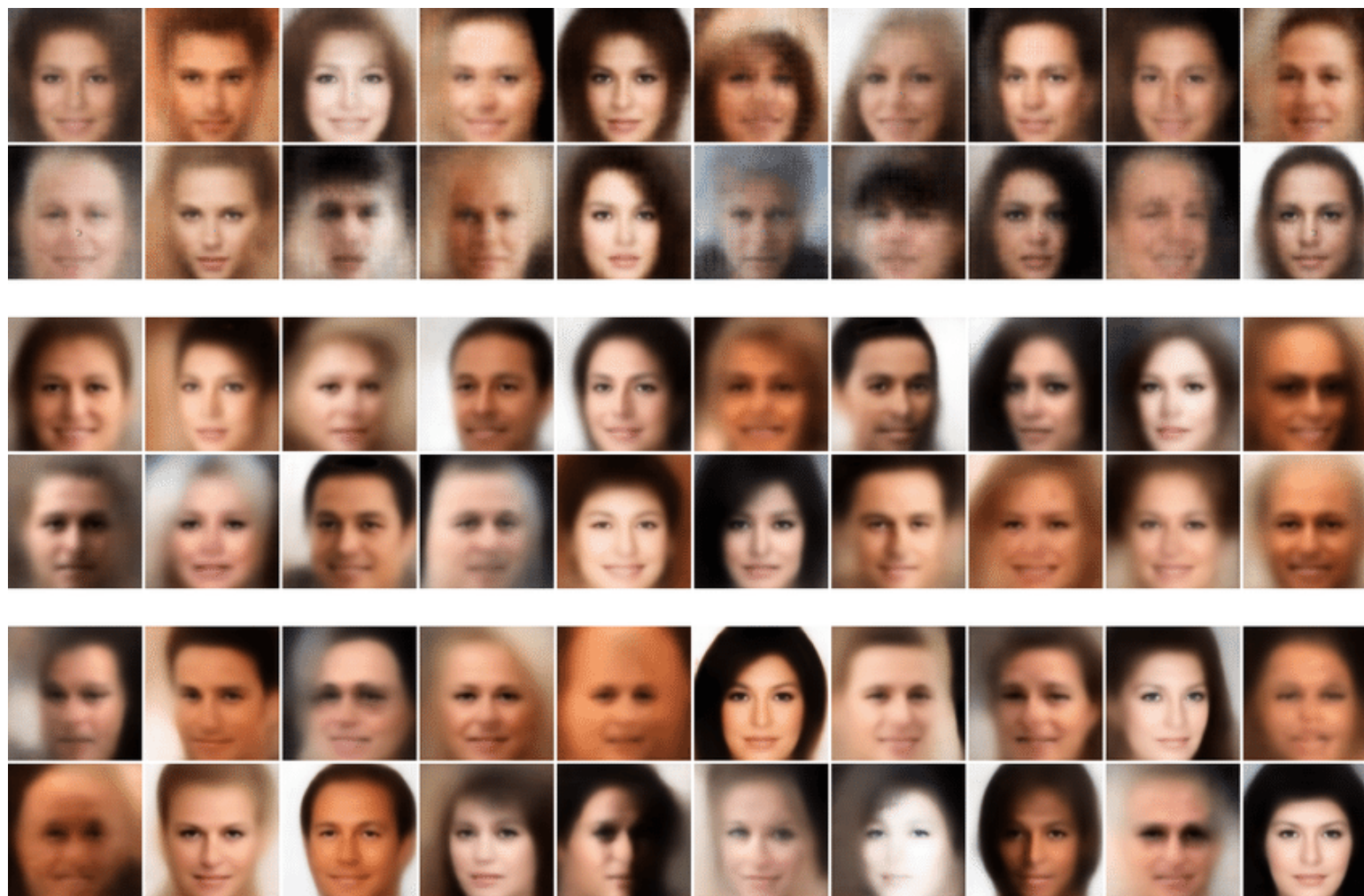


Генерация с VAE

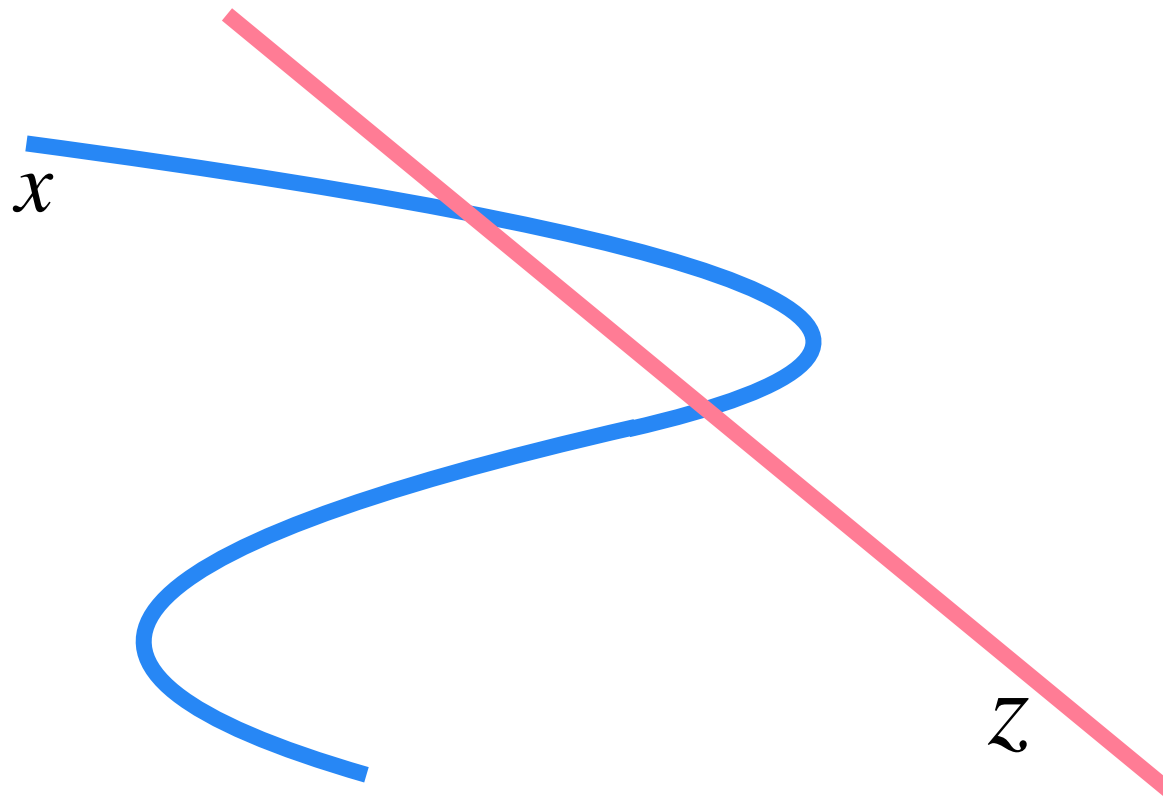


Генерация с VAE

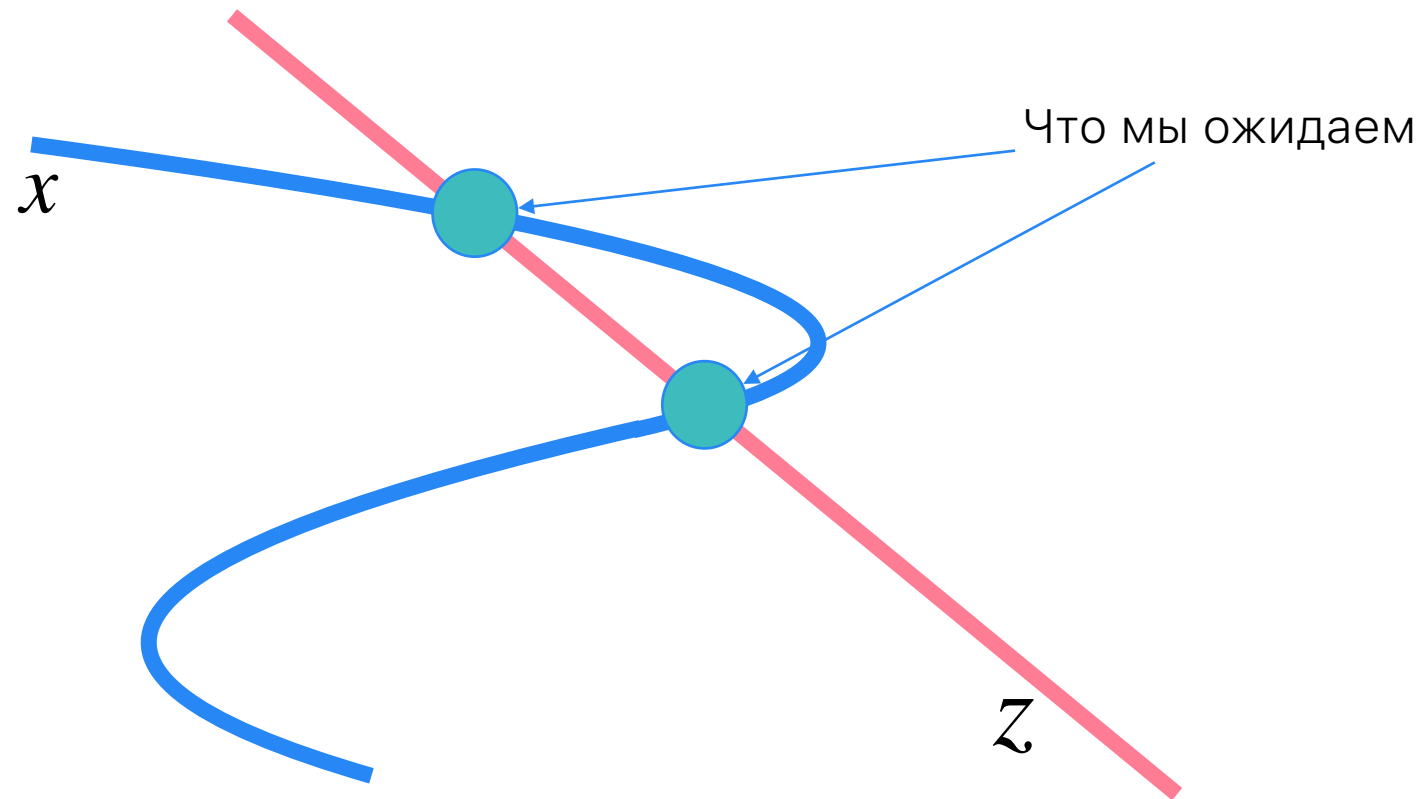
Изображения мутные



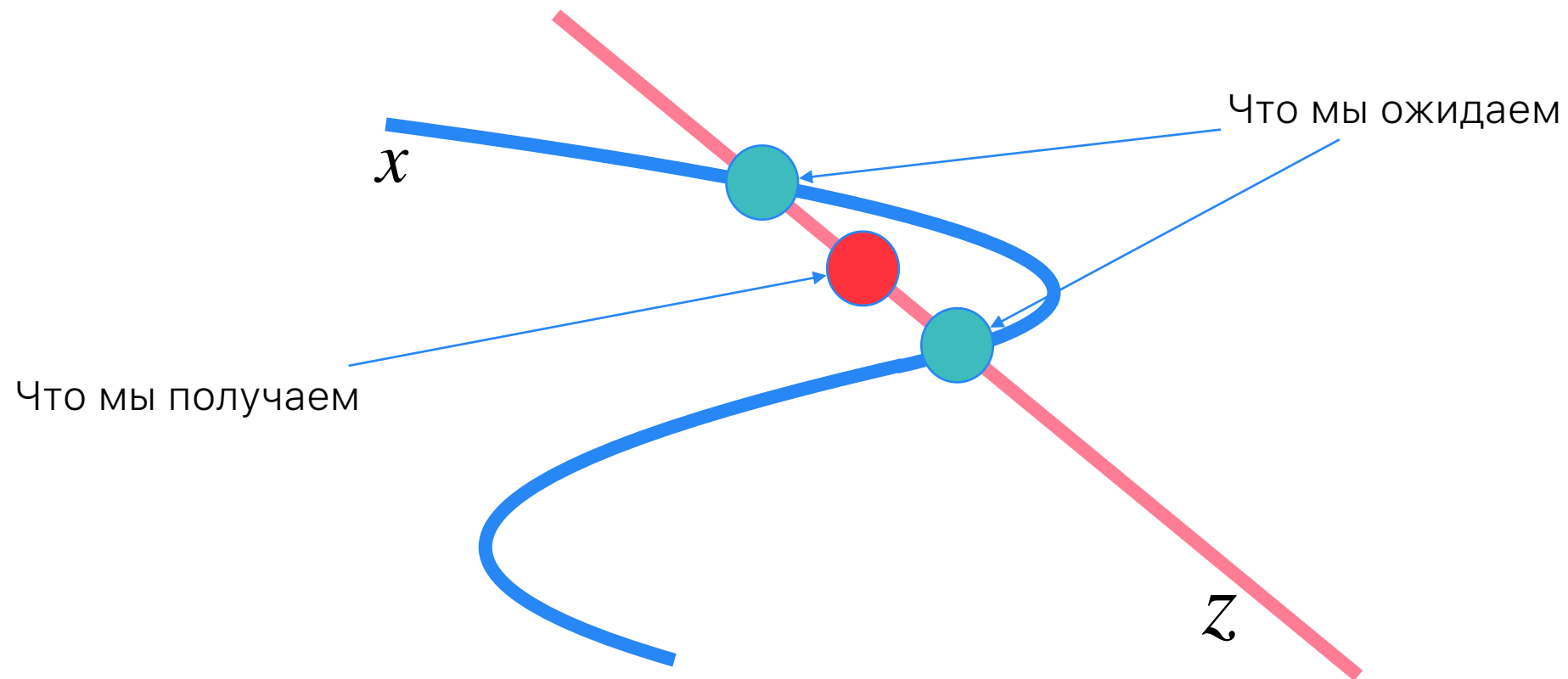
Проблемы MSE



Проблемы MSE



Проблемы MSE



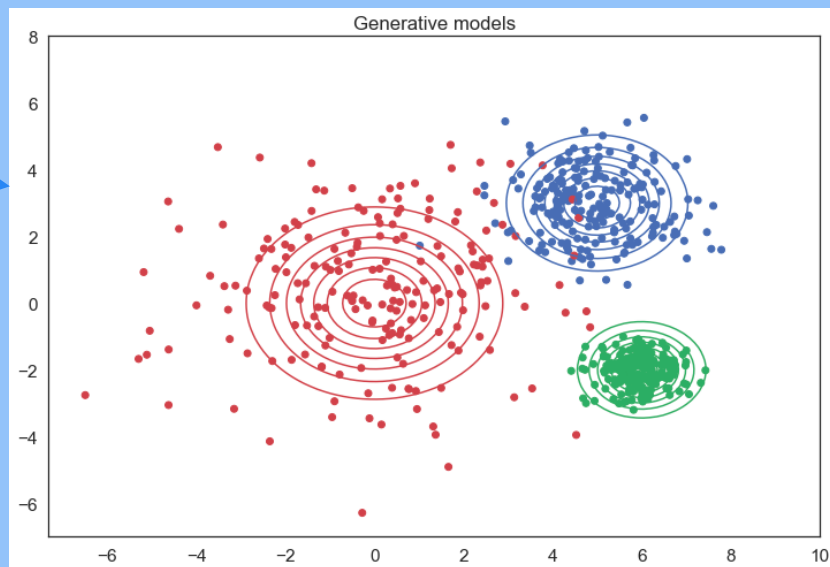
Идея GAN

Generative models



X

f



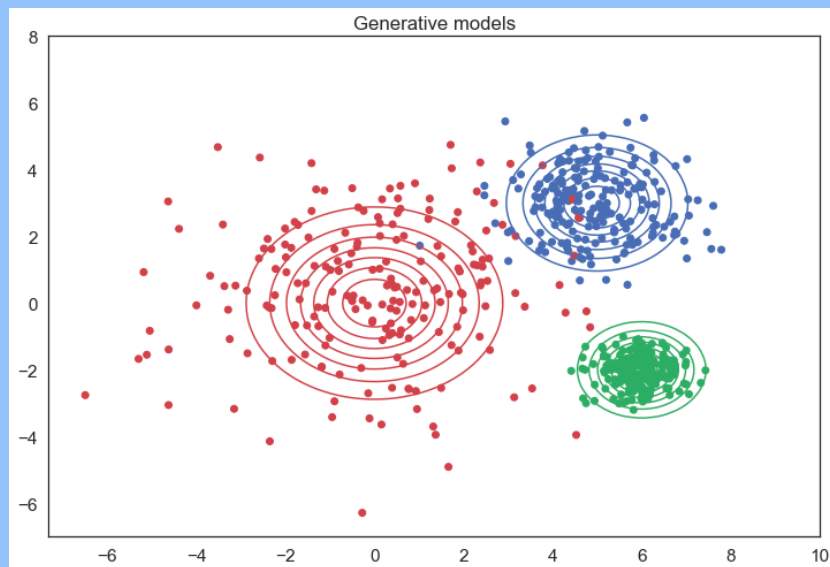
Латентное пространство
(Latent space)

g



$g(f(X))$

Generative models



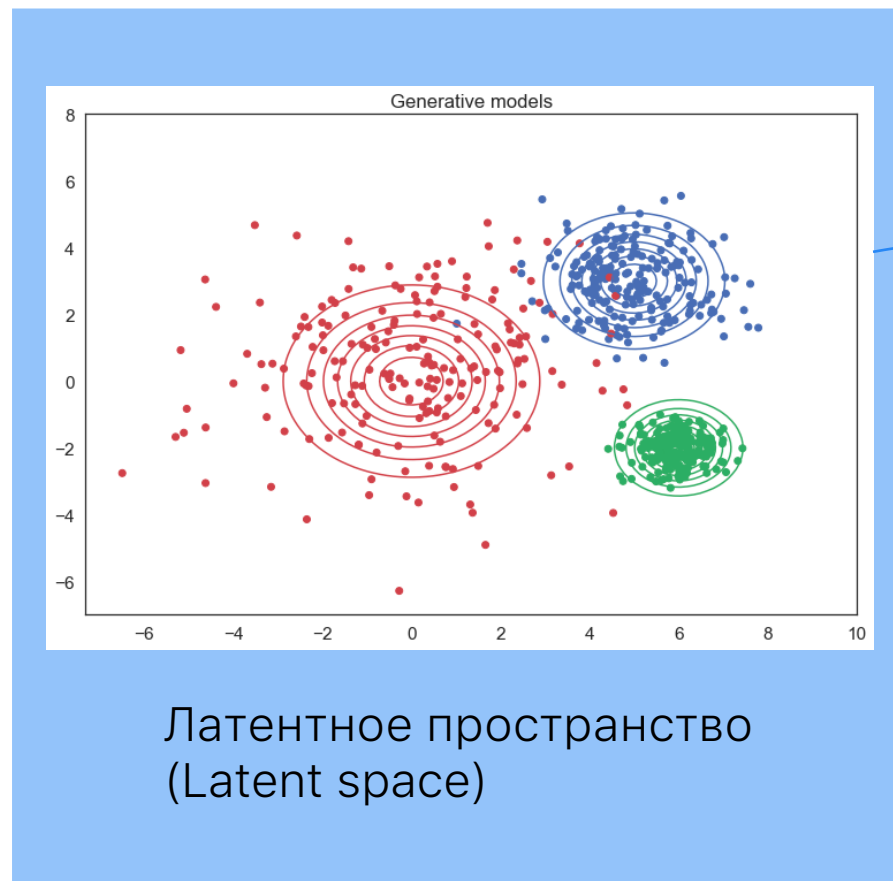
Латентное пространство
(Latent space)

g



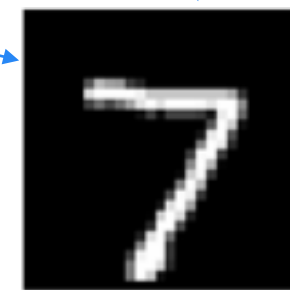
$g(z)$

Generative models



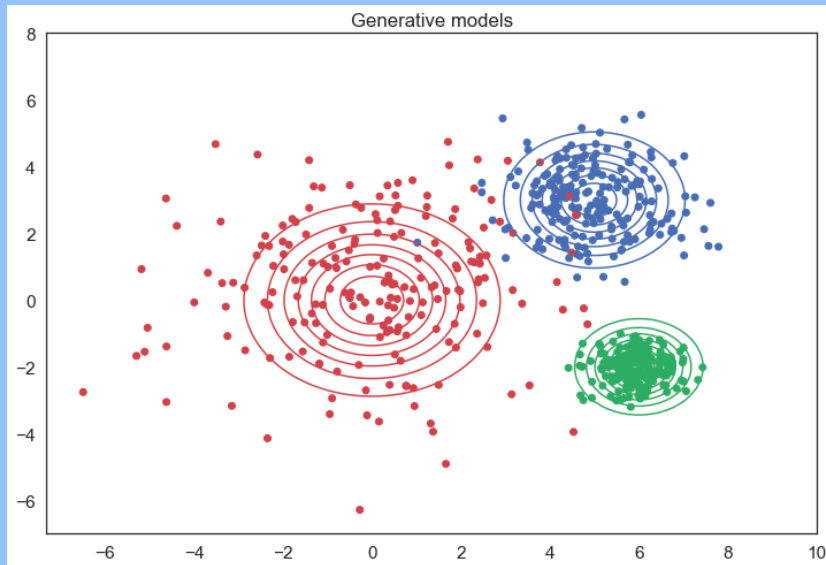
Хорошая картинка?

g



$g(z)$

Generative models



Латентное пространство
(Latent space)

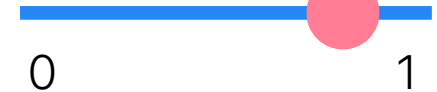
Хорошая картинка?

g



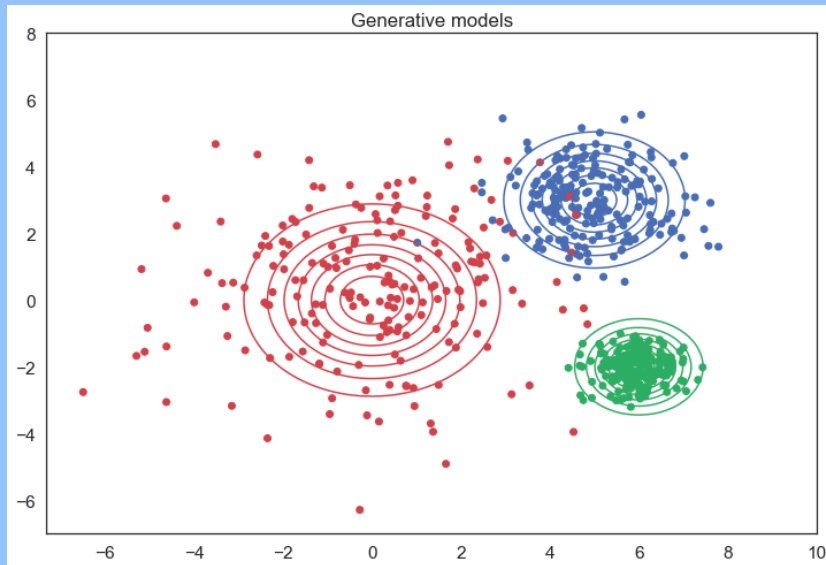
$g(z)$

d



$P(\text{"картинка хорошая"})$

Generative models



Латентное пространство
(Latent space)

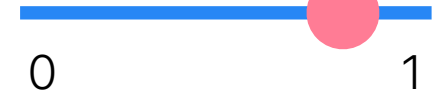
Хорошая картинка?

g



$g(z)$

d



$P(\text{"картинка хорошая"})$

Чего мы хотим?

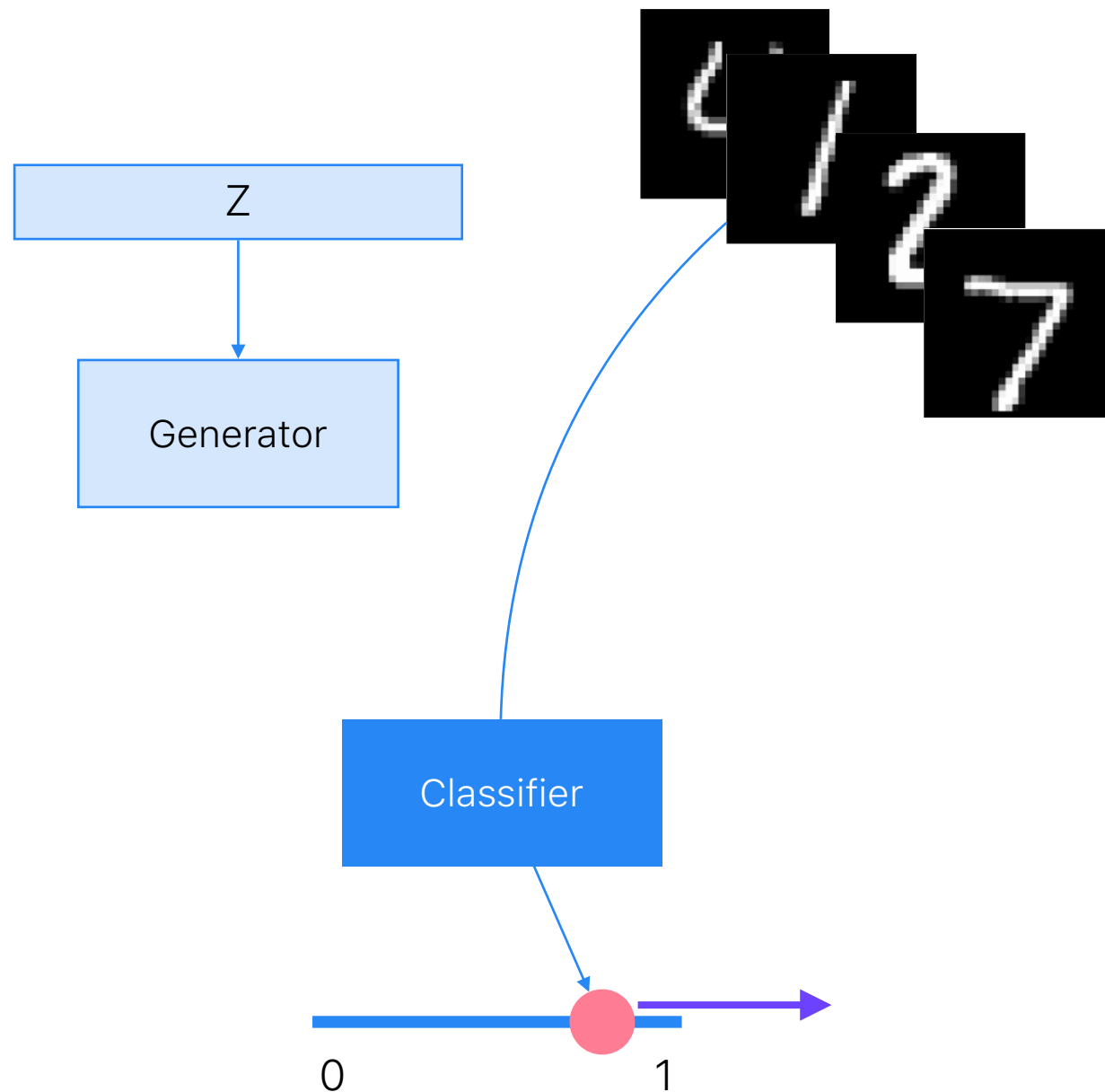
Мы хотим генерировать картинки как и в датасете. Значит, сгенерированная картинка должна быть похожей на датасет.

Чего мы хотим?

Мы хотим генерировать картинки как и в датасете. Значит, сгенерированная картинка должна быть похожей на датасет.

Итак:

1) Для изображений из датасета мы хотим близкую к единице вероятность.

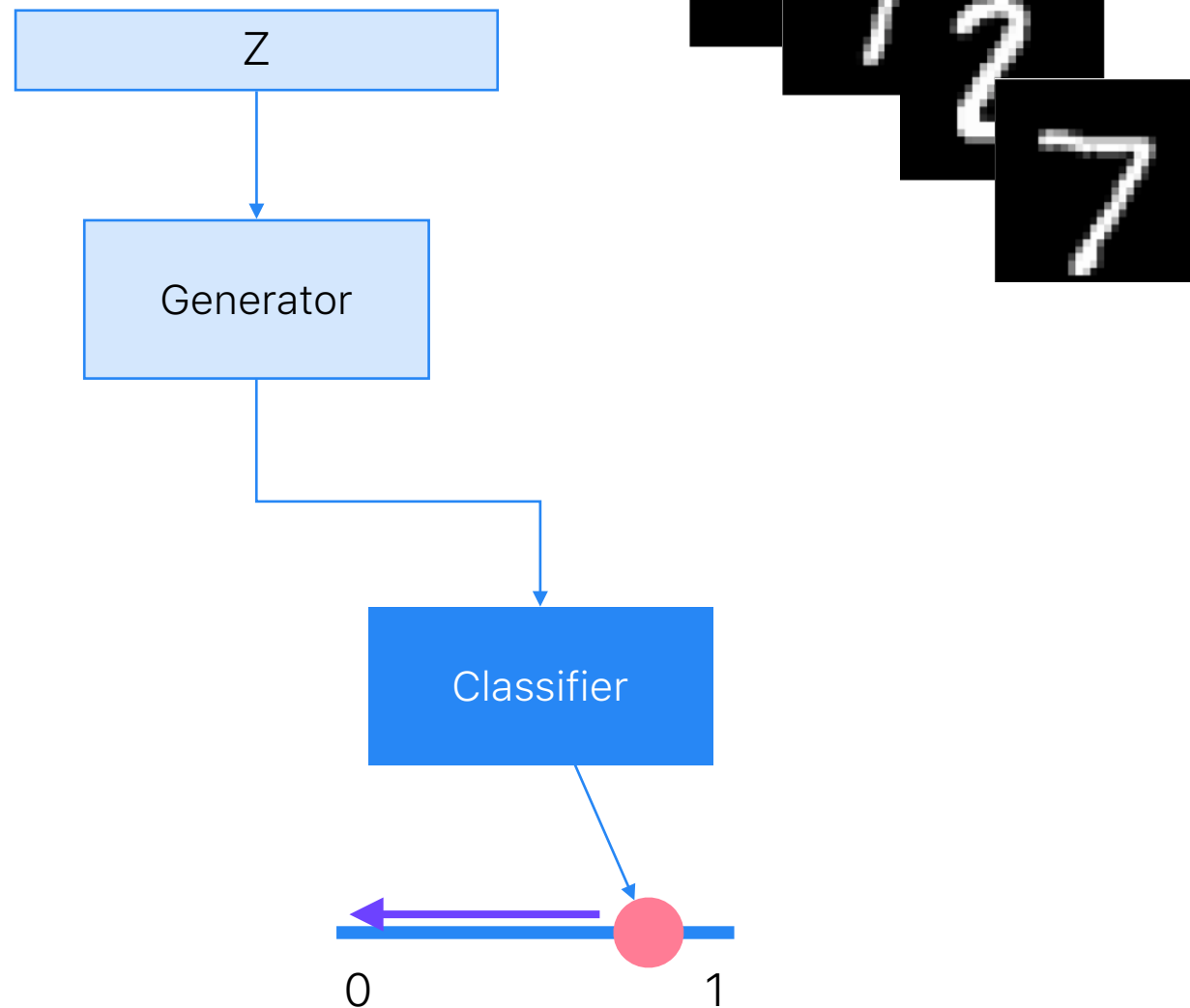


Чего мы хотим?

Мы хотим генерировать картинки как и в датасете. Значит, сгенерированная картинка должна быть похожей на датасет.

Итак:

- 1) Для изображений из датасета мы хотим близкую к единице вероятность.
- 2) Для всех остальных картинок мы хотим получить низкую вероятность.

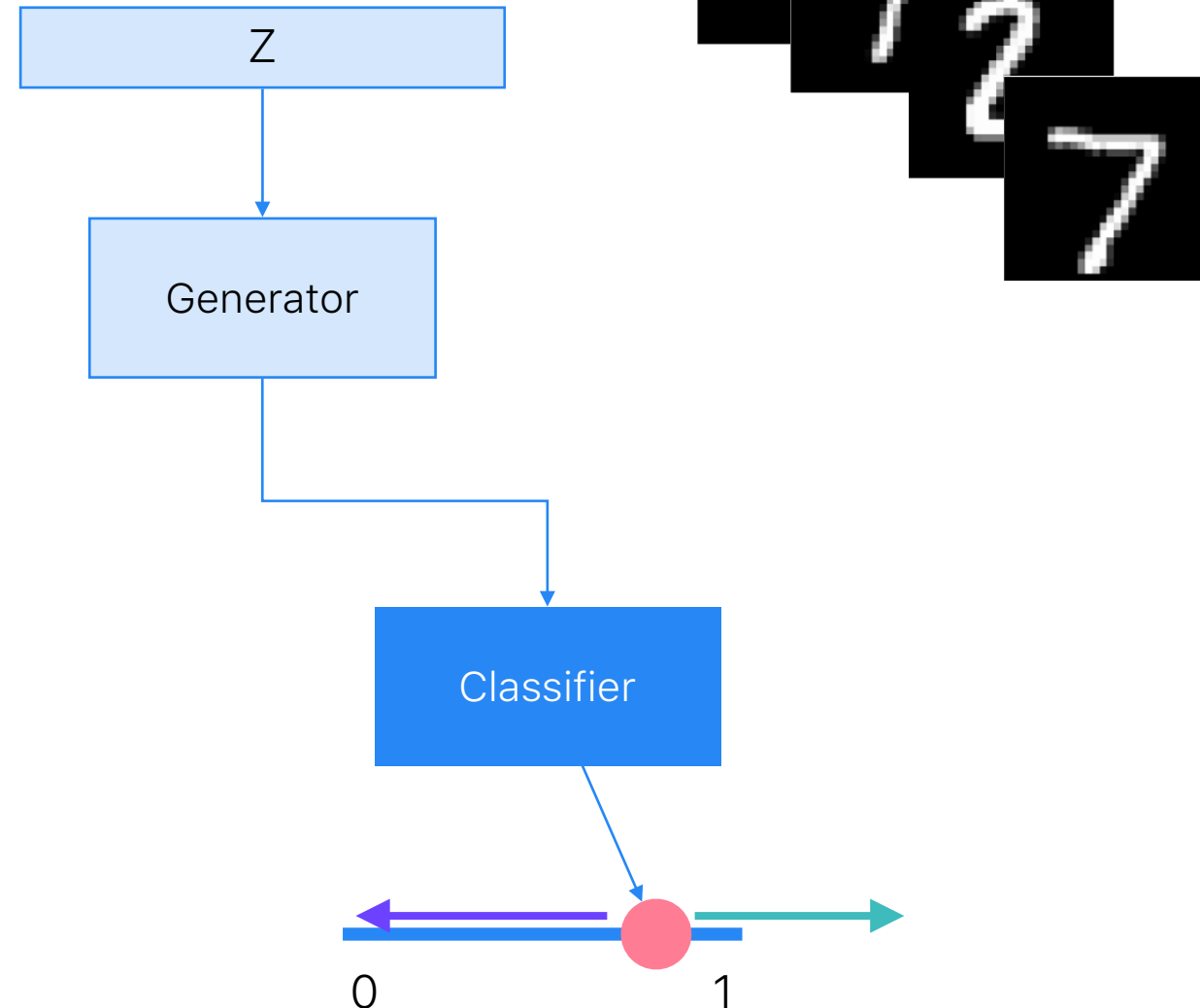


Чего мы хотим?

Мы хотим генерировать картинки как и в датасете. Значит, сгенерированная картинка должна быть похожей на датасет.

Итак:

- 1) Для изображений из датасета мы хотим близкую к единице вероятность.
- 2) Для всех остальных картинок мы хотим получить низкую вероятность.
- 3) С другой стороны при обучении генератора мы хотим добиться высокой вероятности.

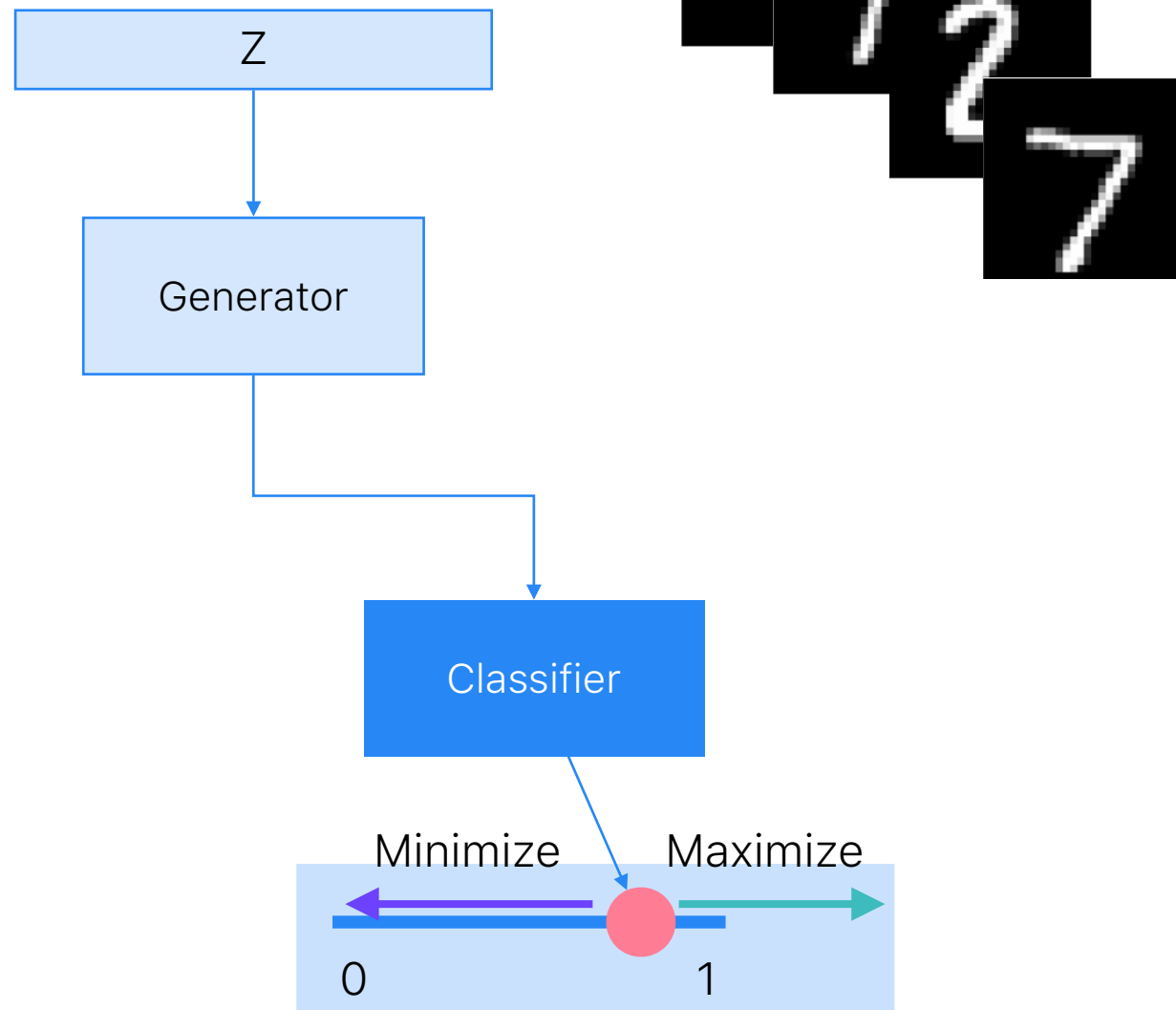


Чего мы хотим?

Мы хотим генерировать картинки как и в датасете. Значит, сгенерированная картинка должна быть похожей на датасет.

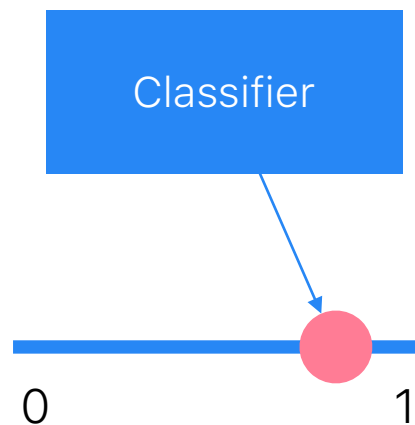
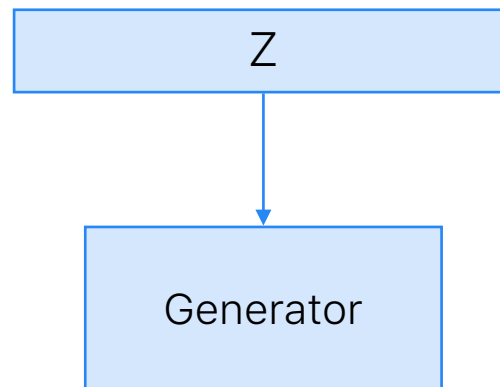
Итак:

- 1) Для изображений из датасета мы хотим близкую к единице вероятность.
- 2) Для всех остальных картинок мы хотим получить низкую вероятность.
- 3) С другой стороны при обучении генератора мы хотим добиться высокой вероятности.



Loss-функция

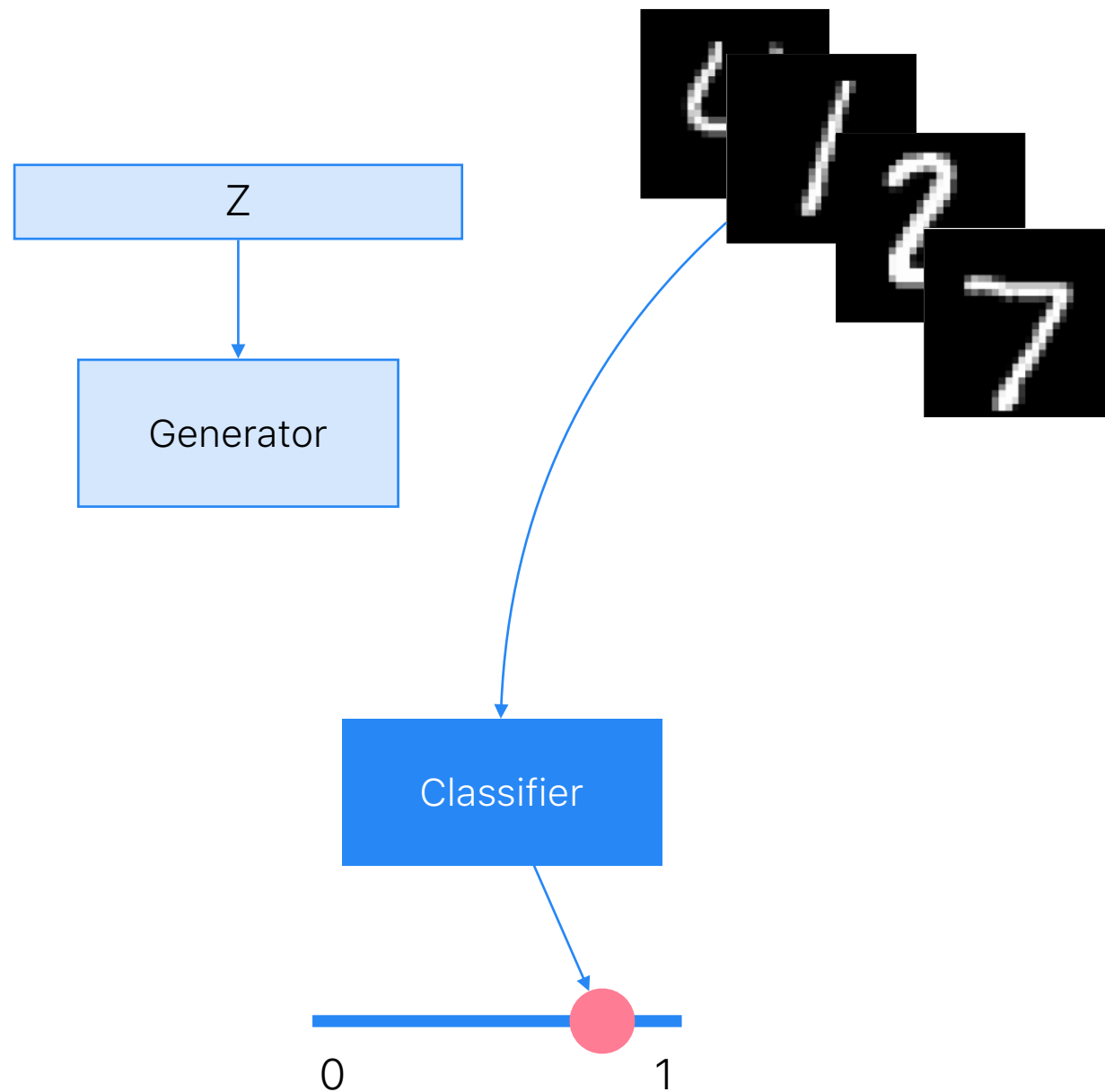
$$L = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$



Loss-функция

$$L = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

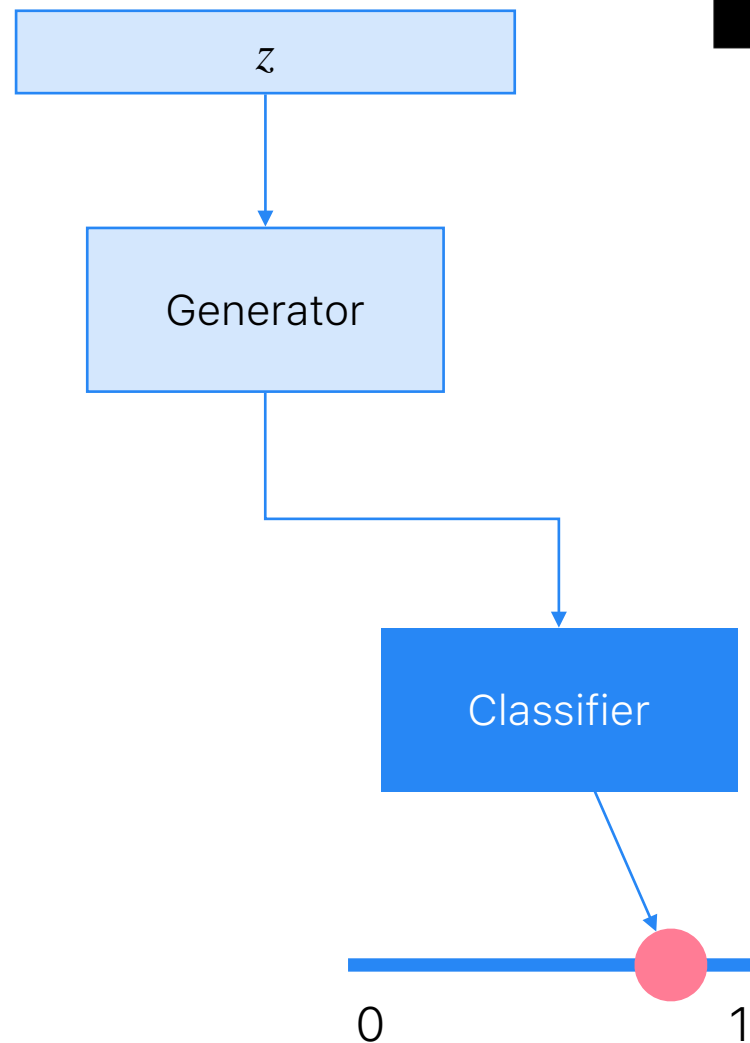
$P(\text{"картинка хорошая"})$



Loss-функция

$1 - P(\text{"картинка хорошая"})$

$$L = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

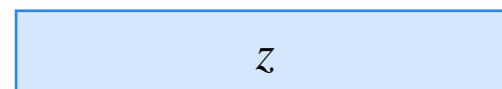


Loss-функция

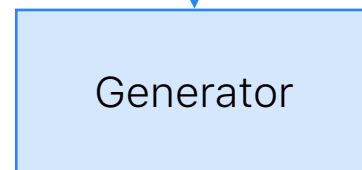
$1 - P(\text{"картинка хорошая"})$

$$L = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

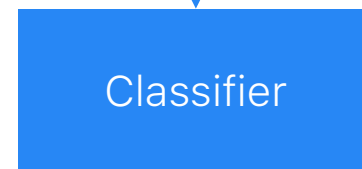
Generator



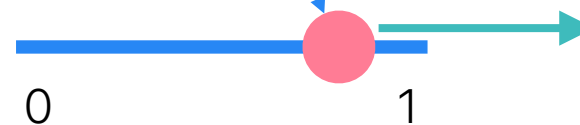
z



Generator



Classifier



Loss-функция

$1 - P(\text{"картинка хорошая"})$

$$L = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

Generator

Discriminator

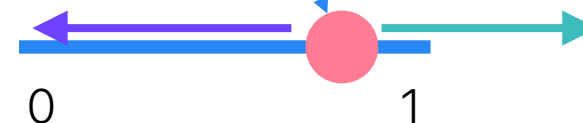
z

Generator

Classifier

0

1



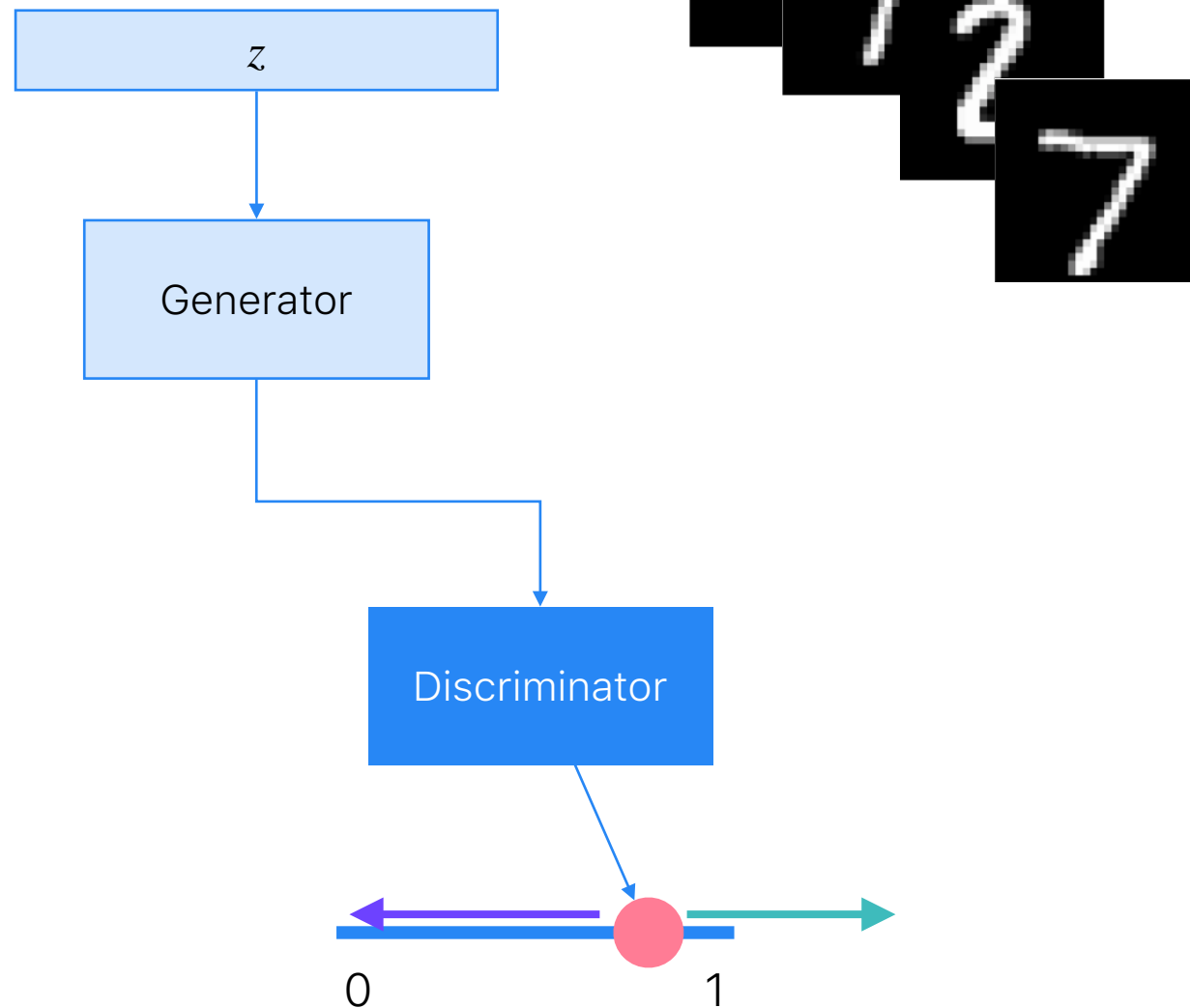
Loss-функция

$1 - P(\text{"картинка хорошая"})$

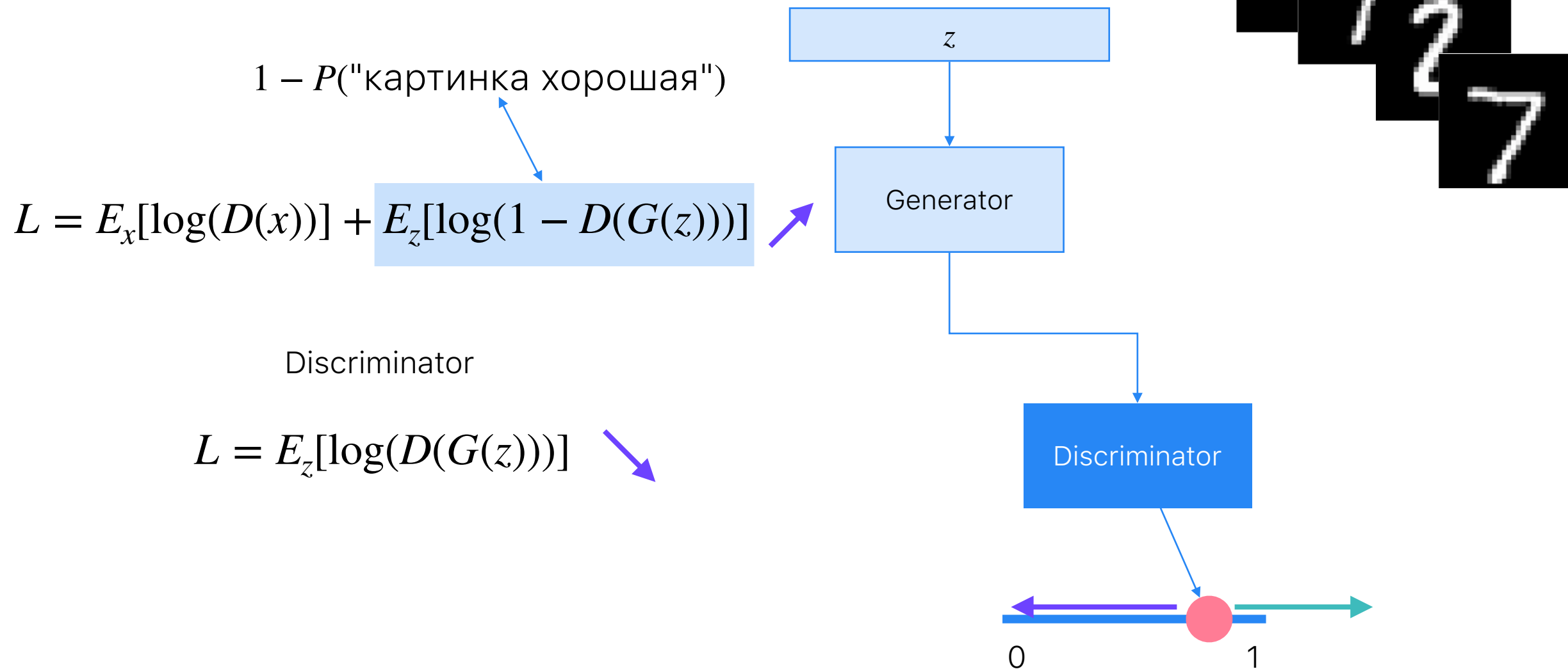
$$L = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

Generator

Discriminator

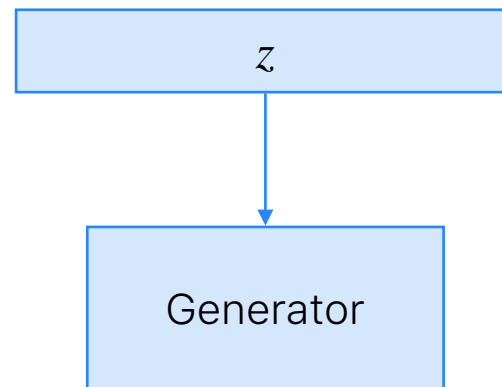


Loss-функция



Практика

$$L = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

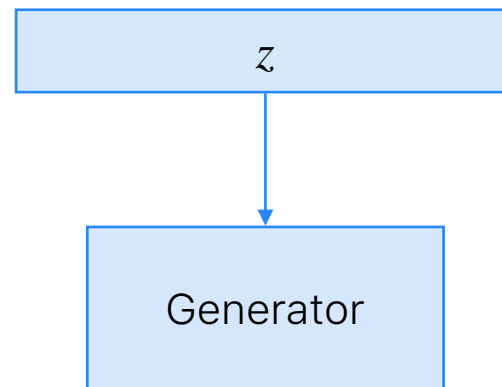


Discriminator



Практика

$$L = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

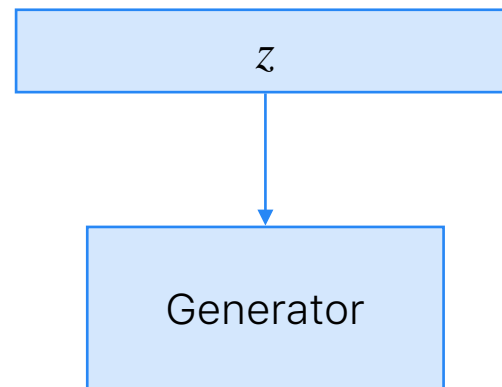


Discriminator



Практика

$$L = \frac{1}{B} \sum_{i \in B} [\log(D(x_i)) + [\log(1 - D(G(z_i)))]]$$

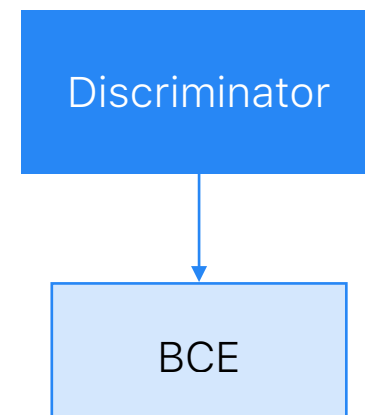
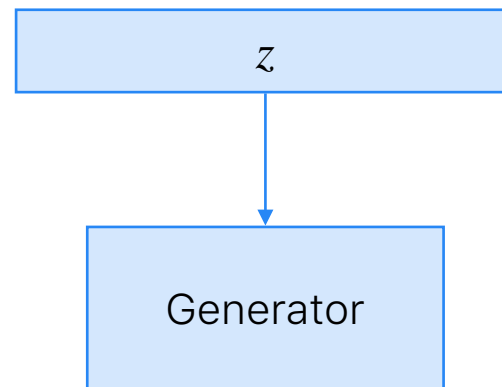


Discriminator



Практика

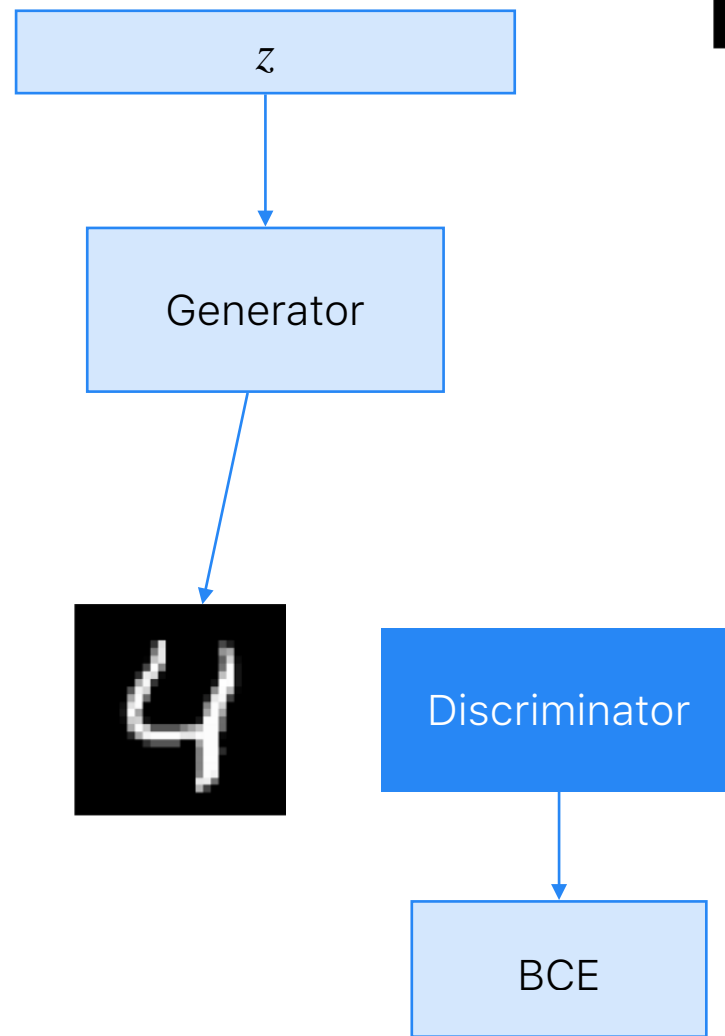
Рассмотрим все этапы одной итерации:



Практика

Рассмотрим все этапы одной итерации:

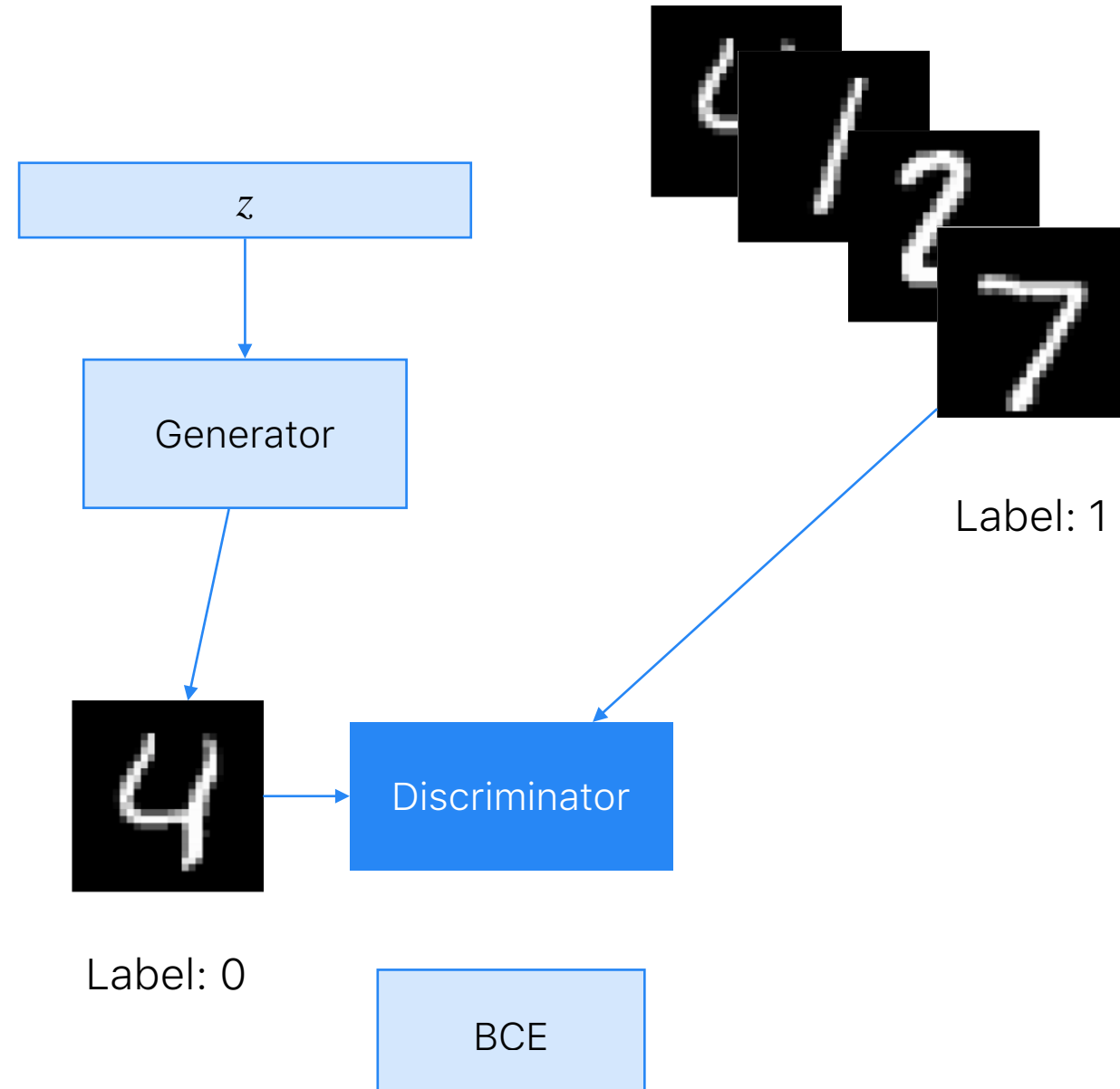
1) Генерируем изображение



Практика

Рассмотрим все этапы одной итерации:

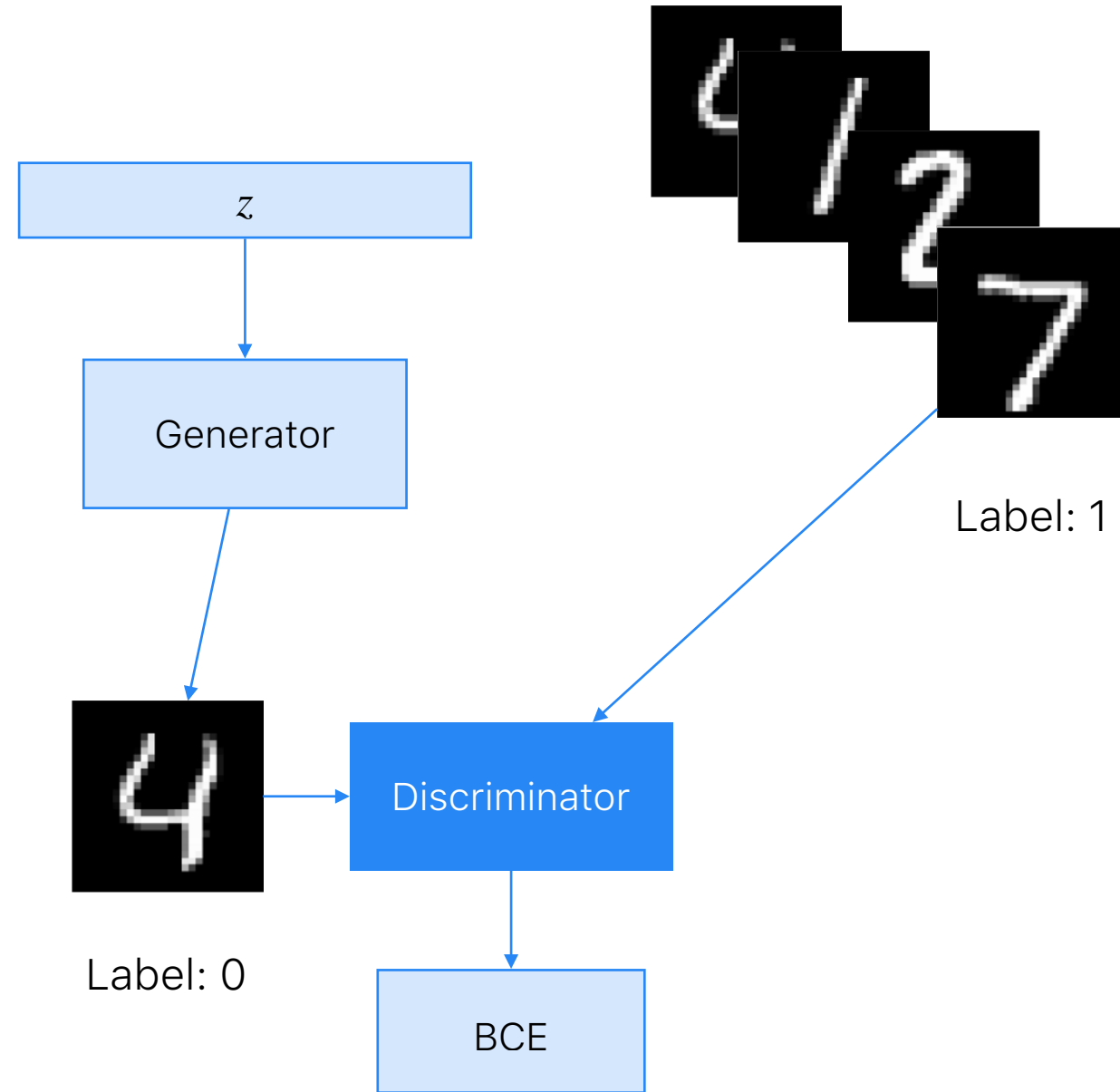
- 1) Генерируем изображение
- 2) Конкатенируем сгенерированные изображения и изображение из батча. Расставляем метки как на картинке



Практика

Рассмотрим все этапы одной итерации:

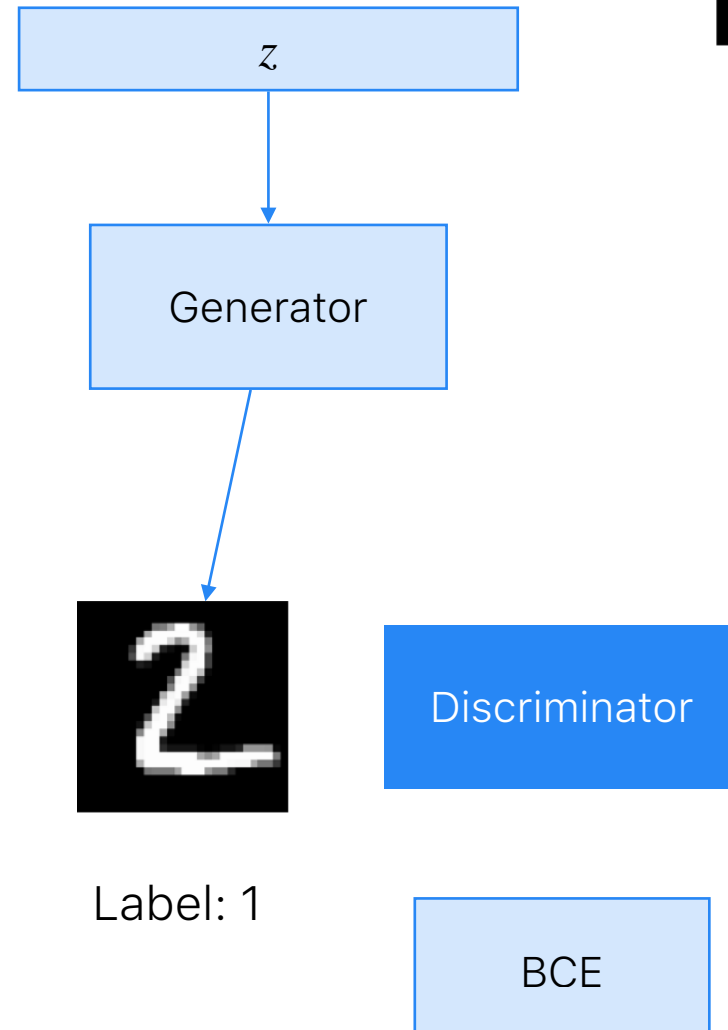
- 1) Генерируем изображение
- 2) Конкатенируем сгенерированные изображения и изображение из батча. Расставляем метки как на картинке
- 3) Считаем BCE и обновляем веса дискриминатора (генератор не трогаем)



Практика

Рассмотрим все этапы одной итерации:

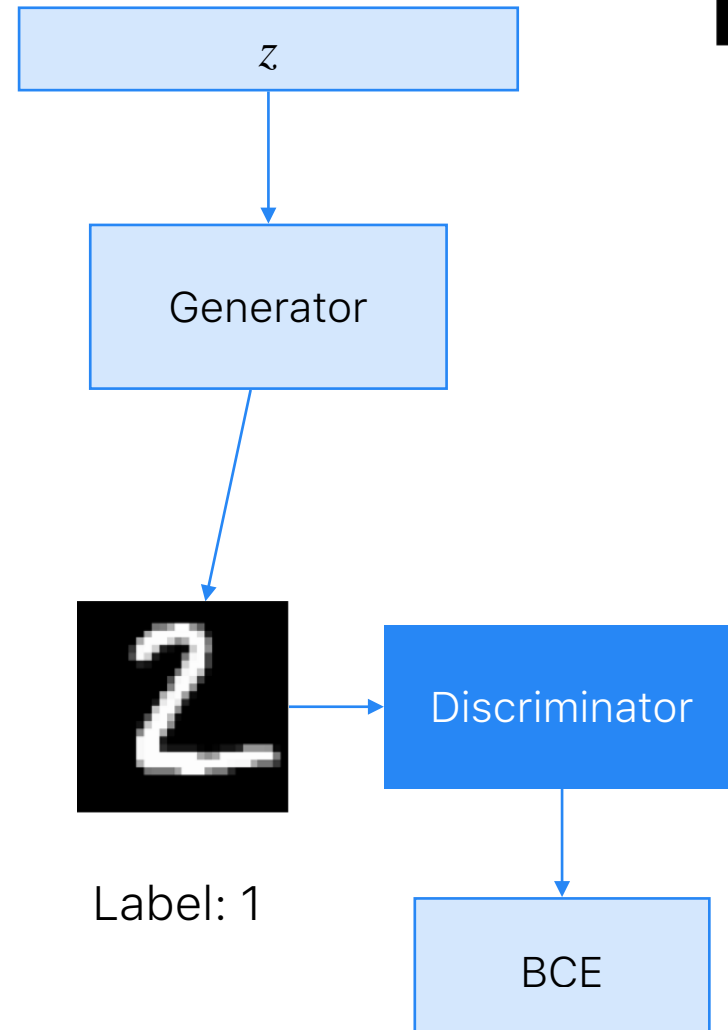
- 1) Генерируем изображение
- 2) Конкатенируем сгенерированные изображения и изображение из батча. Расставляем метки как на картинке
- 3) Считаем BCE и обновляем веса дискриминатора (генератор не трогаем)
- 4) Генерируем новое изображение и инвертируем его метку.



Практика

Рассмотрим все этапы одной итерации:

- 1) Генерируем изображение
- 2) Конкатенируем сгенерированные изображения и изображение из батча. Расставляем метки как на картинке
- 3) Считаем BCE и обновляем веса дискриминатора (генератор не трогаем)
- 4) Генерируем новое изображение и инвертируем его метку.
- 5) Считаем BCE и обновляем веса генератора (дискриминатор не трогаем)

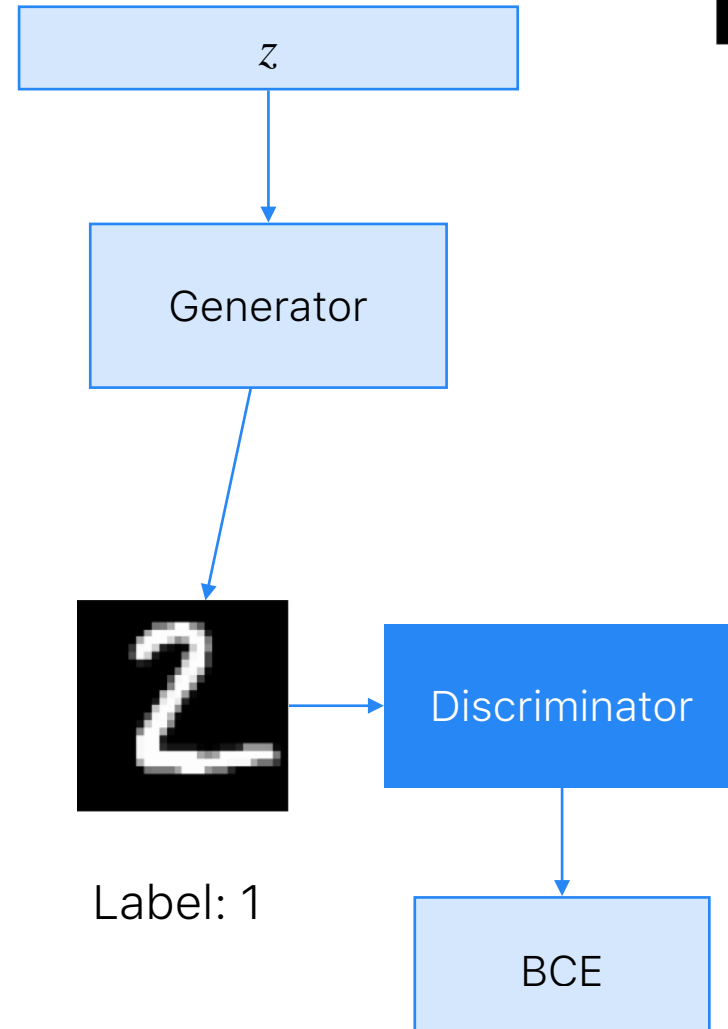


Практика

Рассмотрим все этапы одной итерации:

- 1) Генерируем изображение
- 2) Конкатенируем сгенерированные изображения и изображение из батча. Расставляем метки как на картинке
- 3) Считаем BCE и обновляем веса дискриминатора (генератор не трогаем)
- 4) Генерируем новое изображение и инвертируем его метку.
- 5) Считаем BCE и обновляем веса генератора (дискриминатор не трогаем)

Обучаем дискриминатор



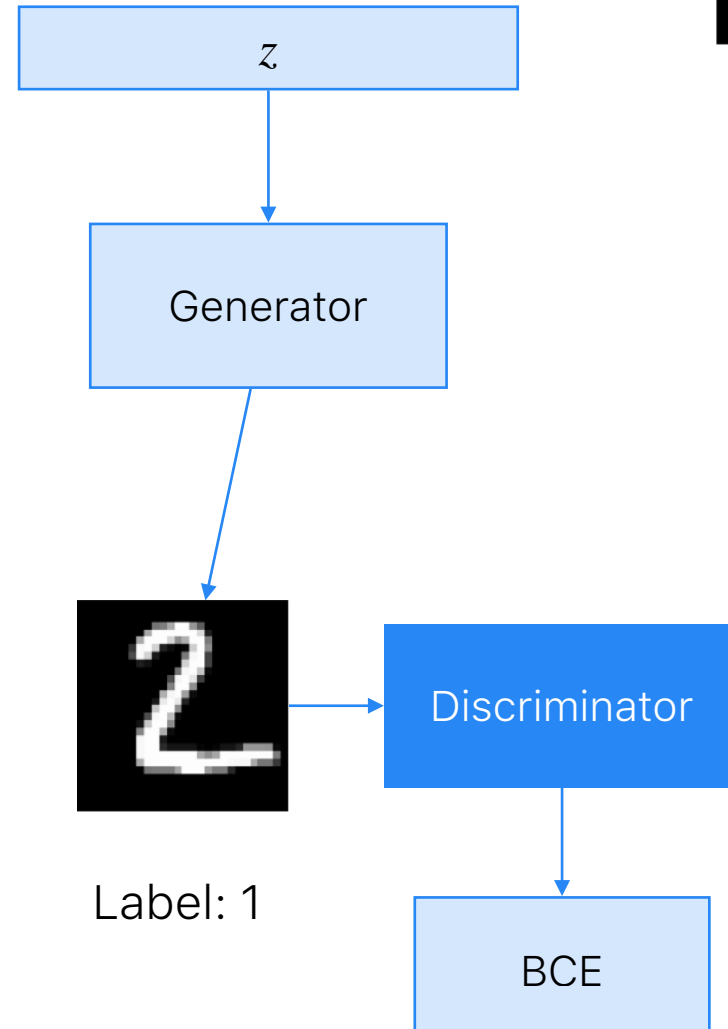
Практика

Рассмотрим все этапы одной итерации:

- 1) Генерируем изображение
- 2) Конкатенируем сгенерированные изображения и изображение из батча. Расставляем метки как на картинке
- 3) Считаем BCE и обновляем веса дискриминатора (генератор не трогаем)
- 4) Генерируем новое изображение и инвертируем его метку.
- 5) Считаем BCE и обновляем веса генератора (дискриминатор не трогаем)

Обучаем дискриминатор

Обучаем генератор



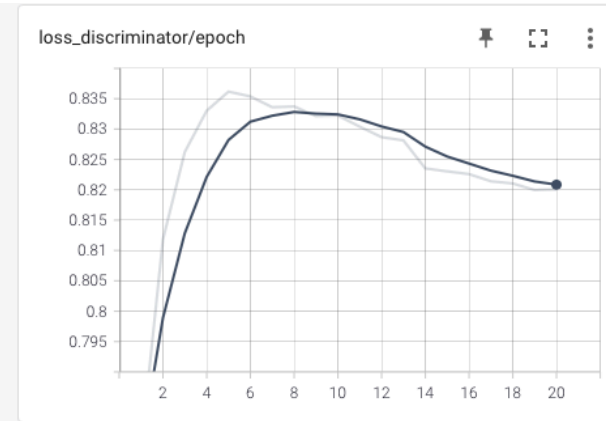
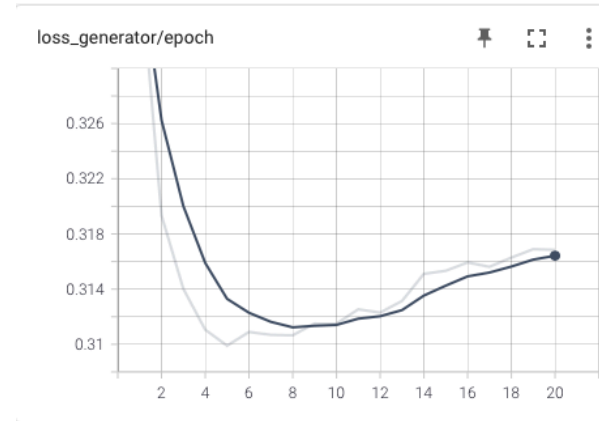
Практика

Обучаем дискриминатор

Рассмотрим все этапы одной итерации:

- 1) Генерируем изображение
- 2) Конкатенируем сгенерированные изображения и изображение из батча. Расставляем метки как на картинке
- 3) Считаем BCE и обновляем веса дискриминатора (генератор не трогаем)
- 4) Генерируем новое изображение и инвертируем его метку.
- 5) Считаем BCE и обновляем веса генератора (дискриминатор не трогаем)

Обучаем генератор



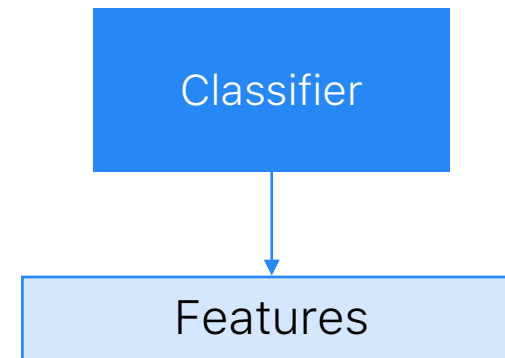
Как измерять качество генерации?

1) Глазами 👁️👁️

Как измерять качество генерации?

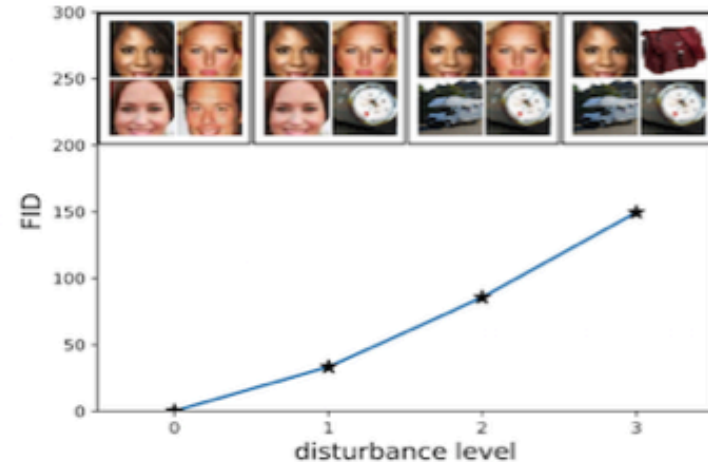
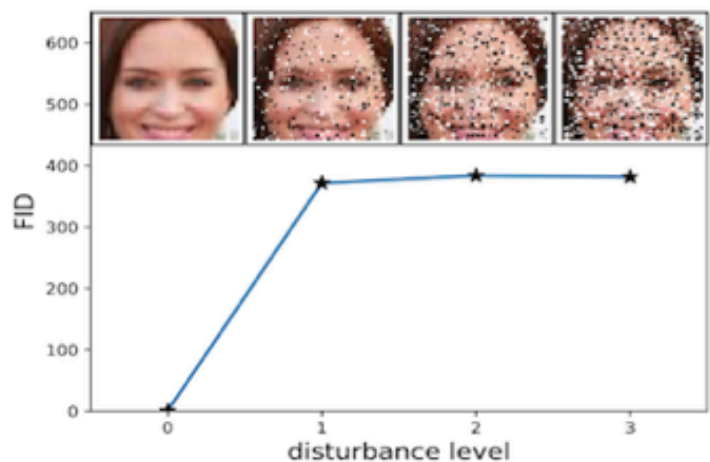
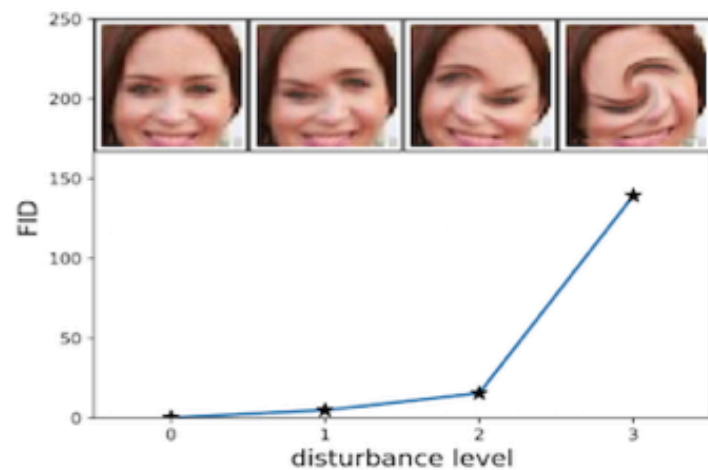
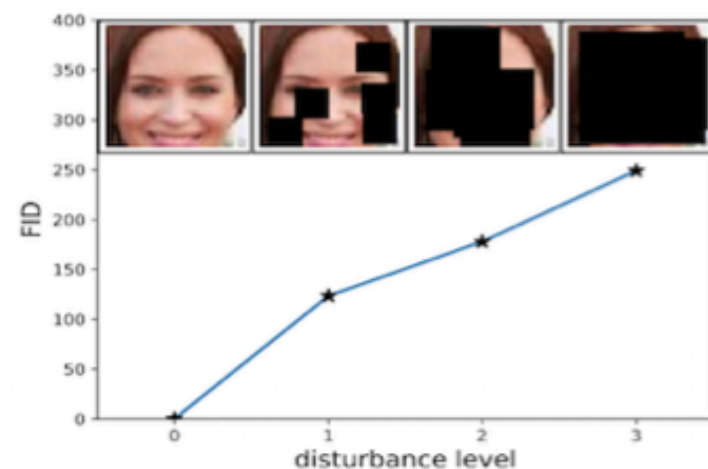
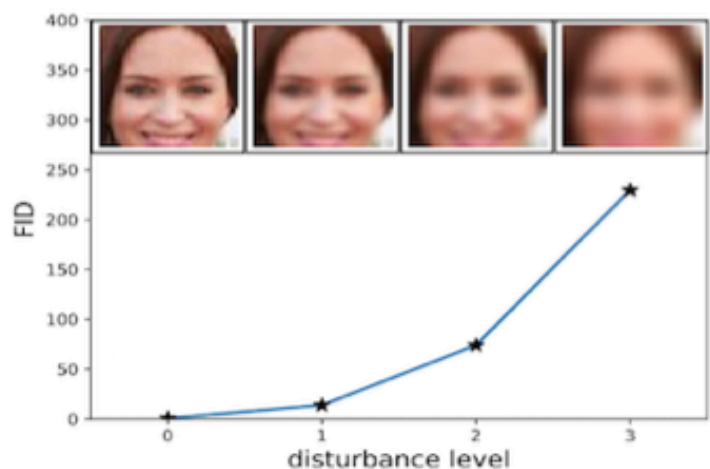
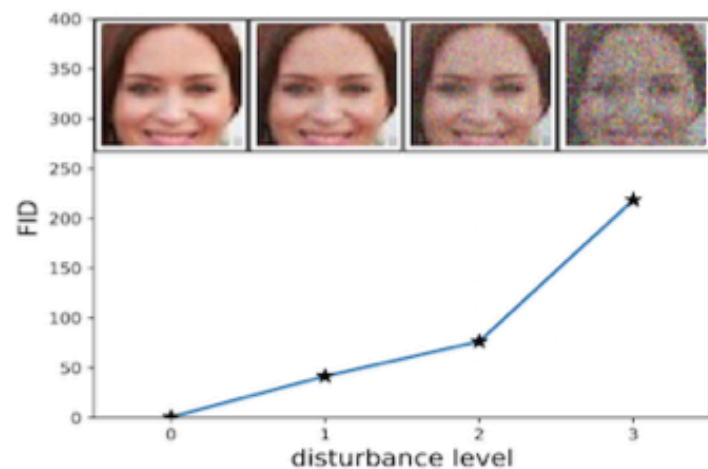
1) Глазами 👁️👁️

2) Fréchet Inception Distance



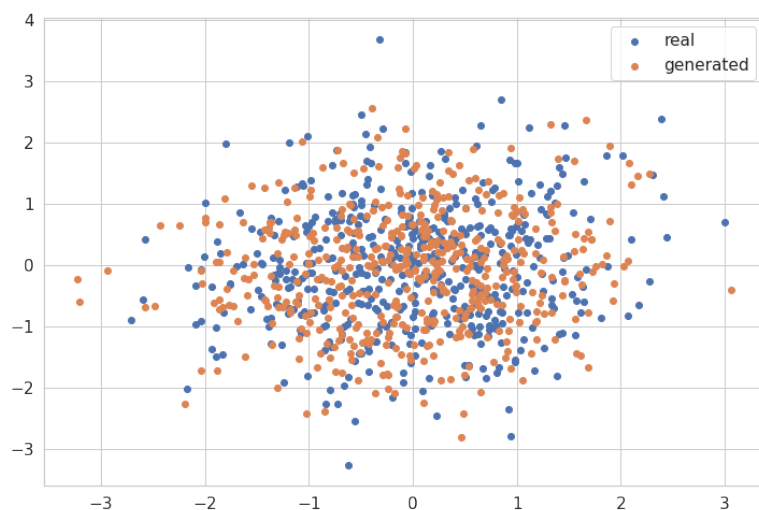
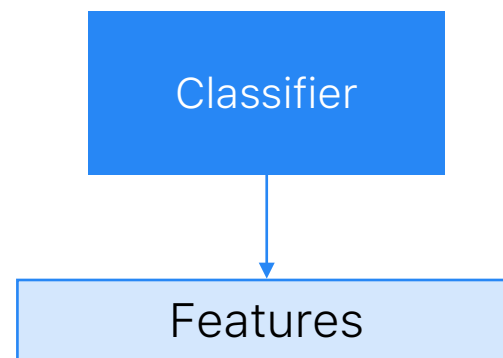
$$\text{FID}(x, g) = \left\| \mu_x - \mu_g \right\|_2^2 + \text{Tr} \left(\Sigma_x + \Sigma_g - 2 \left(\Sigma_x \Sigma_g \right)^{\frac{1}{2}} \right)$$

Как измерять качество генерации?

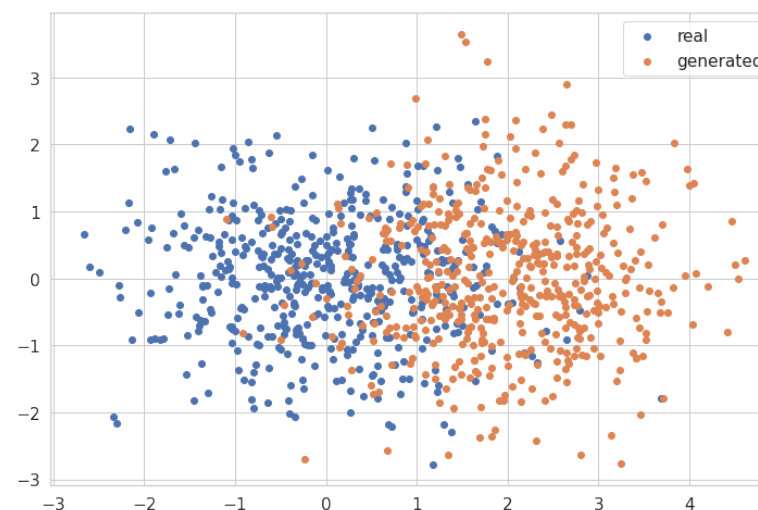


Как измерять качество генерации?

- 1) Глазами 👁️👁️
- 2) Fréchet Inception Distance
- 3) Leave-one-out 1-NN classifier accuracy (в идеале выдает 50%)



Хорошо

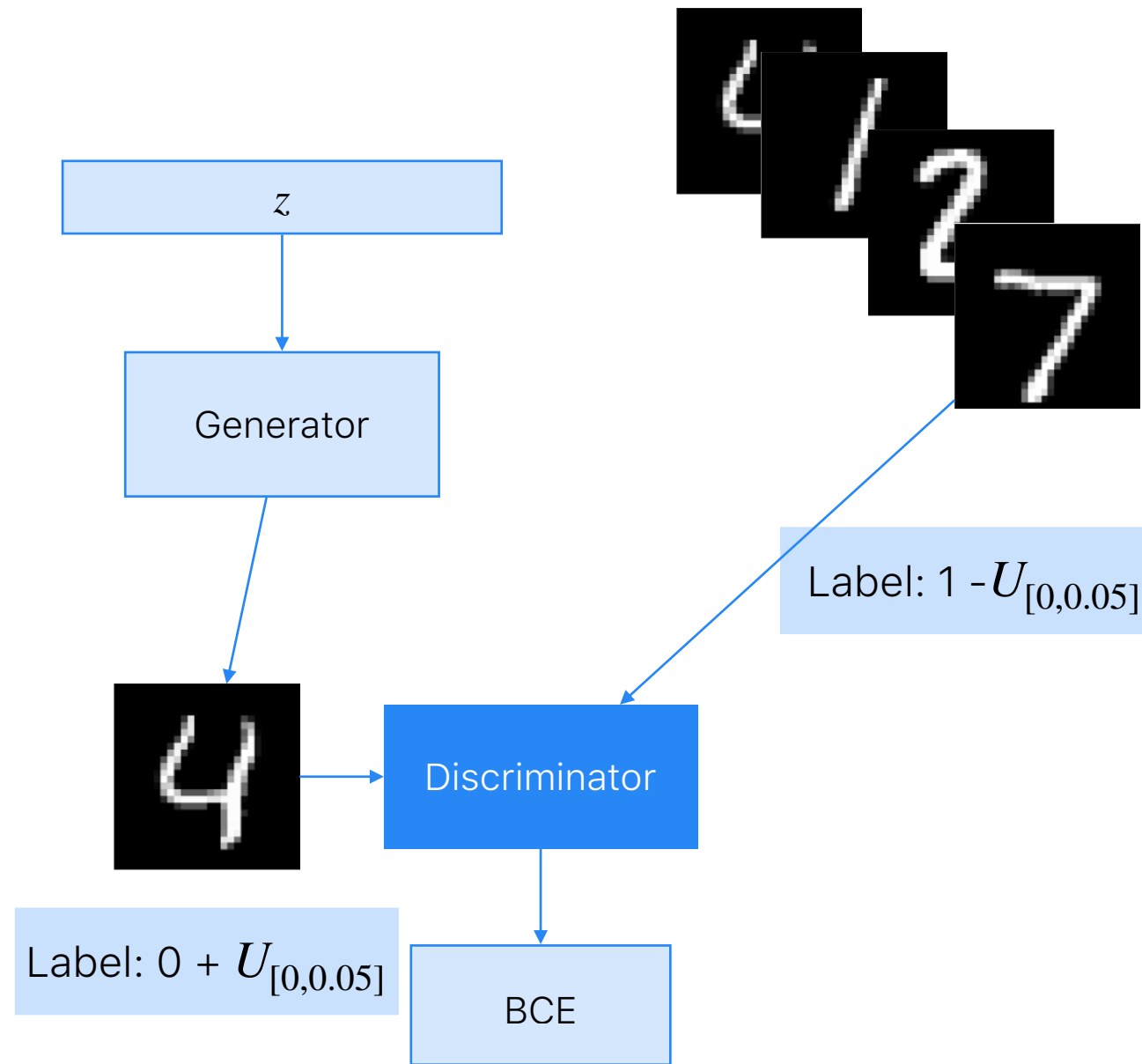


Плохо

Немного трюков

Обучение

При обучении вместо обычных меток можно добавлять зашумленные.



Обучение

При обучении вместо обычных меток можно добавлять зашумленные.

Можно использовать не BCE, а MSE loss.

Что это значит?

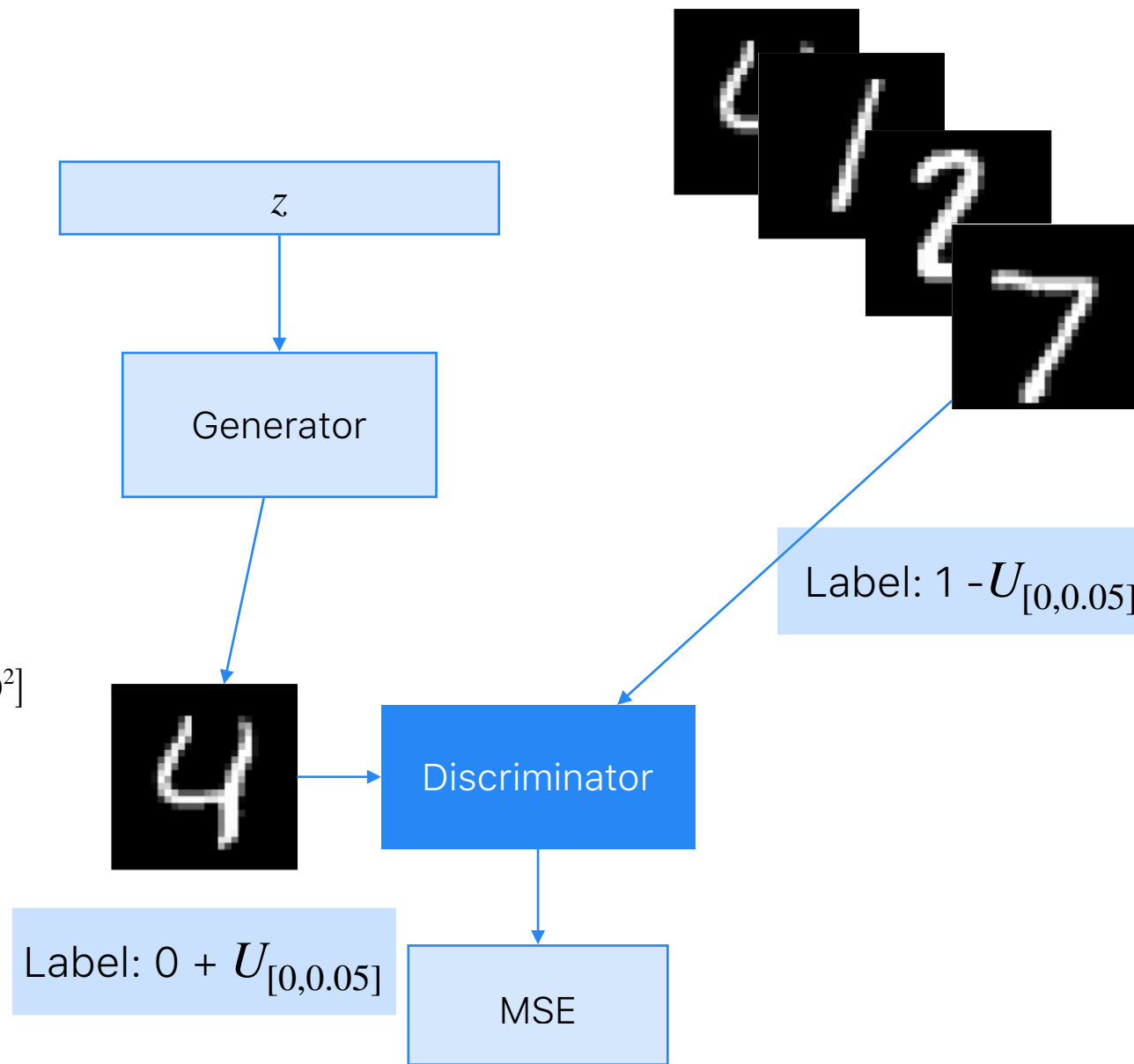
Вместо

$$L = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

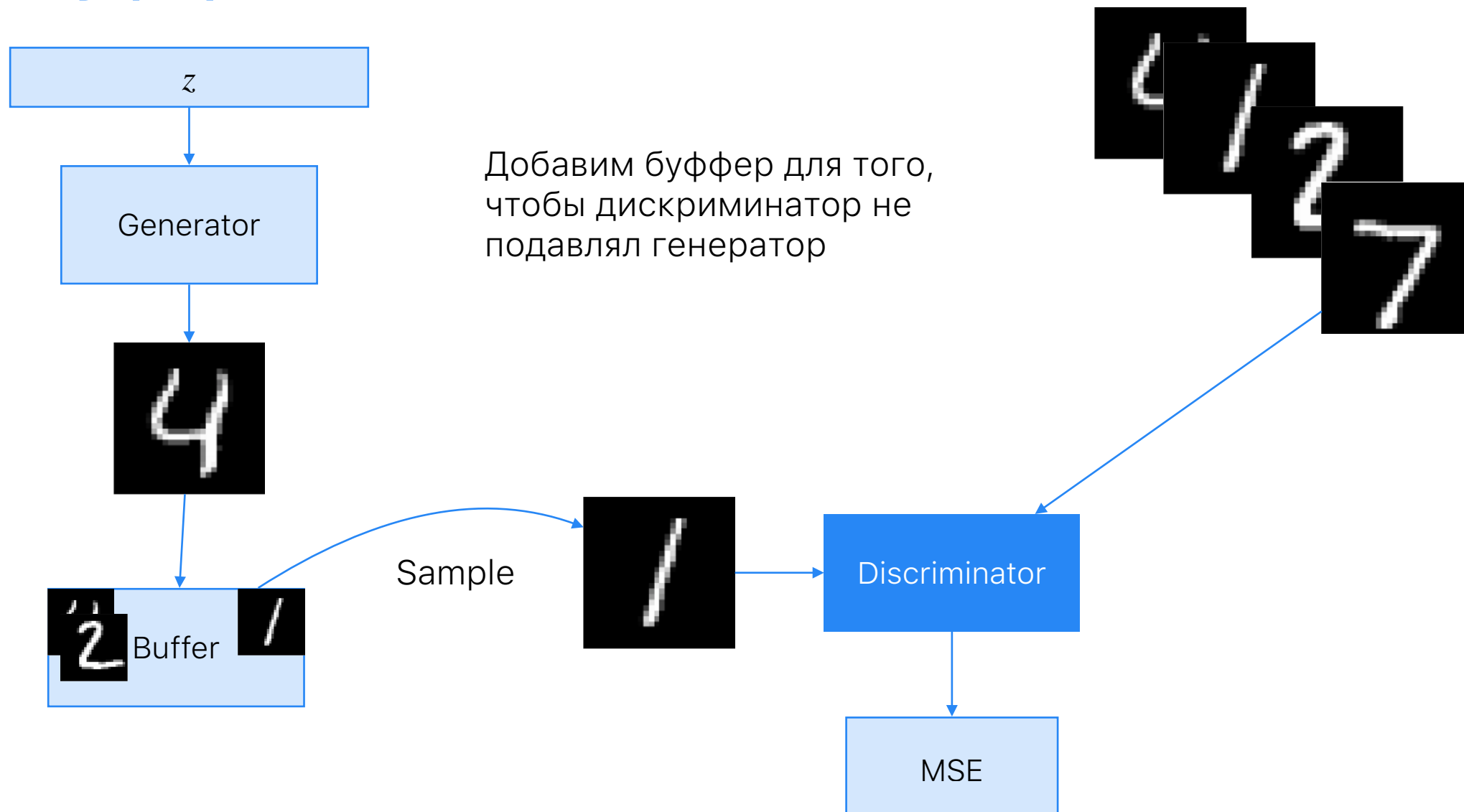
Мы теперь минимизируем

$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2} \mathbb{E}_{x \sim p_{\text{data}}(x)} [(D(x) - b)^2] + \frac{1}{2} \mathbb{E}_{z \sim p_z(z)} [(D(G(z)) - a)^2]$$

$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2} \mathbb{E}_{z \sim p_z(z)} [(D(G(z)) - c)^2]$$



Буфер

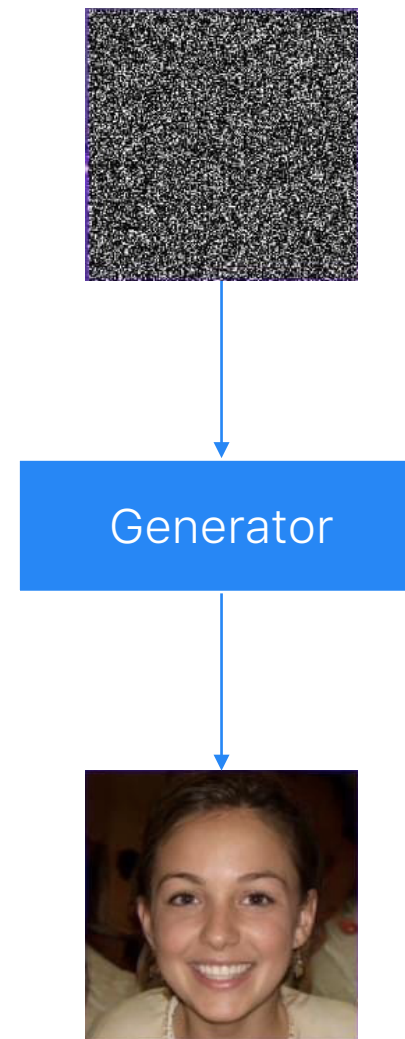


Архитектуры

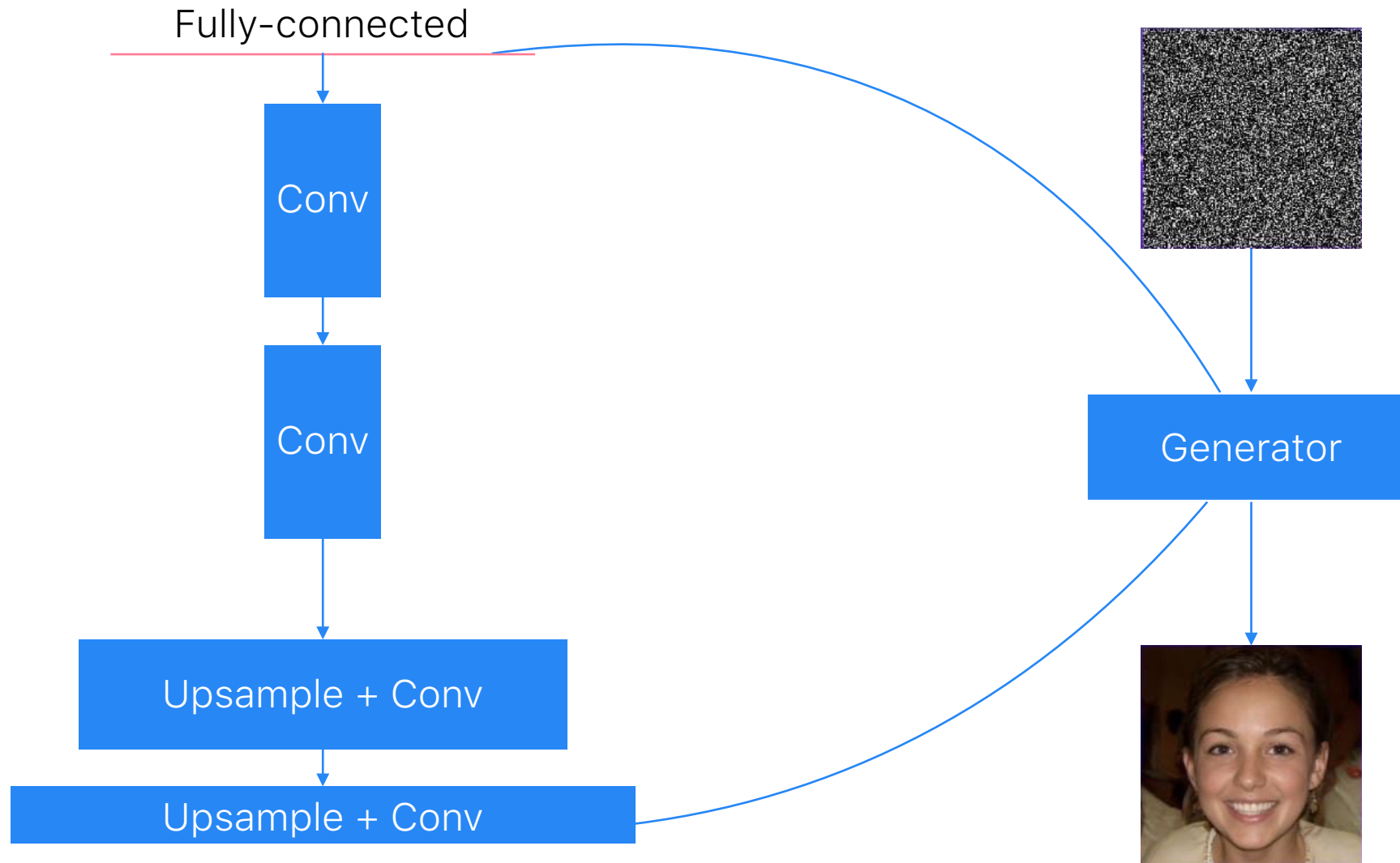


Классическая архитектура

Подаем на вход шум. На выходе получаем изображение из датасета.



Классическая архитектура



StyleGAN

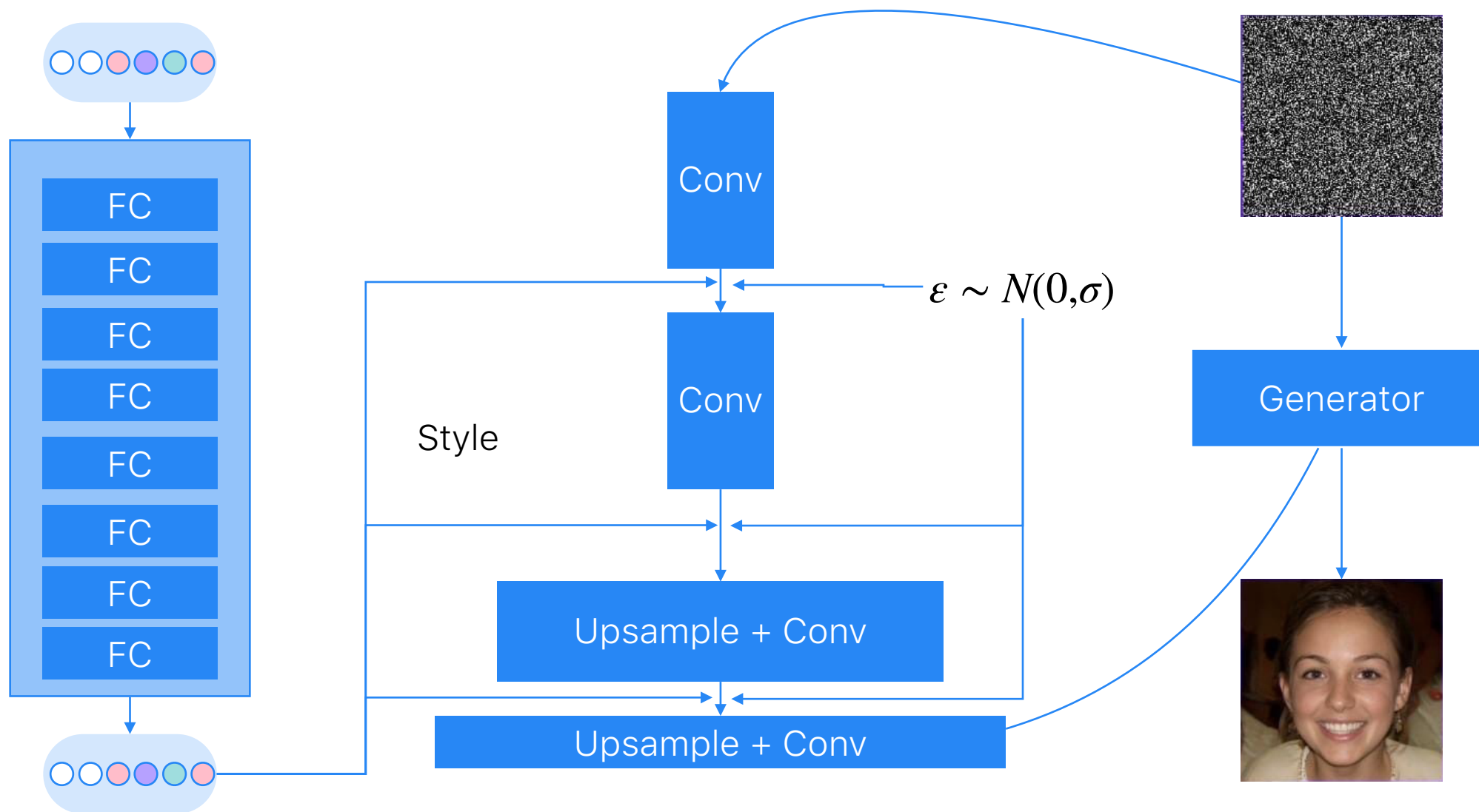
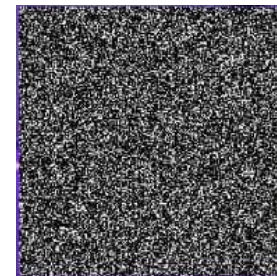


Image-to-image

На входе изображение



Generator

На выходе изображение

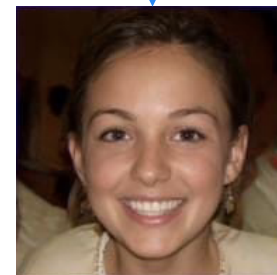


Image-to-image

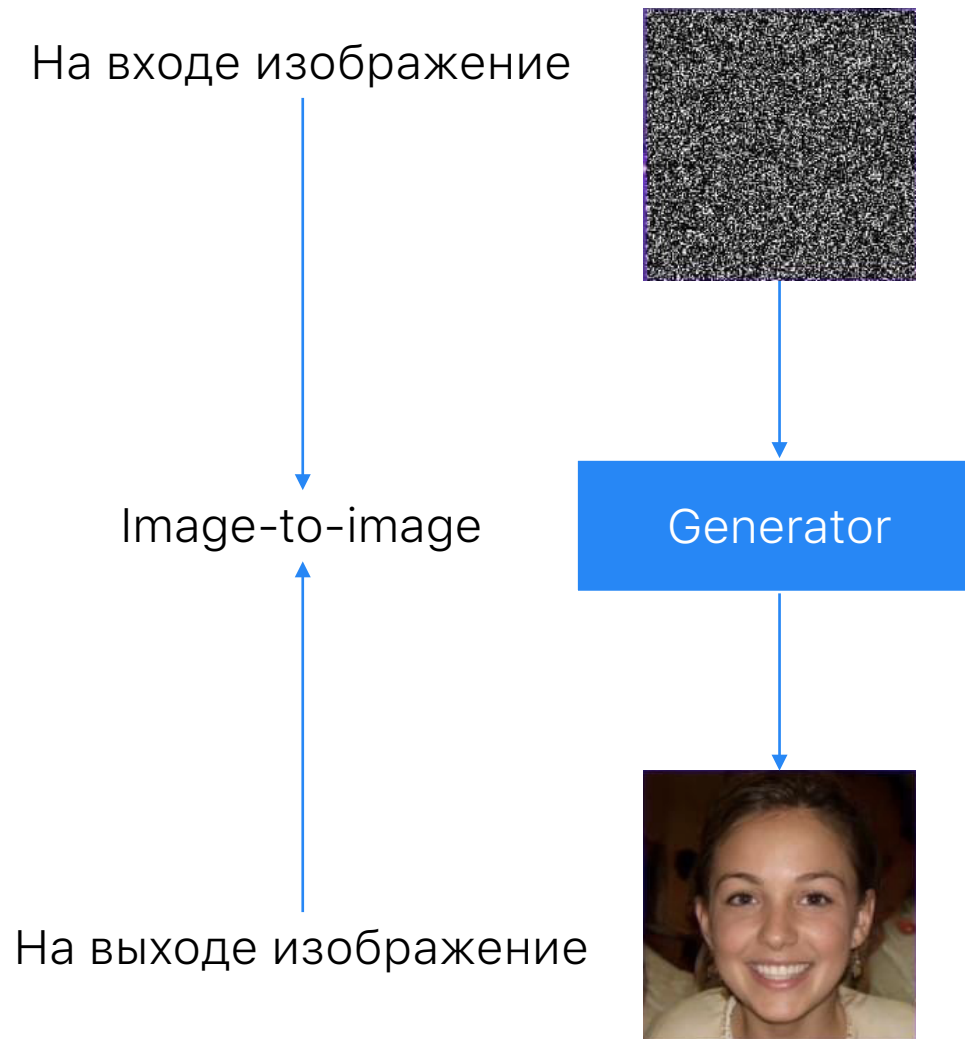


Image-to-image

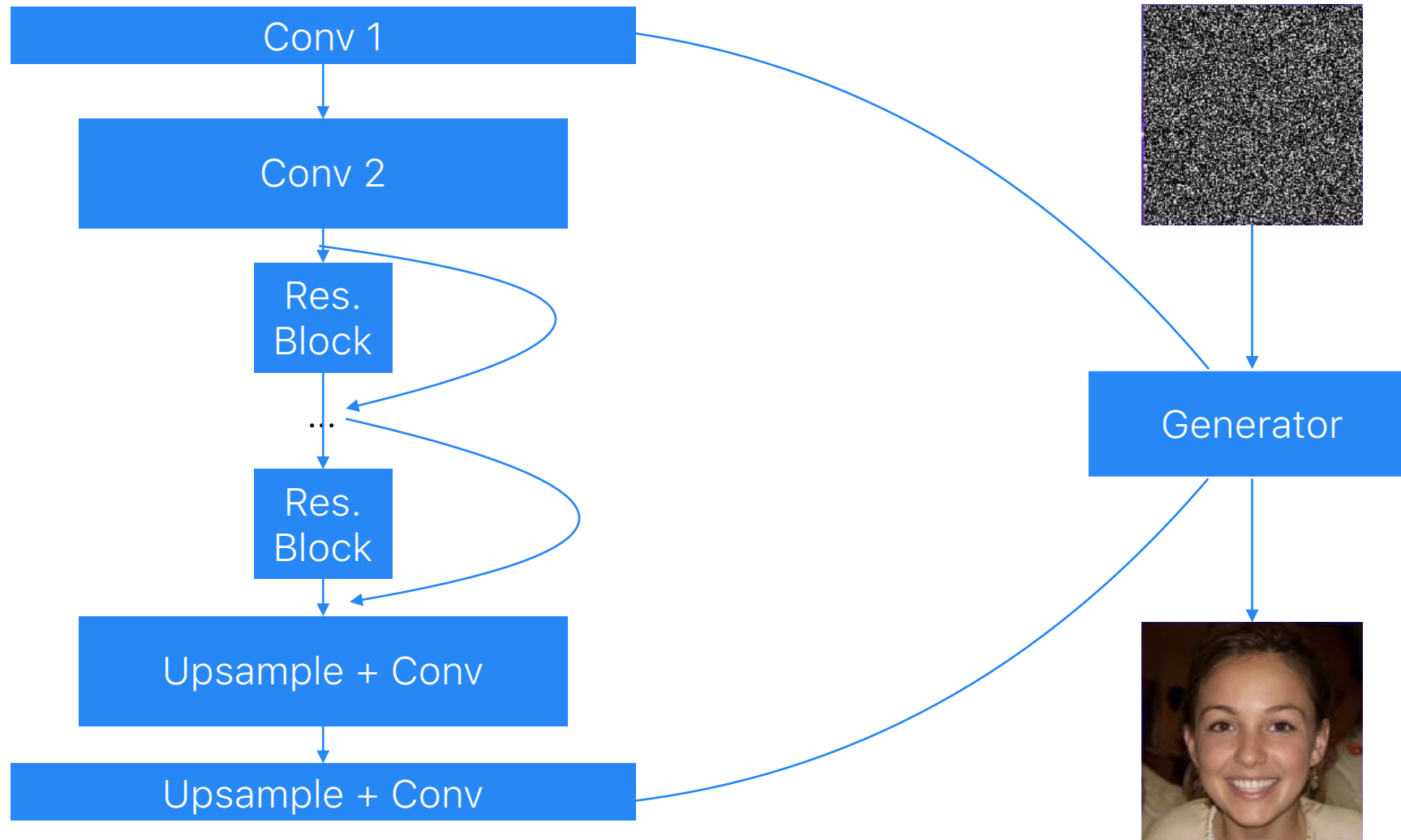
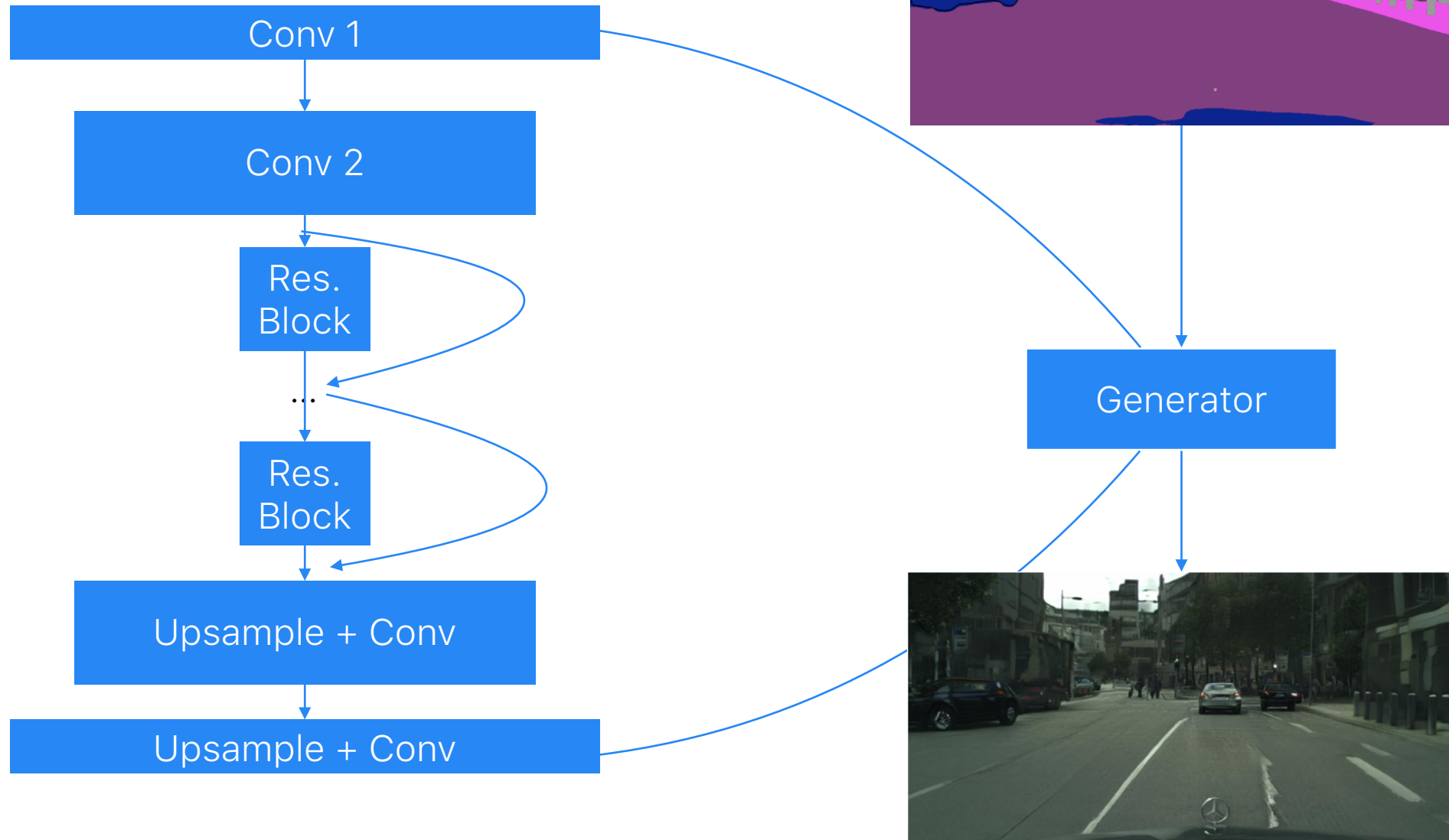
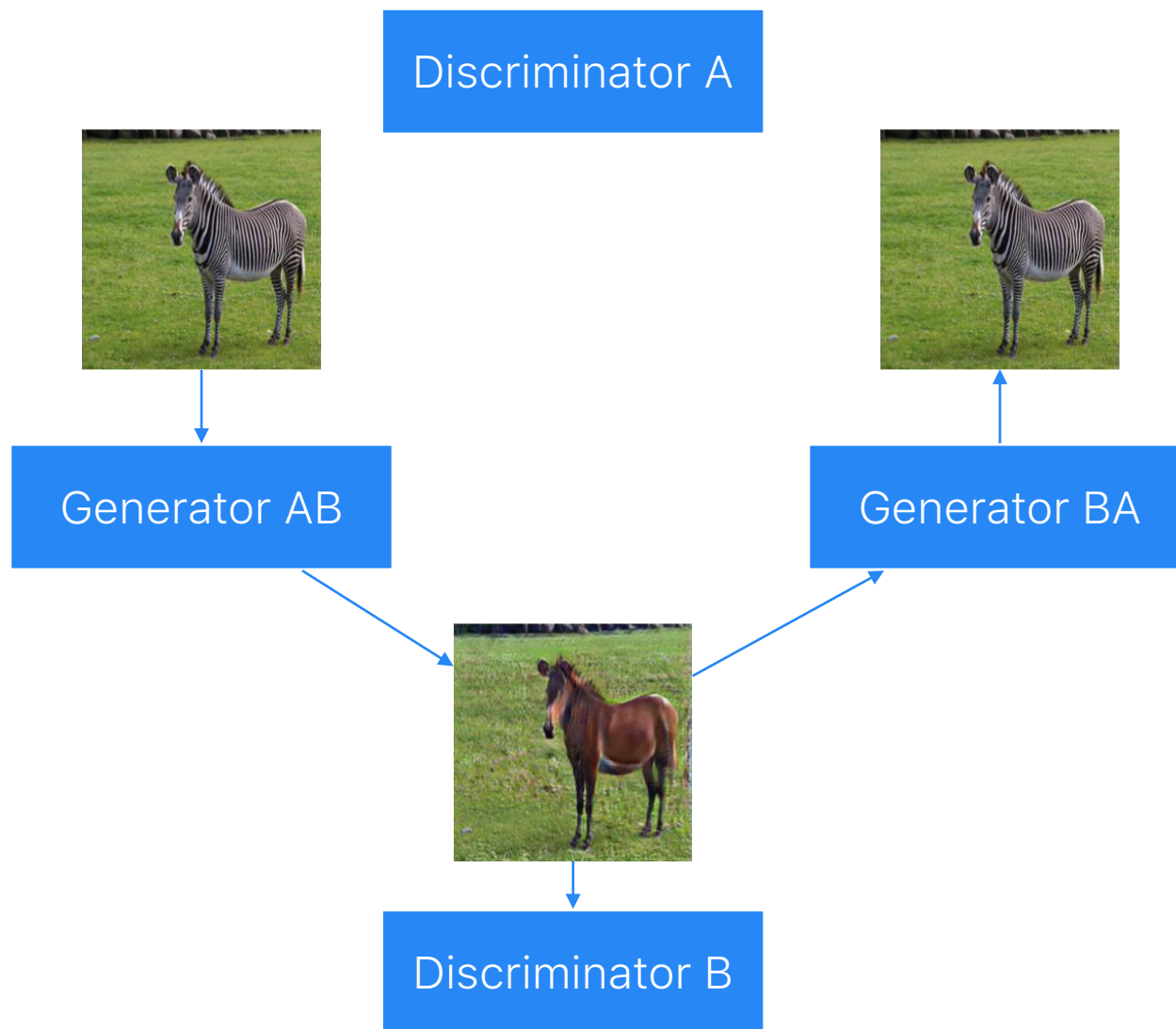


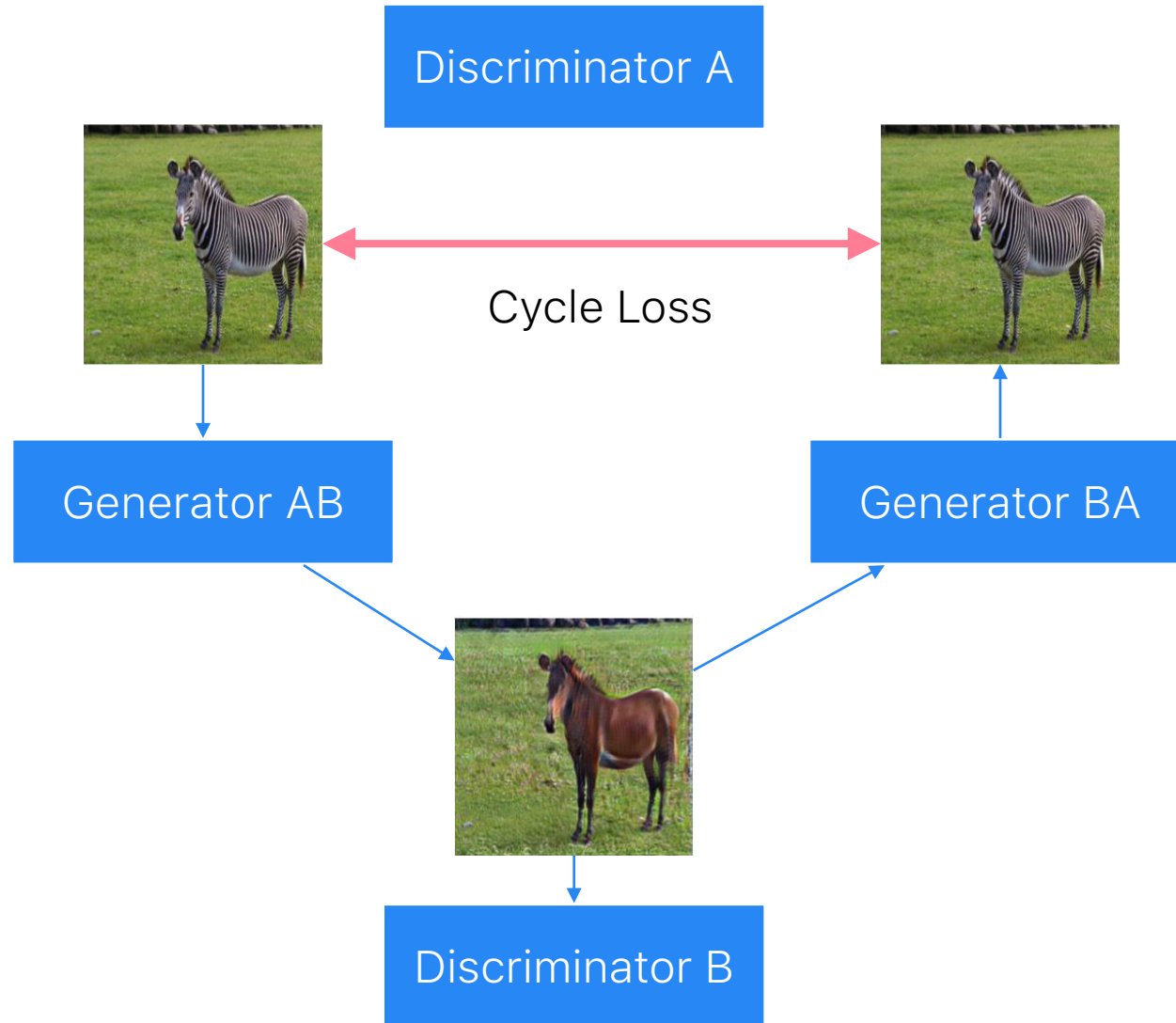
Image-to-image



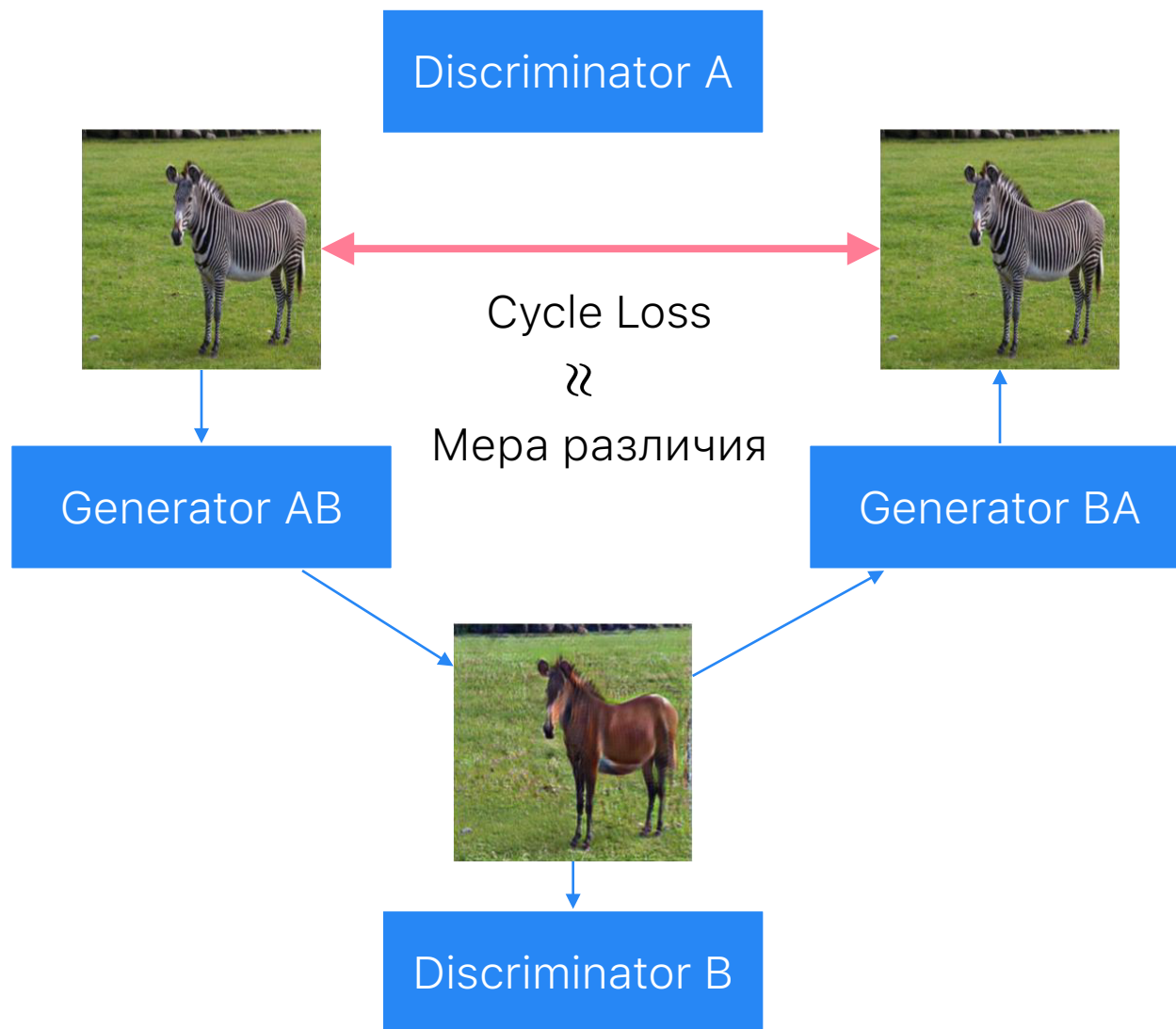
CycleGAN



CycleGAN



CycleGAN



Спасибо за внимание