

# Введение в генеративные модели

Андрей Фильченков  
руководитель ML Lab  
к.ф.-м.н.  
доц. ФИТИП ИТМО



# План выступления

---

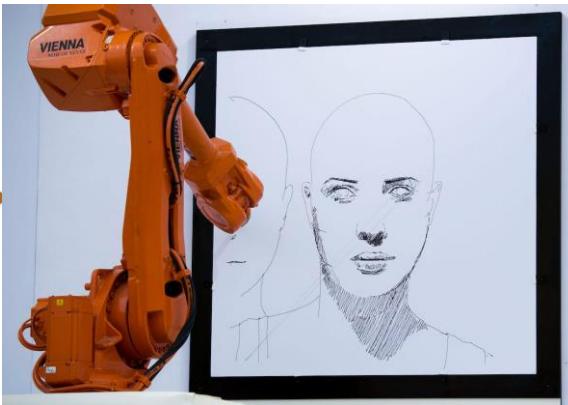
- Задачи дискриминации и генерации
- Задача генерация с точки зрения машинного обучения
- Как измерять качество генерации (изображений)
- Генеративно-состязательные сети
- Проблемы и решения
- Задача оптимальной транспортировки и функции потерь
- Интересные идеи в GANах
- Вариационные кодировщики
- Интересные идеи в VAE
- Немного примеров из жизни

# План выступления

---

- **Задачи дискриминации и генерации**
- Задача генерация с точки зрения машинного обучения
- Как измерять качество генерации (изображений)
- Генеративно-состязательные сети
- Проблемы и решения
- Задача оптимальной транспортировки и функции потерь
- Интересные идеи в GANах
- Вариационные кодировщики
- Интересные идеи в VAE
- Немного примеров из жизни

# Что хотели от искусственного интеллекта?



# Что получалось в реальности

---

- Диагностировать заболевания
- Классифицировать почву
- Предсказывать объем продаж
- Предсказывать стоимость недвижимости
- Строить скоринговые модели
- Классифицировать тексты по категориям
- ...

# Задание №1

---

- Подумав одну минуту, ответьте:

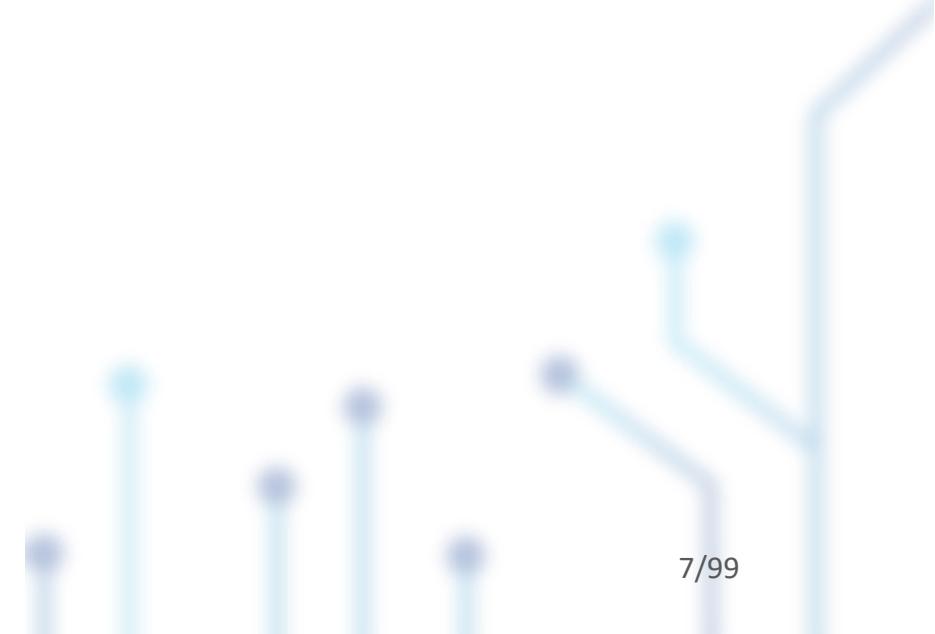
**На картинке изображен Петербург на закате?**



## Задание №2

---

- За одну минуту нарисуйте Петербург на закате



# Дискrimинация vs генерация

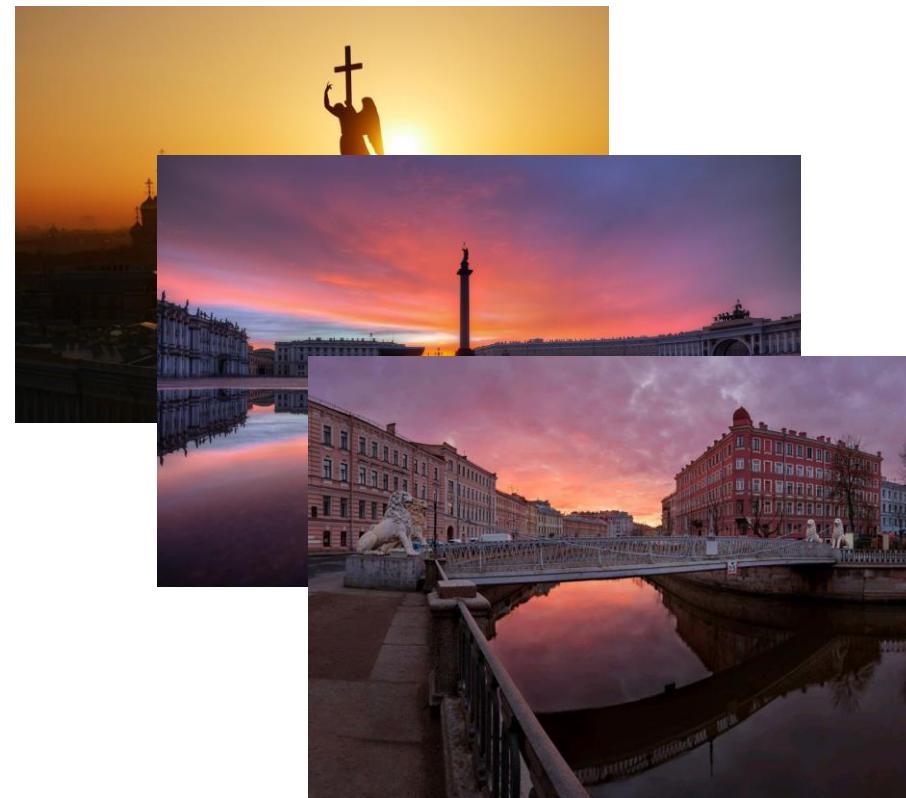
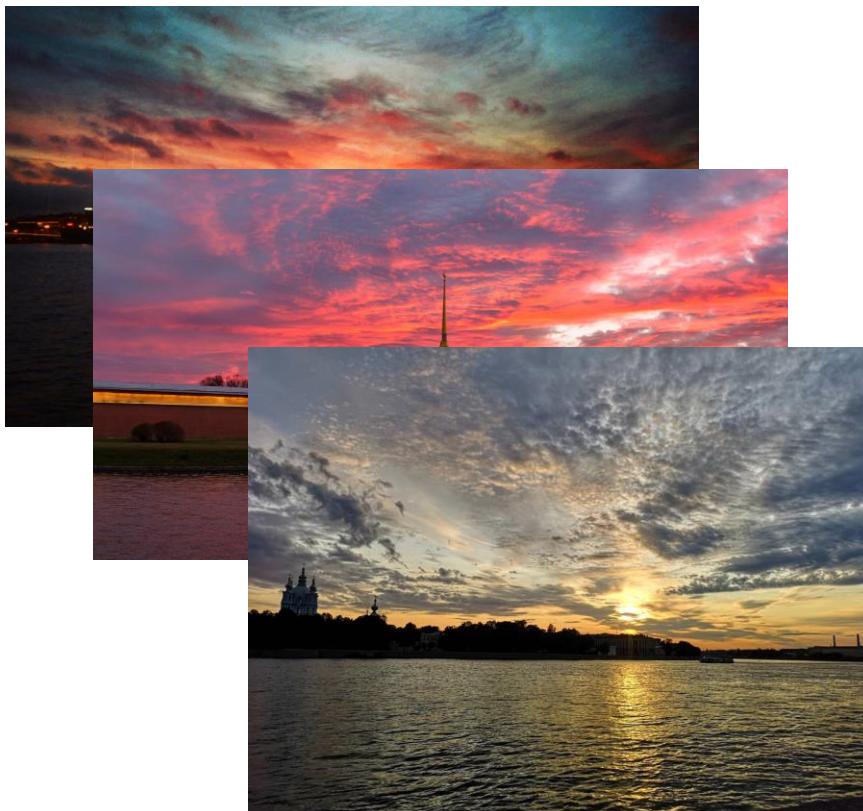
---

- Отличать объекты разных классов — **задача дискrimинации**
- Порождать новые объекты — **задача генерации**
- Задача генерации принципиально сложнее задачи дискrimинации и долгое время относилась к категории сильного искусственного интеллекта

# Задача дискриминации

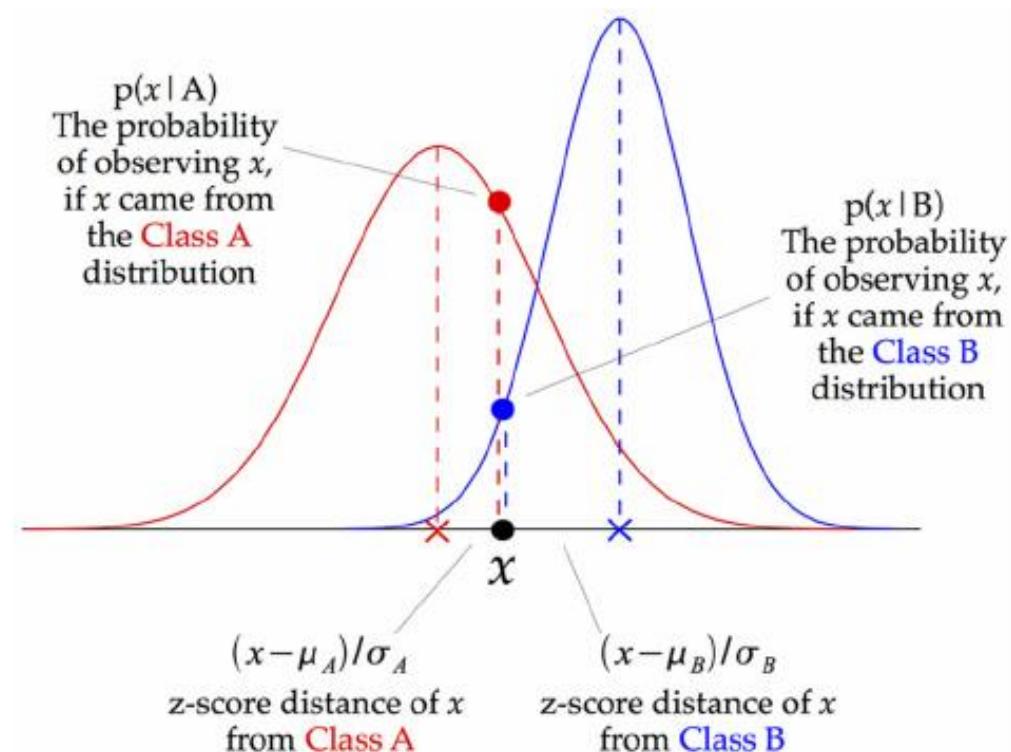
---

Сначала соберем обучающую выборку



# Обучение дискриминации (байес)

- Изображения в каждом классе как-то распределены
- Нам необходимо восстановить эти распределения, а также априорные вероятности классов, чтобы построить оптимальную разделяющую поверхность



# План выступления

---

- Задачи дискриминации и генерации
- **Задача генерация с точки зрения машинного обучения**
- Как измерять качество генерации (изображений)
- Генеративно-состязательные сети
- Проблемы и решения
- Задача оптимальной транспортировки и функции потерь
- Интересные идеи в GANах
- Вариационные кодировщики
- Интересные идеи в VAE
- Немного примеров из жизни

# Задача генерации новых объектов

---

К какому типу задач относится задача генерации новых объектов?

- Обучение с учителем
- Обучение без учителя
- Частичное обучение
- Обучение с подкреплением

# Задача генерации новых объектов

---

- К какому типу задач относится задача генерации новых объектов?
- Это зависит от того, как мы определяем **новизну объекта** и как мы измеряем **степень принадлежности объекта к желаемому множеству**
- Ничто не мешает создать набор эмпирических правил, при помощи которых мы будем создавать новые объекты
- Тогда не нужно никакого машинного обучения



# Задача генерации с точки зрения ML

---

Мы хотим, чтобы

- сгенерированное изображение было изображением
- было изображением из предметной области:
  - все детали изображения соответствовали бы предметной области
  - взаимодействие между деталями было бы реалистично
  - ...

Для сложных объектов мы не можем это эксплицитно описать.

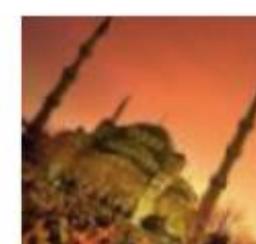
# Генерация это обучение без учителя

---

- Нам необходимо восстановить скрытую структуру зависимостей в объектах
  - У нас нет для этого разметки
- 
- Это задача **обучения без учителя**
  - На деле, мы будем восстанавливать **распределение над объектами**

# Задача генерации

Получив обучающую выборку, необходимо восстановить распределение. Тогда генерация — просто сэмплирование из восстановленного распределения.



Обучающая выборка  $p_{\text{data}}(x)$

Сгенерированная выборка  $p_{\text{model}}(x)$

$p_{\text{model}}(x)$  должно быть очень похоже на  $p_{\text{data}}(x)$

# Как измерить похожесть распределений

---

**Дивергенция Кульбака—Лейблера:**

Для  $P$ , распределенного согласно  $p$ , и  $Q$ , распределенного согласно  $q$ ,

$$D_{\text{KL}}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)}$$

Также называется **относительной энтропией**  $P$  к  $Q$

Важно, что оно несимметрично, поэтому не является расстоянием

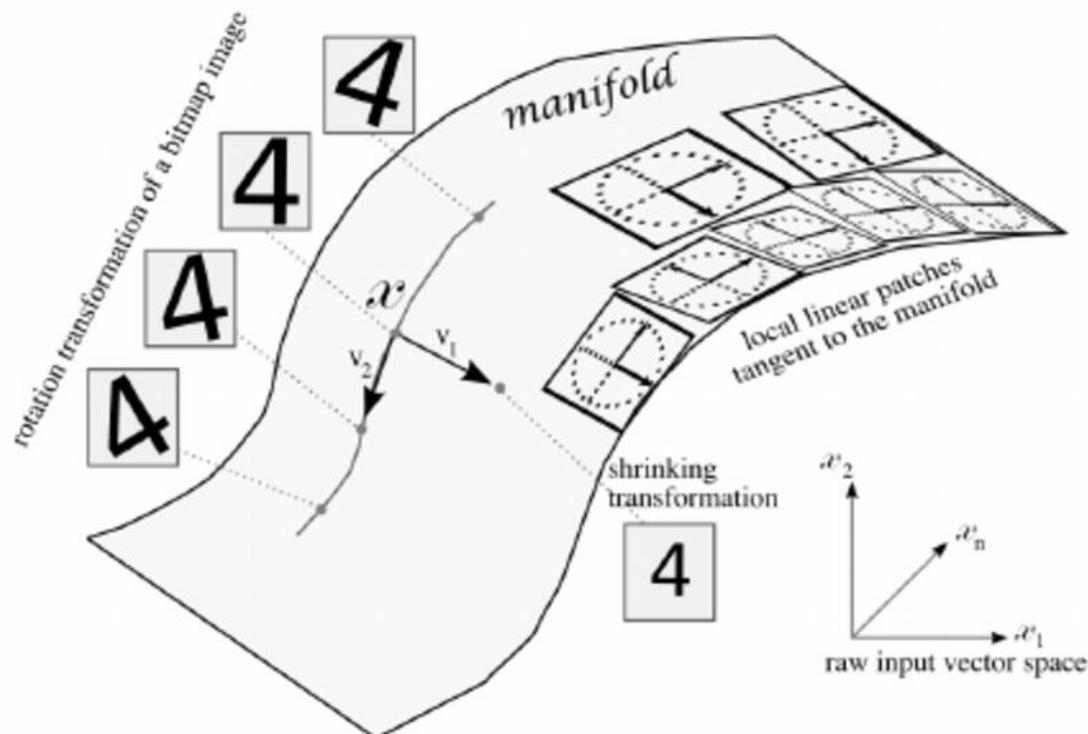
# Сложности

---

- Оценка плотности распределения — основная проблема обучения без учителя (да и статистики в целом)
- Чем выше размерность пространства, тем сложнее восстанавливать многомерные распределения

# Гипотеза многообразия

Многомерные данные из реального мира лежат в низкоразмерном многообразии внутри соответствующего многомерного пространства



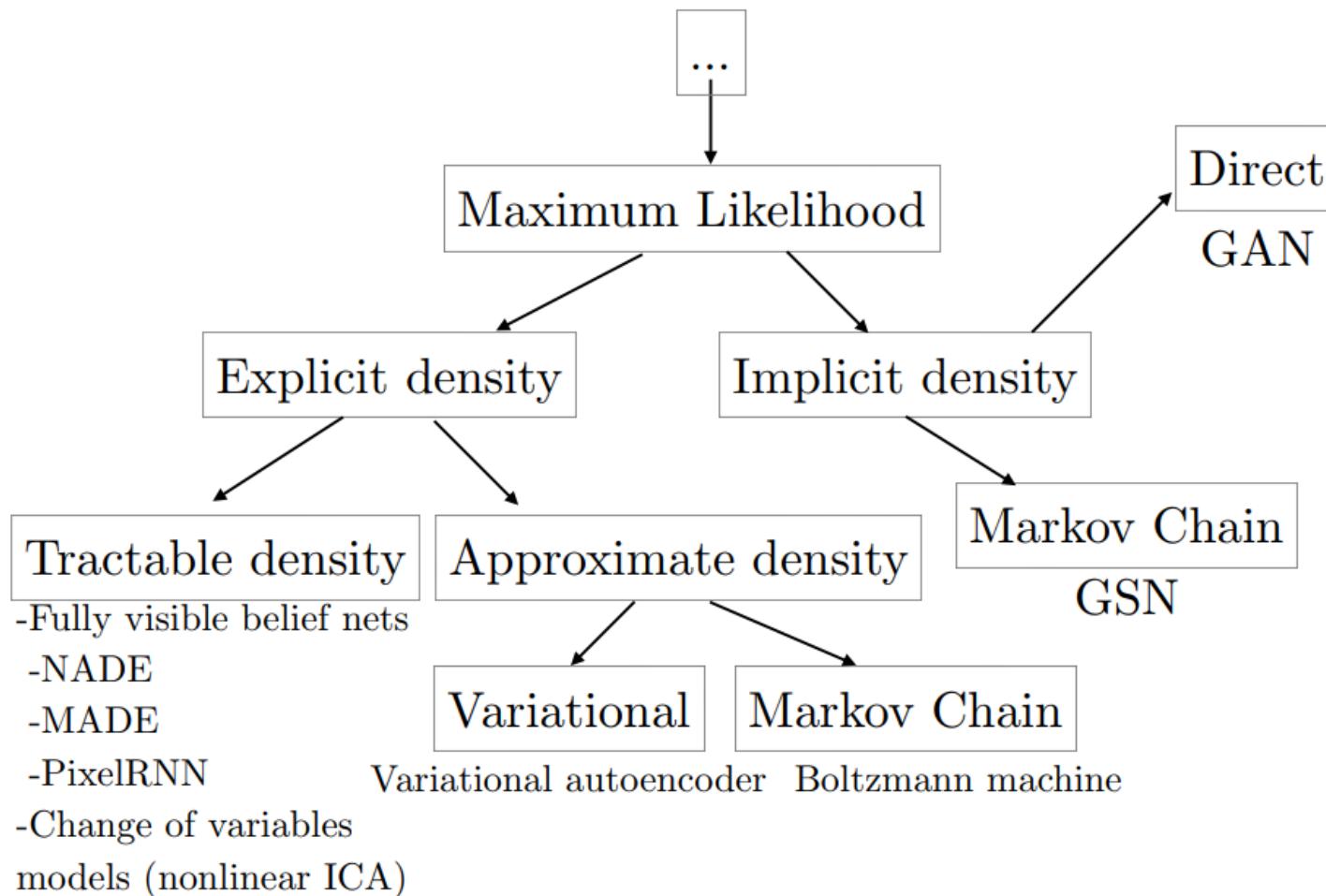
# Варианты восстановления плотностей

---

Два основных способа:

- **Непосредственное восстановление плотности:** в явном виде задаем  $p_{\text{model}}(x)$
- **Неявное восстановление плотности:** обучаем модель, которая может сэмплировать согласно  $p_{\text{model}}(x)$  без непосредственного задания этого распределения

# Таксономия моделей



# План выступления

---

- Задачи дискриминации и генерации
- Задача генерация с точки зрения машинного обучения
- **Как измерять качество генерации (изображений)**
- Генеративно-состязательные сети
- Проблемы и решения
- Задача оптимальной транспортировки и функции потерь
- Интересные идеи в GANах
- Вариационные кодировщики
- Интересные идеи в VAE
- Немного примеров из жизни

# Inception score

---

- Максимизируем качество сгенерированных изображений:  
 $p(y|x = G(z))$  должно быть хорошо предсказуемо
- Максимизируем разнообразие:

$$\int_z p(y|x = G(z)) dz$$

Комбинируем:

$$IS(G) = \exp \left( \mathbb{E}_{\mathbf{x} \sim p_a} D_{KL}( p(y|\mathbf{x}) \| p(y) ) \right),$$

Для оценки  $p(y|x = G(z))$  используем обученную сеть Inception (отсюда название)

# Fréchet Inception Distance (FID)

---

- Измеряем расстояние между картами признаков у настоящих и сгенерированных изображений, предполагая, что они распределены по Гауссу:

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

где  $\text{Tr}$  — след матрицы.

# План выступления

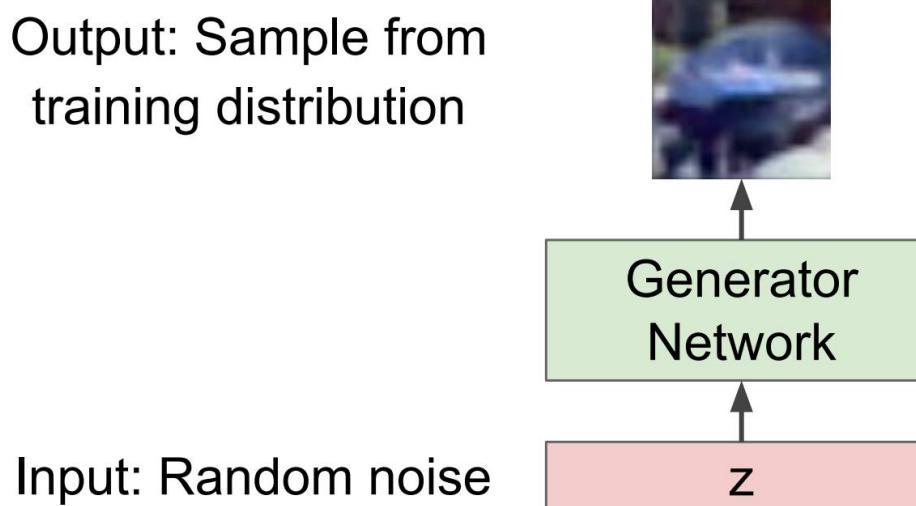
---

- Задачи дискриминации и генерации
- Задача генерация с точки зрения машинного обучения
- Как измерять качество генерации (изображений)
- **Генеративно-состязательные сети**
- Проблемы и решения
- Задача оптимальной транспортировки и функции потерь
- Интересные идеи в GANах
- Вариационные кодировщики
- Интересные идеи в VAE
- Немного примеров из жизни

# Основные постулаты GANов

---

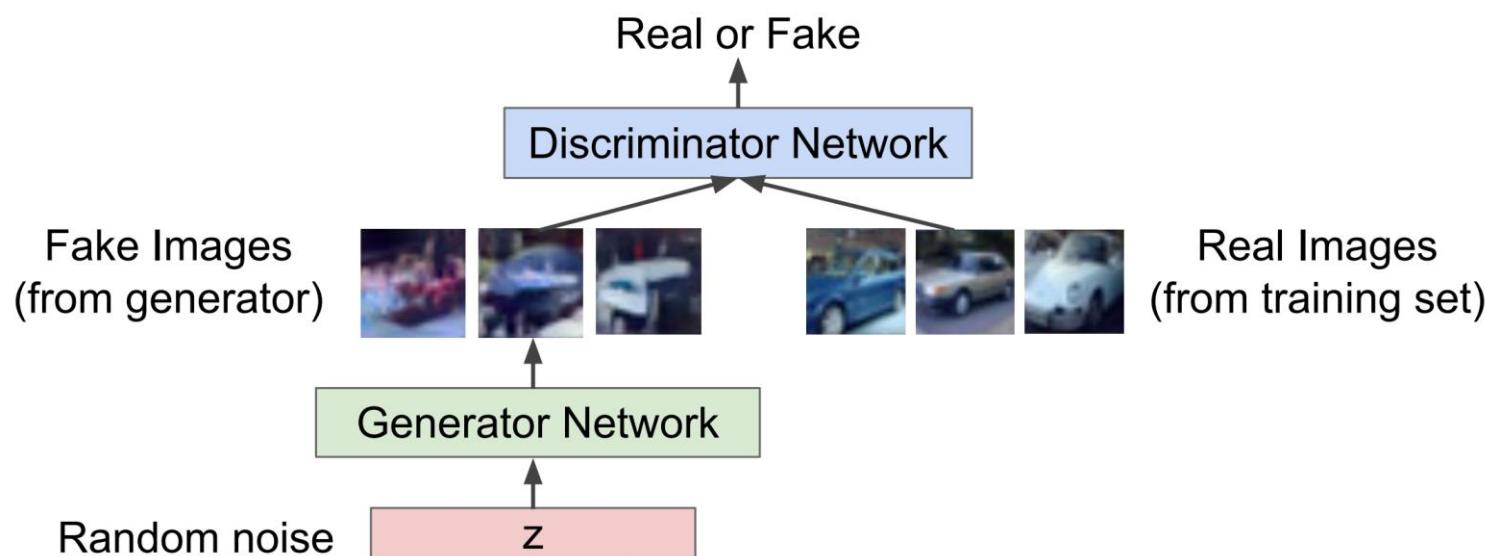
- Не будем пытаться непосредственно представить распределения
- Будем обучать преобразование в объекты из случайного шума
- Будем использовать теоретико-игровой подход



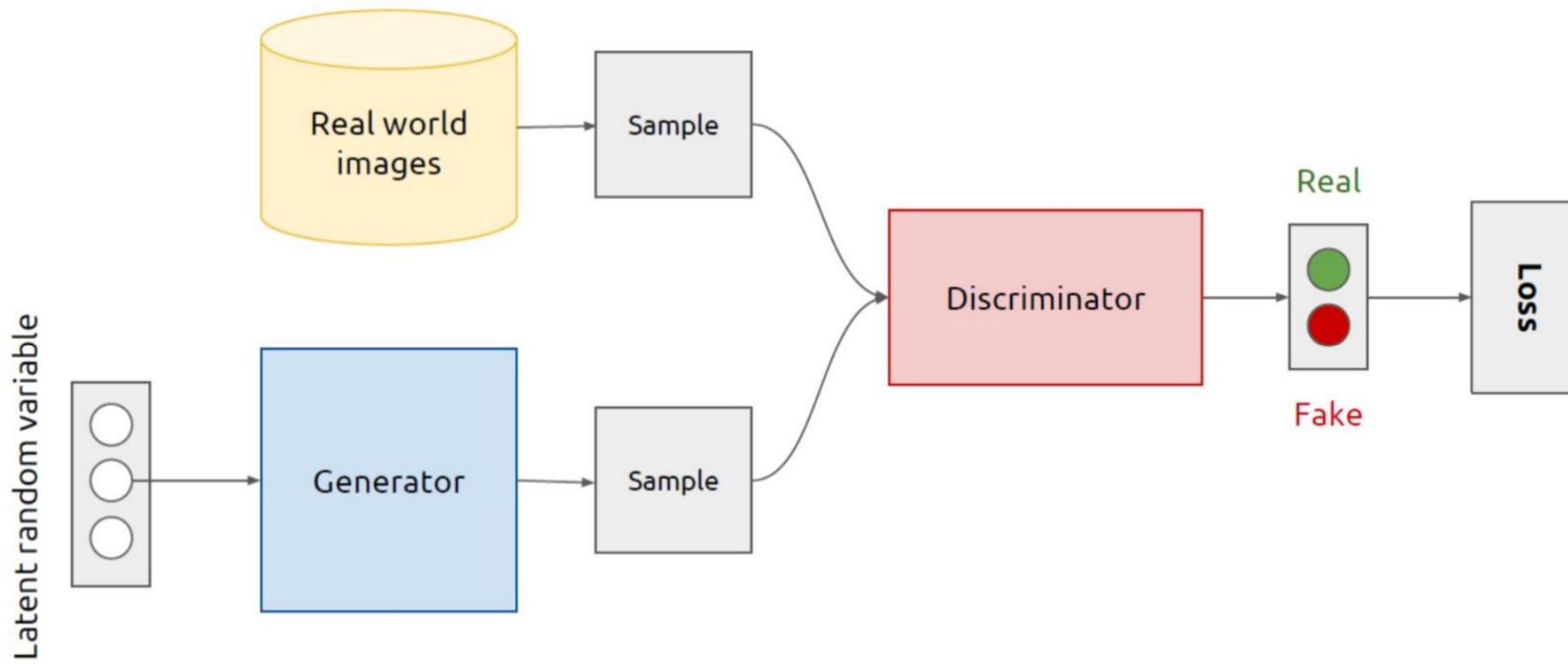
# Генератор vs Дискриминатор

- **Генератор** — сеть, которая будет генерировать объекты
- **Дискриминатор** — сеть, которая будет отличать сгенерированные объекты от настоящих

Обучение будет построено как противоборство двух этих «игроков»



# Обучение GANов



# Минимаксная игра

---

Можно обучать совместно в постановке минимаксной игры

**Минимаксная целевая функция:**

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

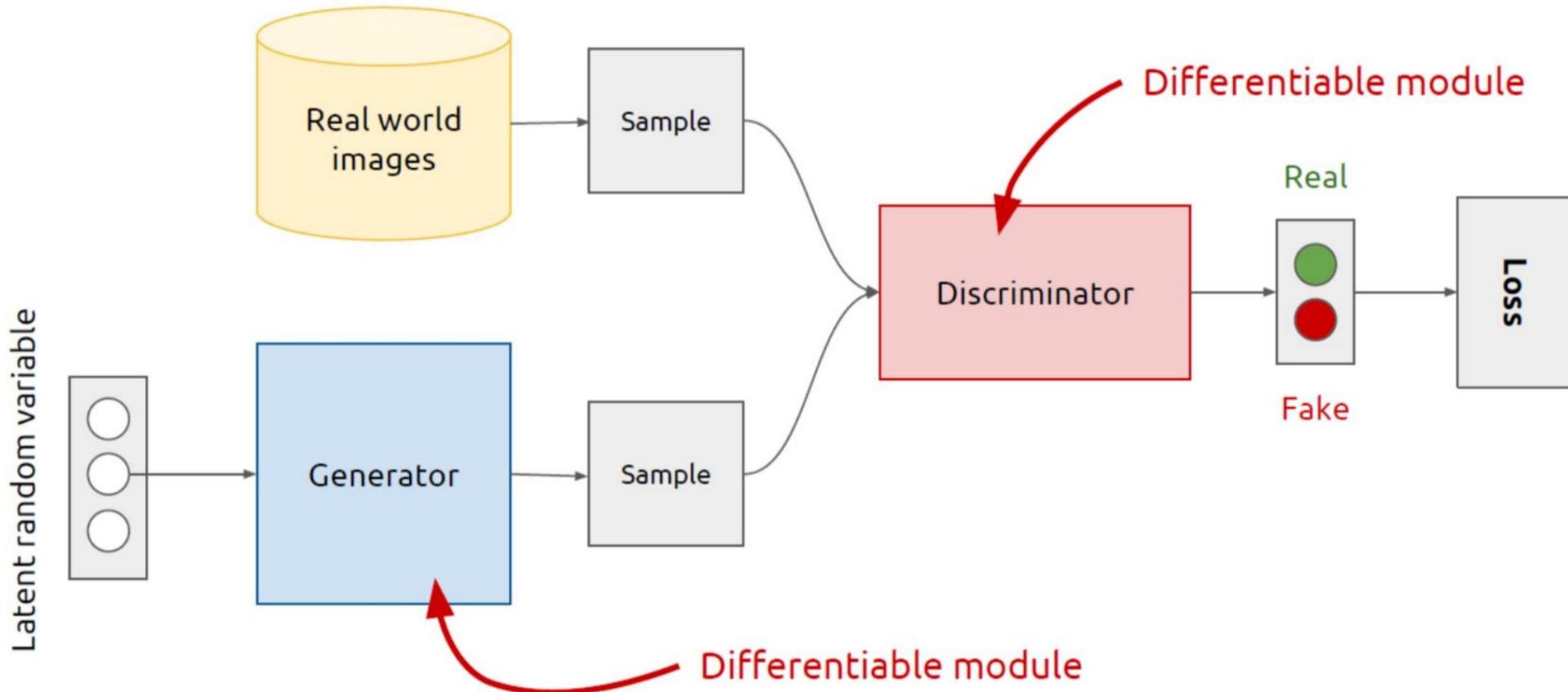
где  $D_{\theta_d}$  – дискриминатор с параметрами  $\theta_d$

(пытается максимизировать целевую функцию так, и сделать так, чтобы  $D(x)$  был близок к 1 (настоящий), а  $D(G(z))$  был близок к 0 (сгенерированный))

и  $G_{\theta_g}$  – генератор с параметрами  $\theta_g$

(пытается минимизировать целевую функцию и сделать  $D(G(z))$  близким к 1)

# Обучение GANов обратным распространением ошибки



# Поочередное обучение

---

Поочередно:

**1. Градиентный подъем по дискриминатору**

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

**2. Градиентный спуск по генератору**

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

# Более хорошее поочередное обучение

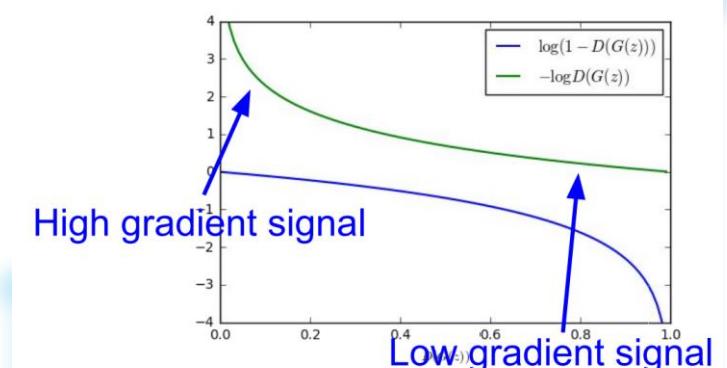
Поочередно:

## 1. Градиентный подъем по дискриминатору

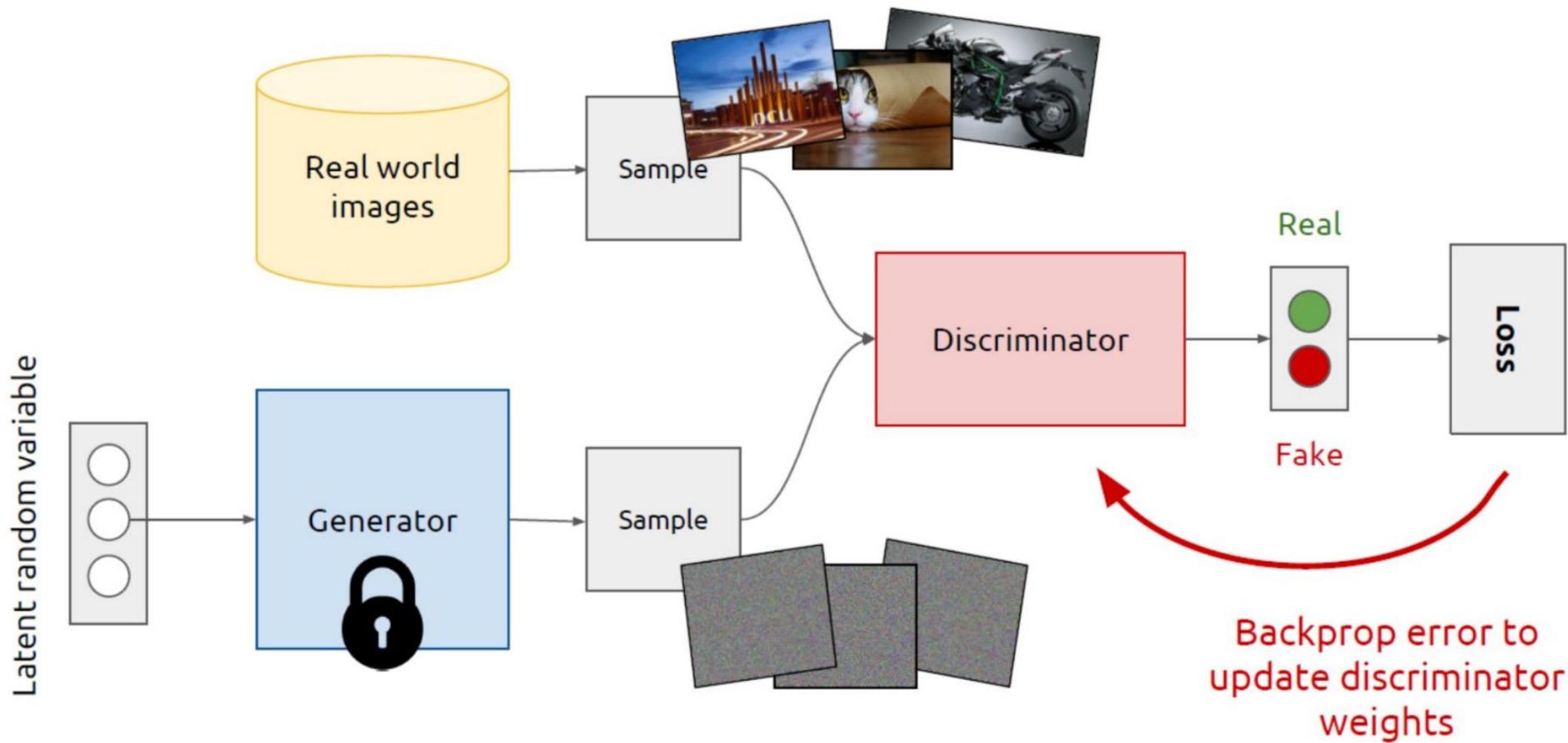
$$\max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{\text{data}}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

## 2. Градиентный подъем по генератору

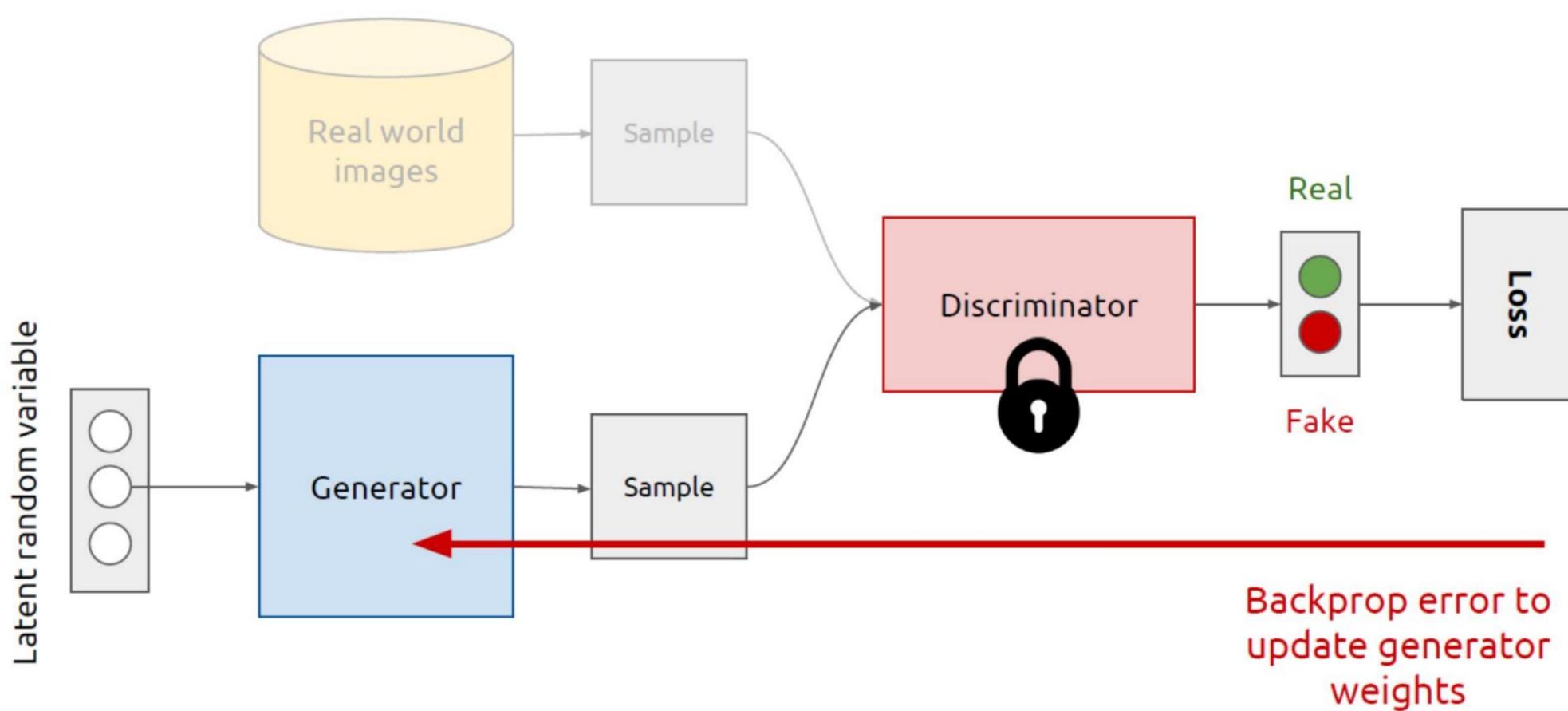
$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$



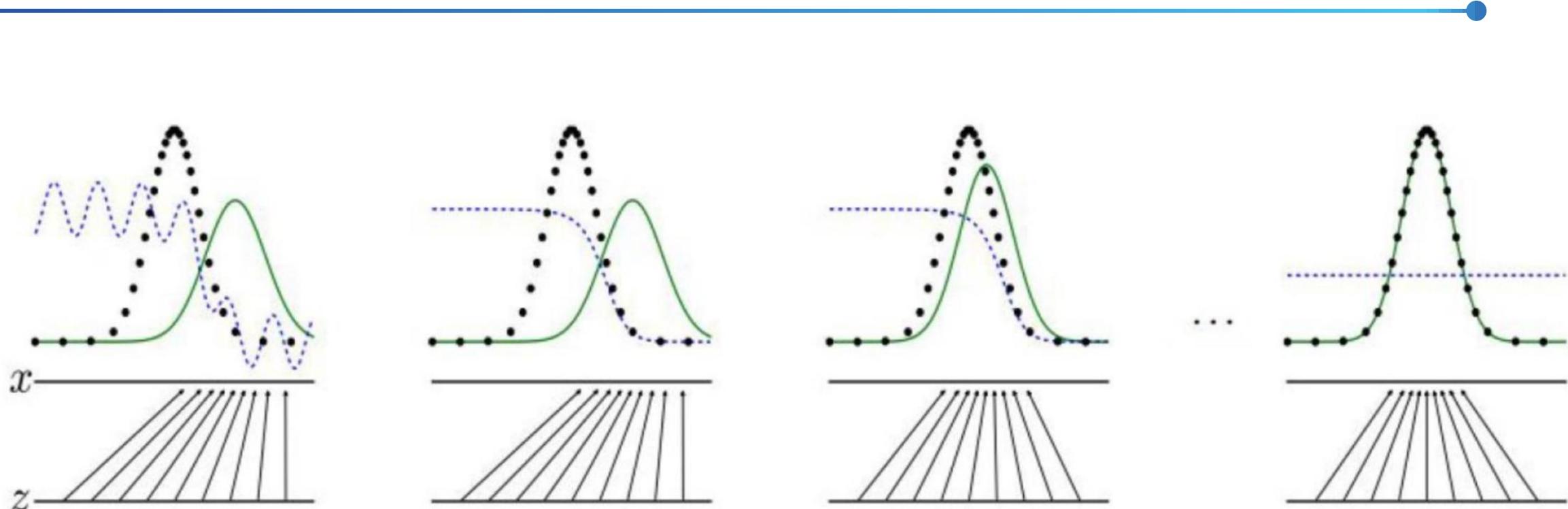
# Обучение дискриминатора



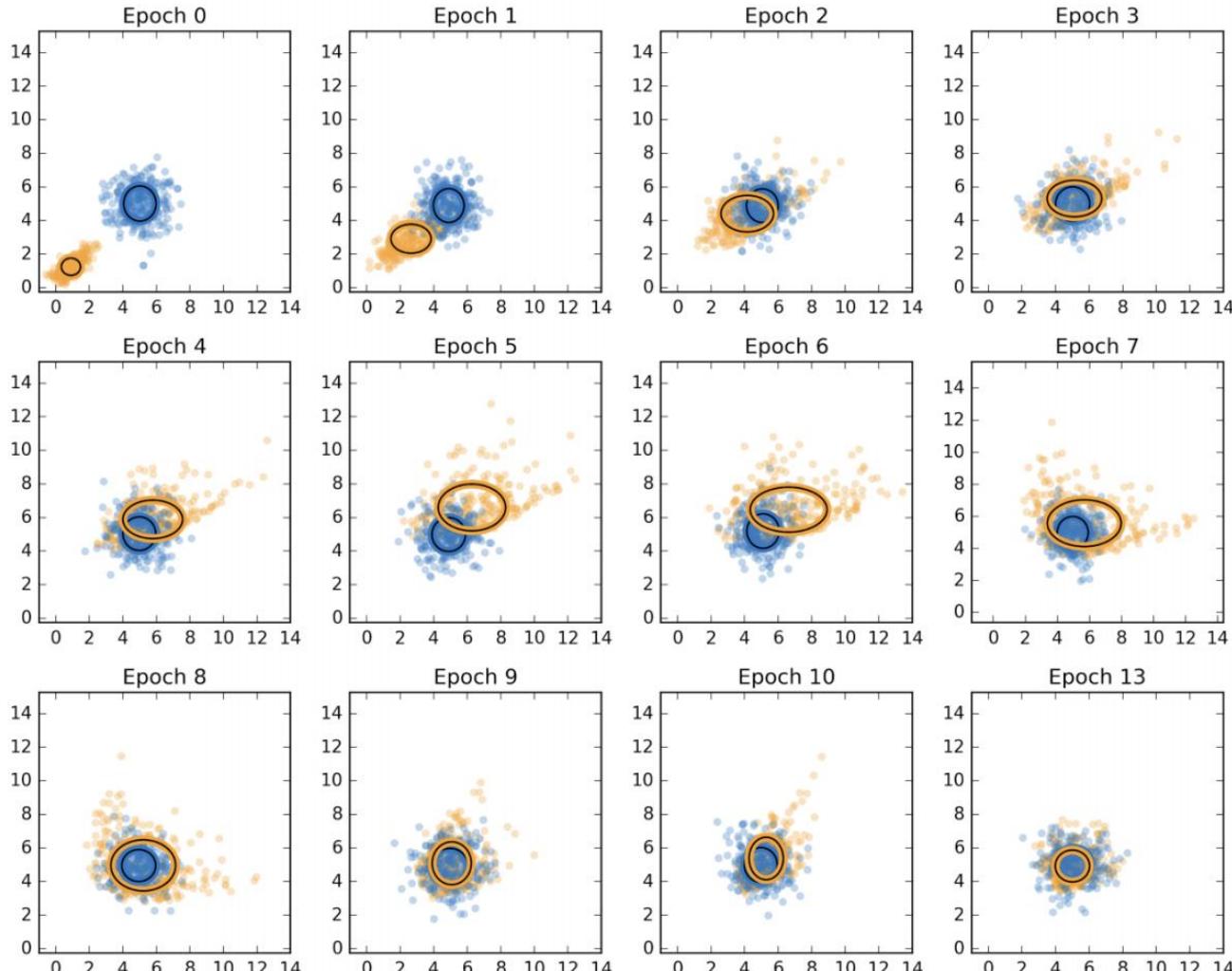
# Обучение генератора



# Идеальный сценарий обучения



# Распределения генератора (иdeal)



# План выступления

---

- Задачи дискриминации и генерации
- Задача генерация с точки зрения машинного обучения
- Как измерять качество генерации (изображений)
- Генеративно-состязательные сети
- **Проблемы и решения**
- Задача оптимальной транспортировки и функции потерь
- Интересные идеи в GANах
- Вариационные кодировщики
- Интересные идеи в VAE
- Немного примеров из жизни

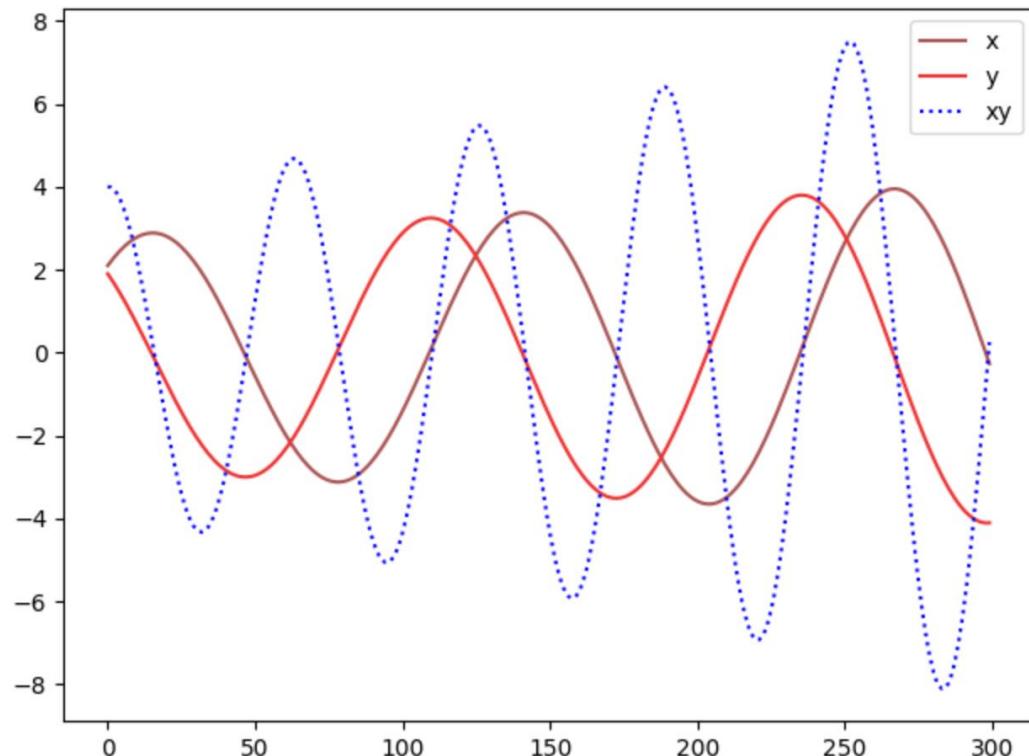
# Проблемы GANов

---

- Нет гарантий сходимости
- Осцилляция
- Слишком сильный дискриминатор (исчезающий градиент)
- Схлопывание мод распределения (mode collapse)
- KL-дивергенция это не лучшая функция для оптимизации

# Гарантии сходимости

- Пусть игрок А контролирует значение  $x$  и хочет максимизировать  $xy$ , а игрок В контролирует значение  $y$  и хочет минимизировать  $xy$
- Градиентный спуск не гарантирует сходимости



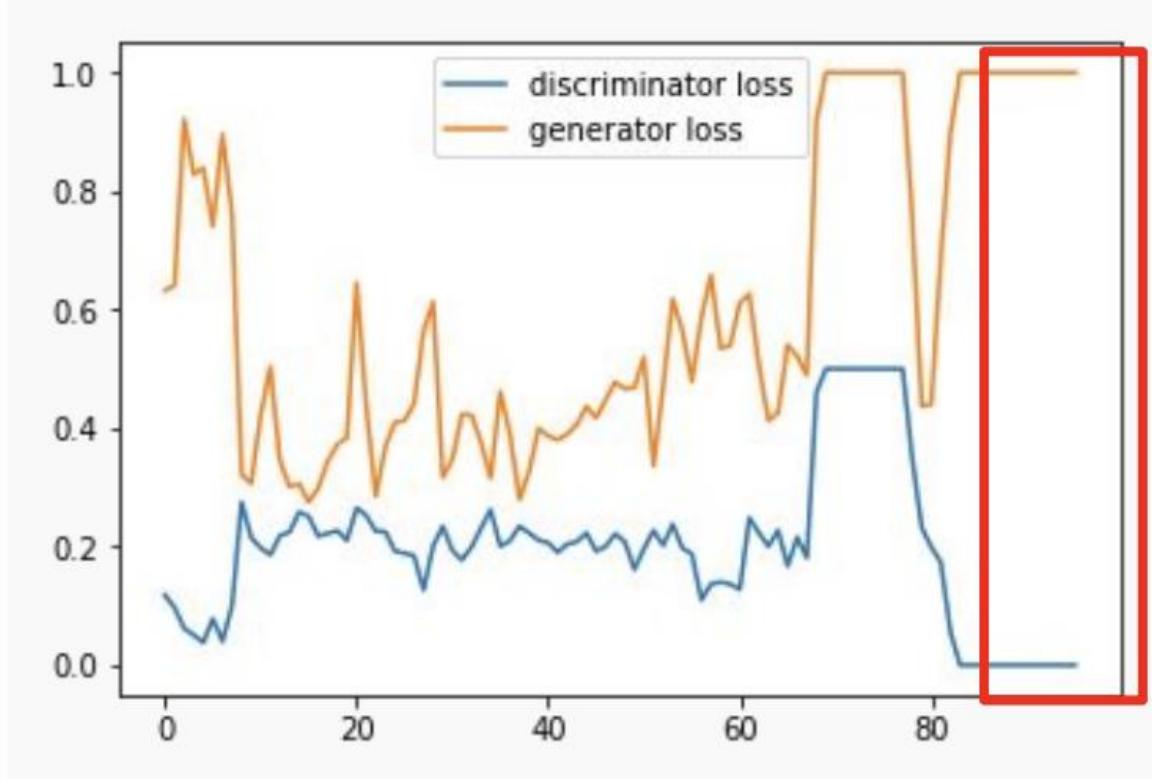
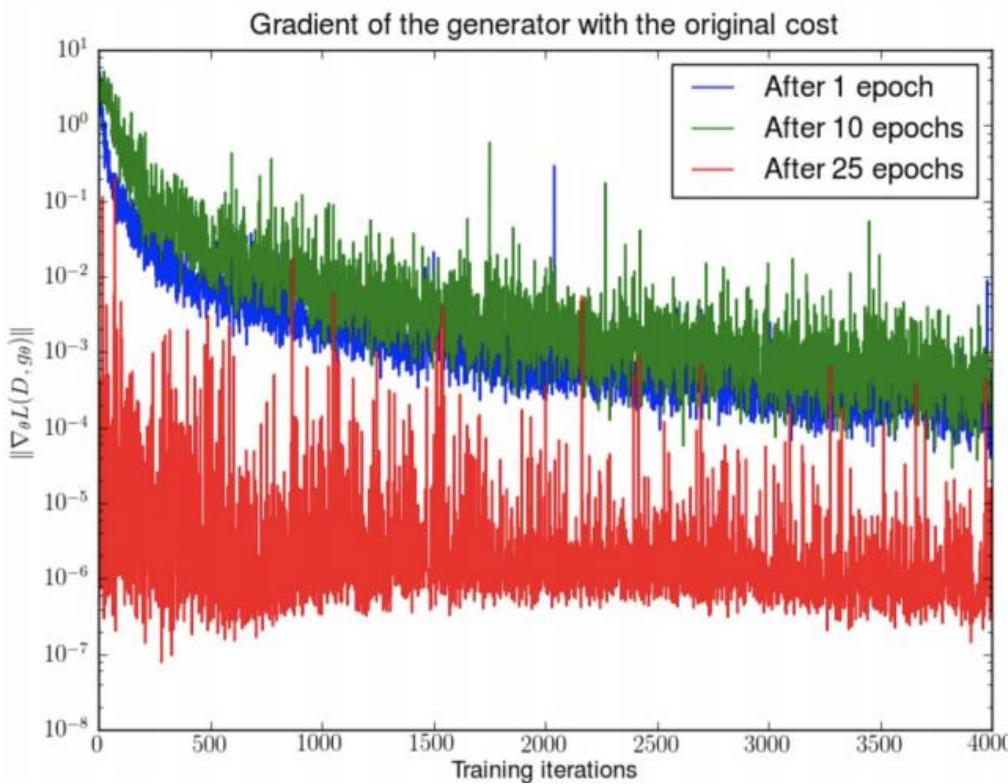
# Борьба с осцилляцией

---

- Подбор гиперпараметров
- Уход от постановки минимаксной игры
- Сопоставление карт признаков (feature matching)

# Затухание градиента

- Слишком сильный дискриминатор тормозит обучение генератора



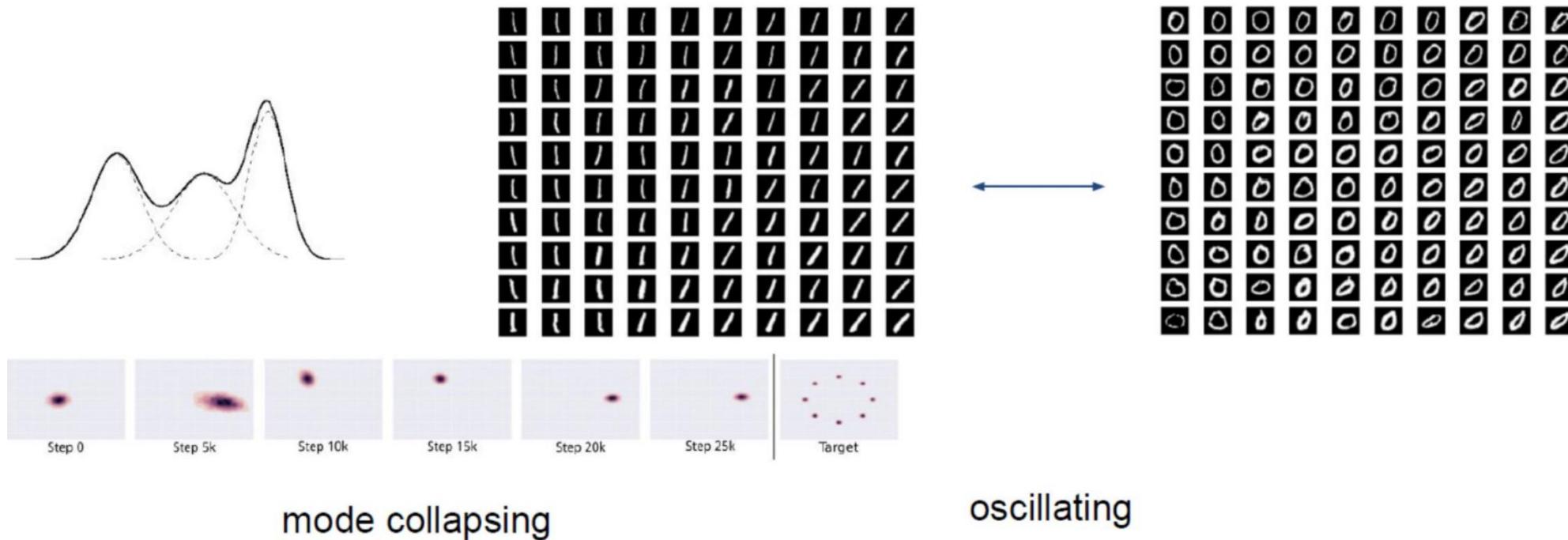
# Борьба с затуханием градиента

---

- Добавление шума дискриминатору
- Торможение обучения дискриминатора

# Mode collapse

- Генератор воспроизводит только некоторые из мод распределения



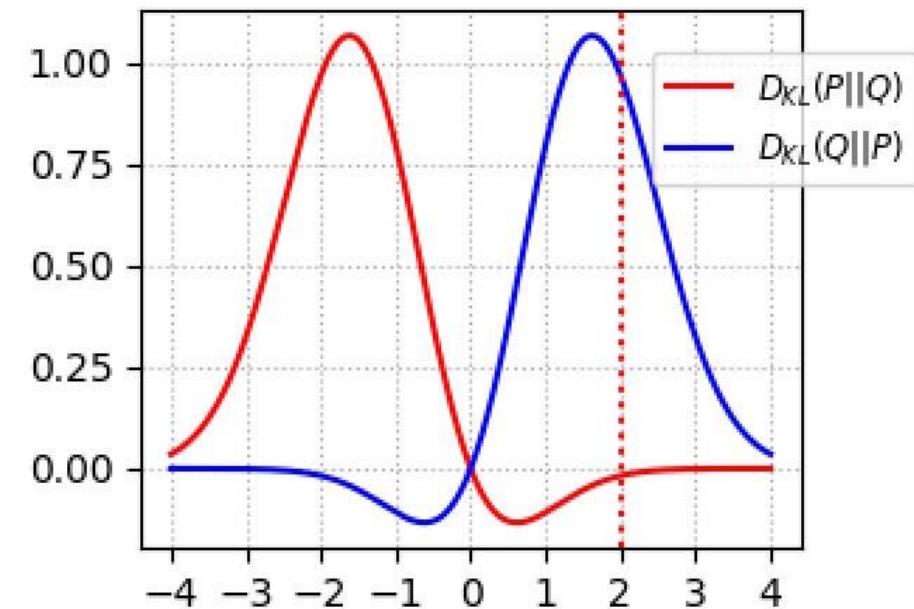
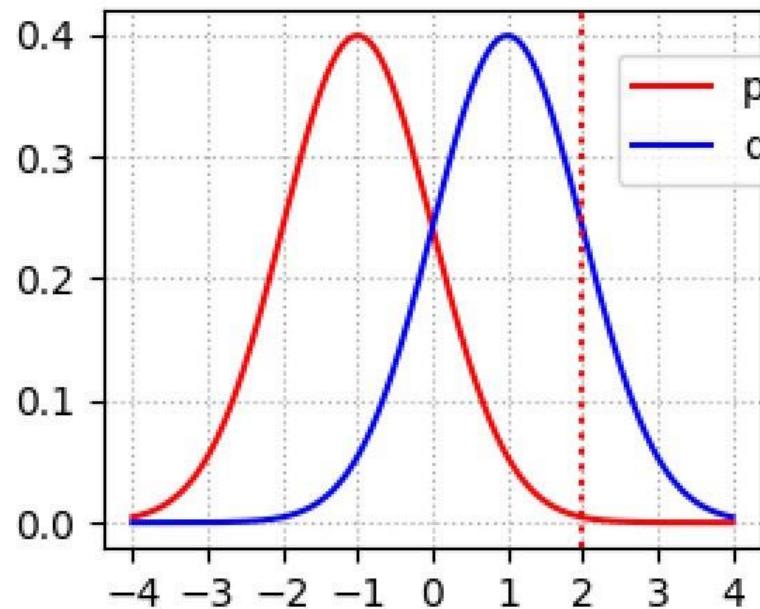
# Борьба с mode collapse

---

- Множественные генераторы (AdaGAN, MAD-GAN)
- Сопоставление карт признаков
- Минибатчевая дискrimинация
- CGANы

# Что не так с KL-дивергенцией

- Иногда бывает равна 0
- Не совсем точно отображает наши ожидания



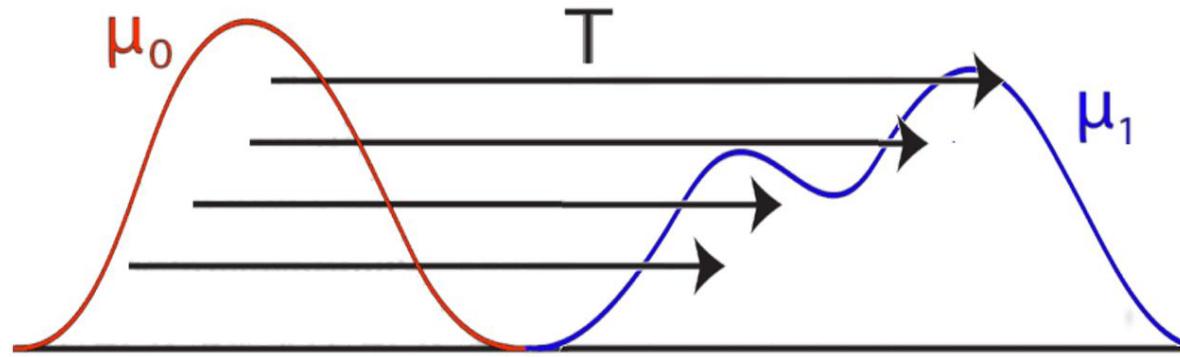
# План выступления

---

- Задачи дискриминации и генерации
- Задача генерация с точки зрения машинного обучения
- Как измерять качество генерации (изображений)
- Генеративно-состязательные сети
- Проблемы и решения
- **Задача оптимальной транспортировки и функции потерь**
- Интересные идеи в GANах
- Вариационные кодировщики
- Интересные идеи в VAE
- Немного примеров из жизни

# Задача оптимального перемещения

- Известна как задача Монжа—Канторовича
- Задача Монжа: переместить оптимальным способом одну кучу в земли в другую

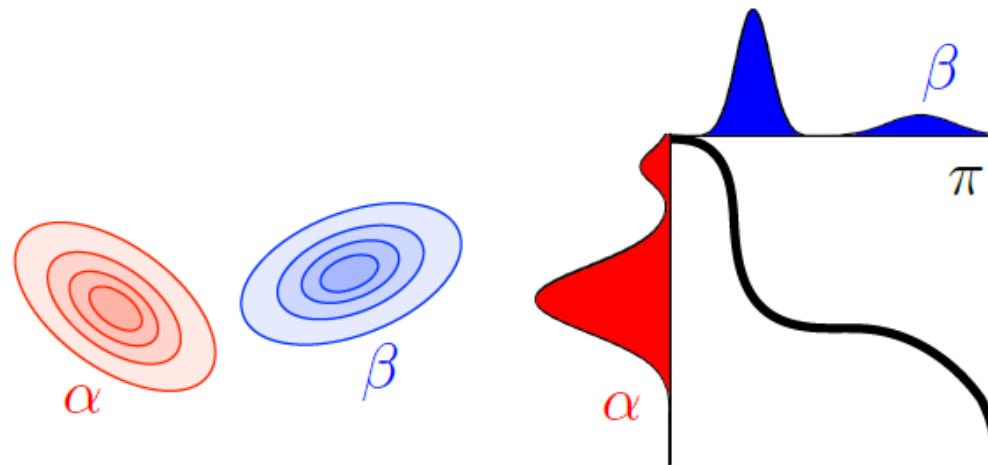


- В непрерывном случае:

$$\min_T \left\{ \int c(x, T(x)) d\alpha(x) : T_{\#}(\alpha) = \beta \right\}$$

# Задача Монжа-Канторовича

- Найти оптимальный план перемещения между двумя мерами



# Расстояние Вассерштейна

---

Расстояние Вассерштейна:

$$W(p_{\text{data}}, p_{\text{model}}) = \inf_{\gamma \in \Pi(p_{\text{data}}, p_{\text{model}})} \mathbb{E}_{(x,y) \sim \gamma} [| |x - y| |]$$

где  $\Pi(p_{\text{data}}, p_{\text{gen}})$  это множество совместных распределений над  $p_{\text{data}}$  и  $p_{\text{model}}$ .

Его нельзя посчитать напрямую, но можно найти приблизительное решение

Оно эквивалентно

$$W(p_{\text{data}}, p_{\text{gen}}) = \sup_{|f_L| < 1} (\mathbb{E}_{x \sim p_{\text{data}}} [f(x)] - \mathbb{E}_{x \sim p_{\text{model}}} [f(x)])$$

Супремум берется во всем функциям с константой Липшица, не большей 1.

# Wasserstein loss

---

- Будем пользоваться этим функционалом для обучения

	Discriminator/Critic	Generator
GAN	$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [\log D(\mathbf{x}^{(i)}) + \log (1 - D(G(\mathbf{z}^{(i)})))]$	$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (D(G(\mathbf{z}^{(i)})))$
WGAN	$\nabla_w \frac{1}{m} \sum_{i=1}^m [f(x^{(i)}) - f(G(z^{(i)}))]$	$\nabla_{\theta} \frac{1}{m} \sum_{i=1}^m f(G(z^{(i)}))$

- Для поддержания константы обрежем градиенты (clipping)

# WGAN-GP

---

- Вместо обрезания градиента, добавим регуляризацию, зависящую от градиента:

$$L = \underbrace{\mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)]}_{\text{Original critic loss}} + \underbrace{\lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]}_{\text{Our gradient penalty}}.$$

# LSGAN

---

- Вместо логарифмов будем оптимизировать квадратичную функцию потерь:

## 1. Функция потерь для дискриминатора

$$\min_{\theta_d} \left[ \mathbb{E}_{x \sim p_{\text{data}}} (D_{\theta_d}(x) - b)^2 + \mathbb{E}_{z \sim p(z)} (D_{\theta_d}(G_{\theta_g}(z)) - a)^2 \right]$$

## 2. Функция потерь для генератора:

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} (D_{\theta_d}(G_{\theta_g}(z)) - c)^2$$

Обычно

- либо  $a = -1, b = 1, c = 0$
- либо  $a = 0, b = 1, c = 1$

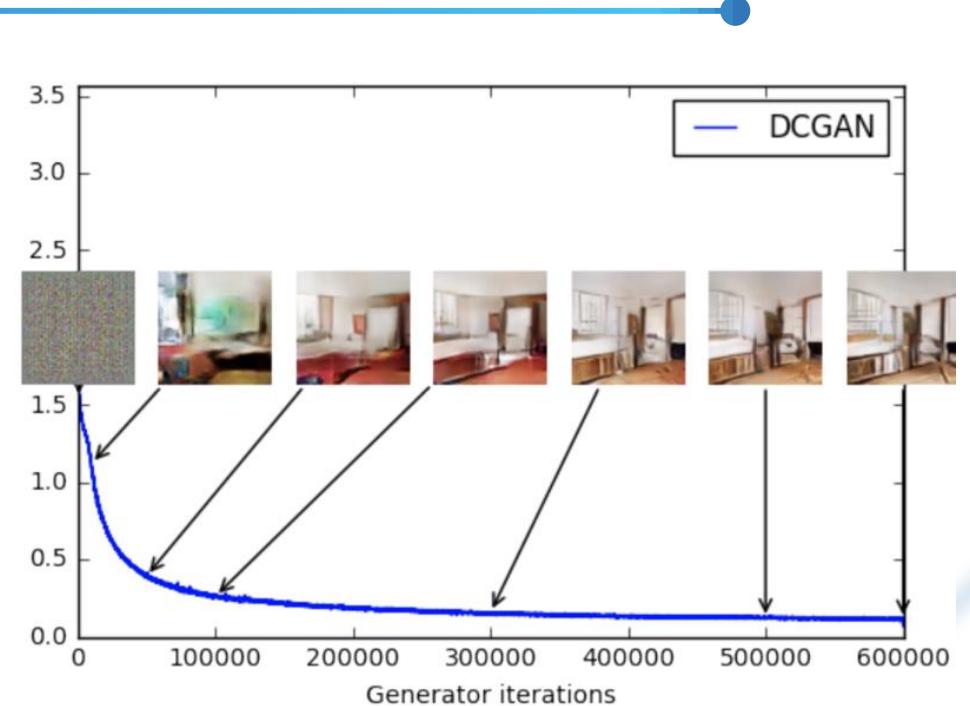
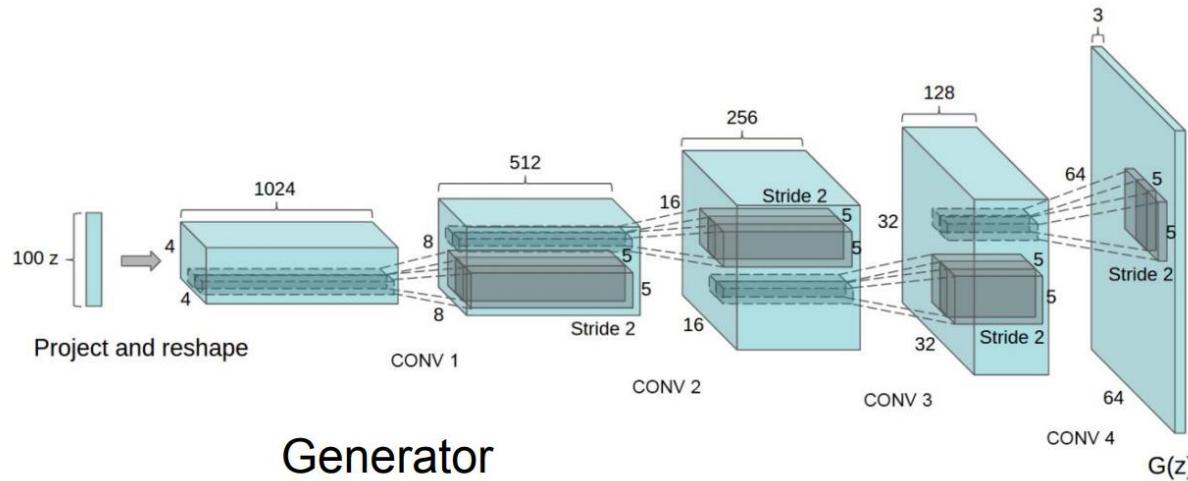
# План выступления

---

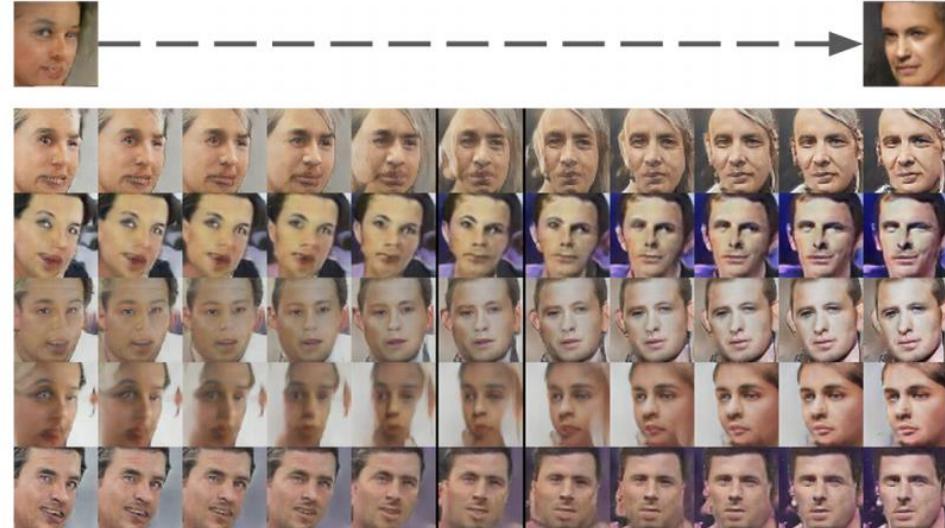
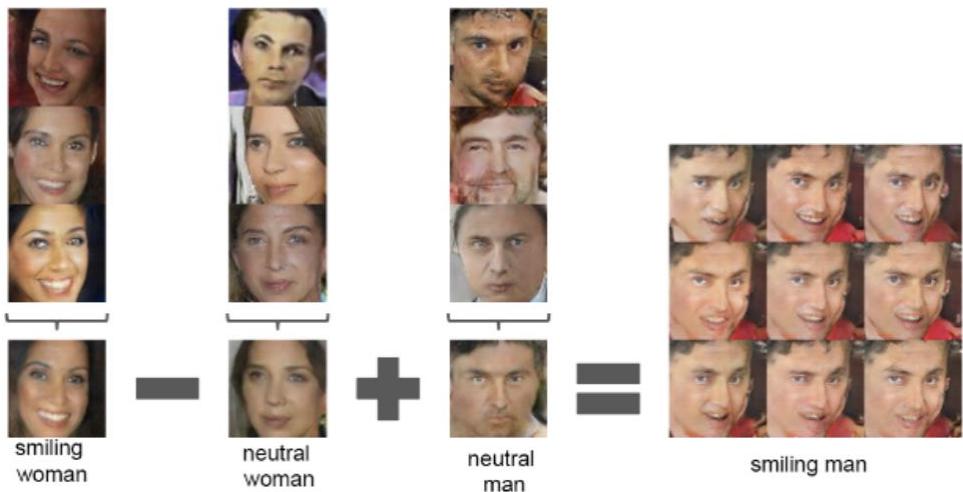
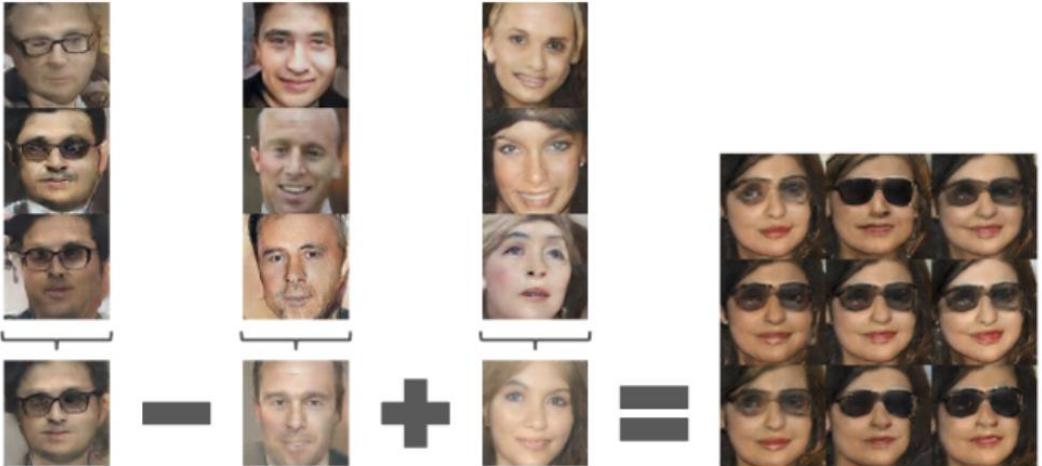
- Задачи дискриминации и генерации
- Задача генерация с точки зрения машинного обучения
- Как измерять качество генерации (изображений)
- Генеративно-состязательные сети
- Проблемы и решения
- Задача оптимальной транспортировки и функции потерь
- **Интересные идеи в GANах**
- Вариационные кодировщики
- Интересные идеи в VAE
- Немного примеров из жизни

# DCGAN

- Нет пулинга (вместо него свертки)
- Батчнормализация в генераторе и дискриминаторе
- Нет полносвязных скрытых слов
- ReLU в генераторе на всех слоях и tanh на выходе
- LeakyReLU в дискриминаторе на всех слоях



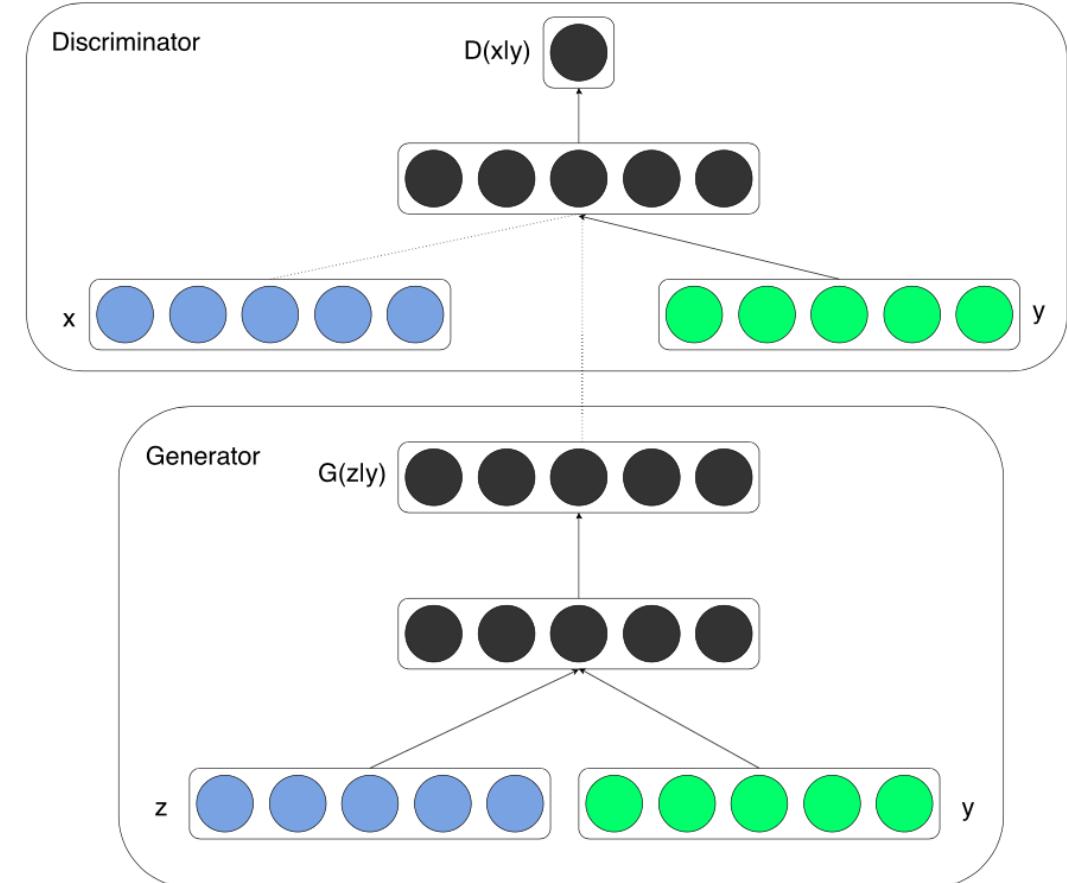
# Семантическая арифметика и интерполяирование



# CGAN

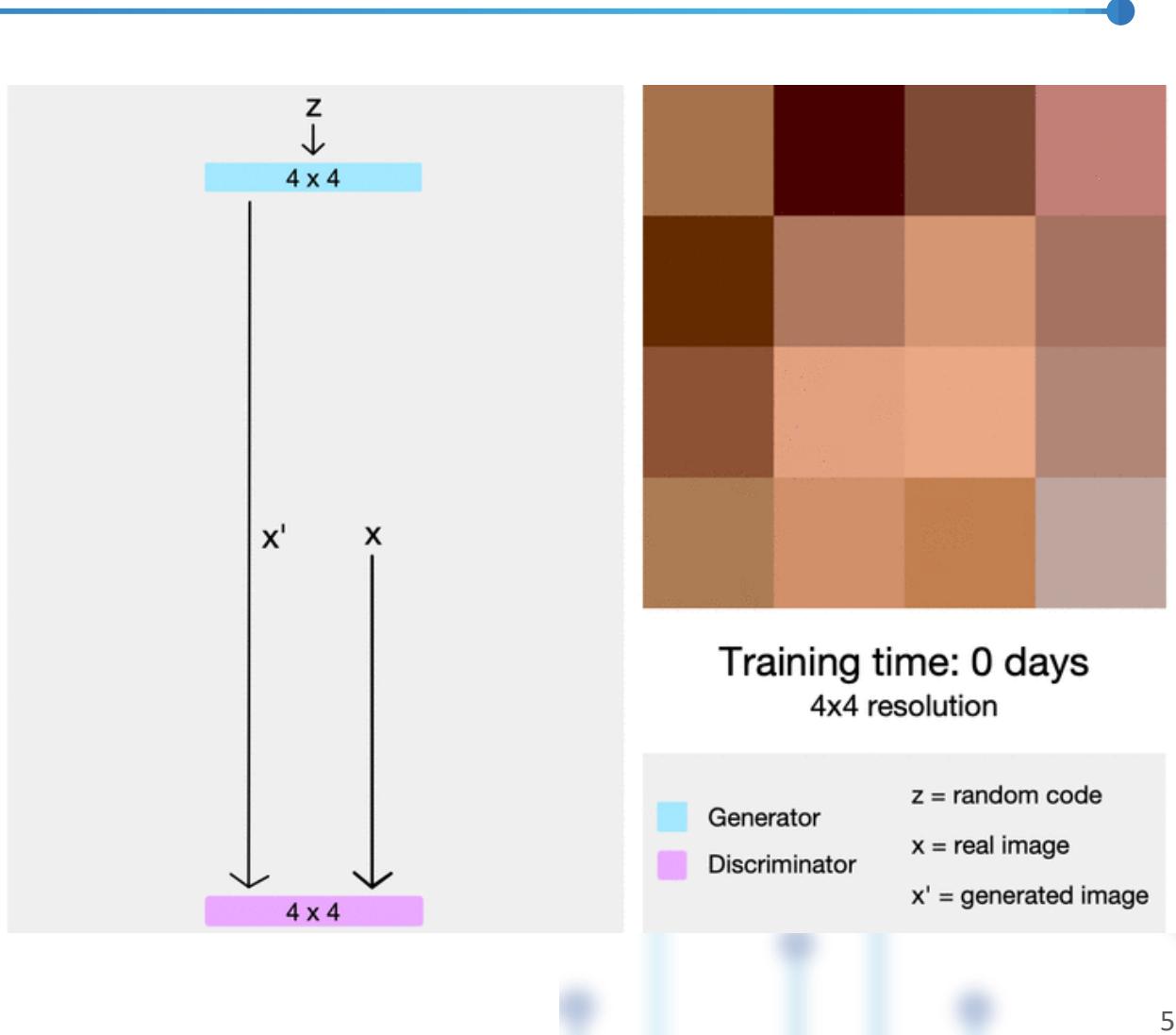
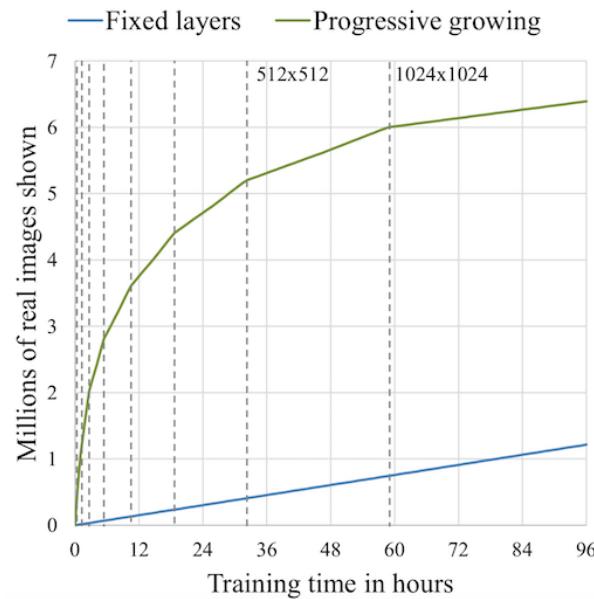
- Добавим вектор контекста
- Оптимизируемая функция

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



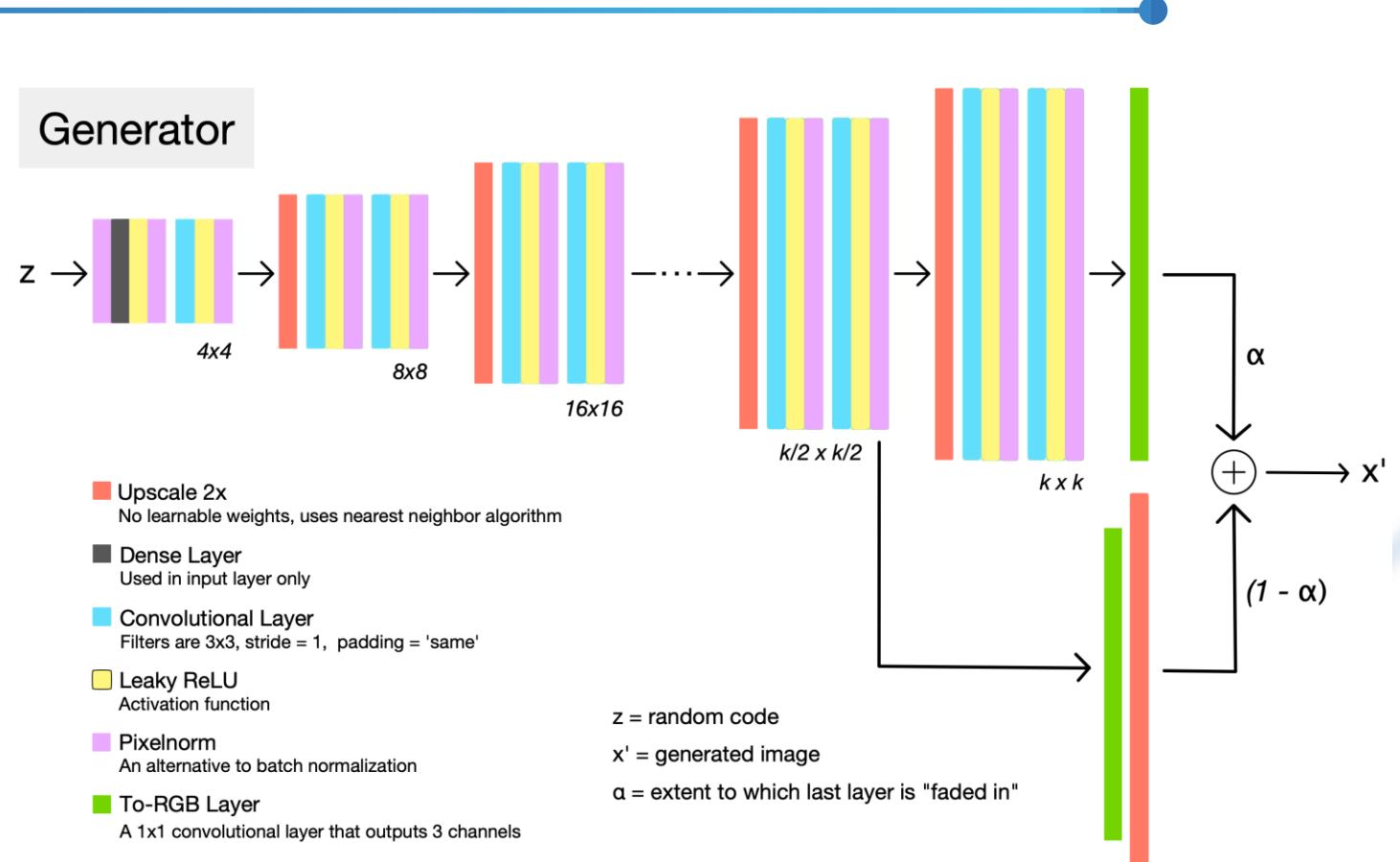
# ProGAN

- Последовательное добавление слоев большего размера после полного обучения предыдущих слоев



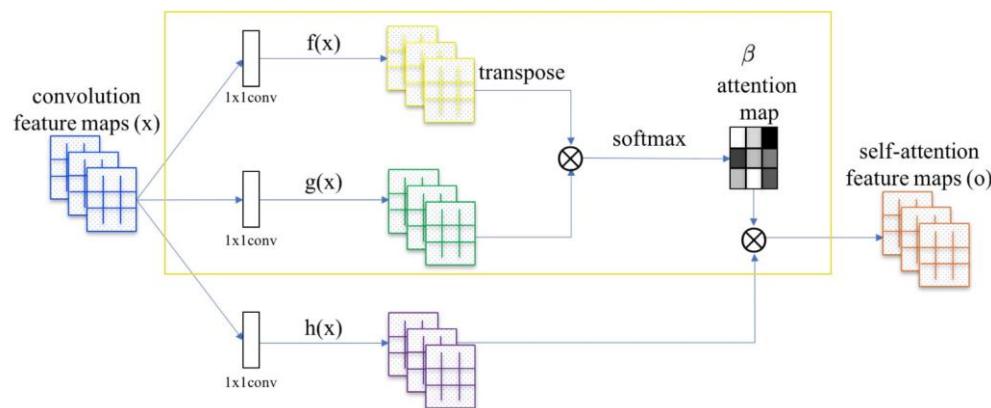
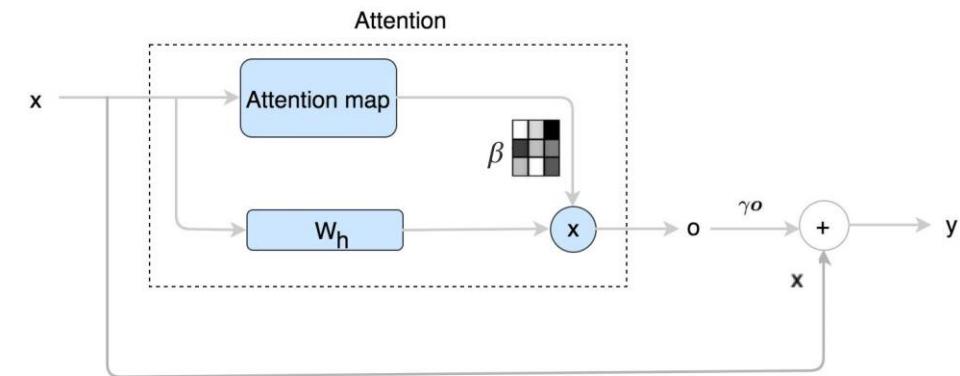
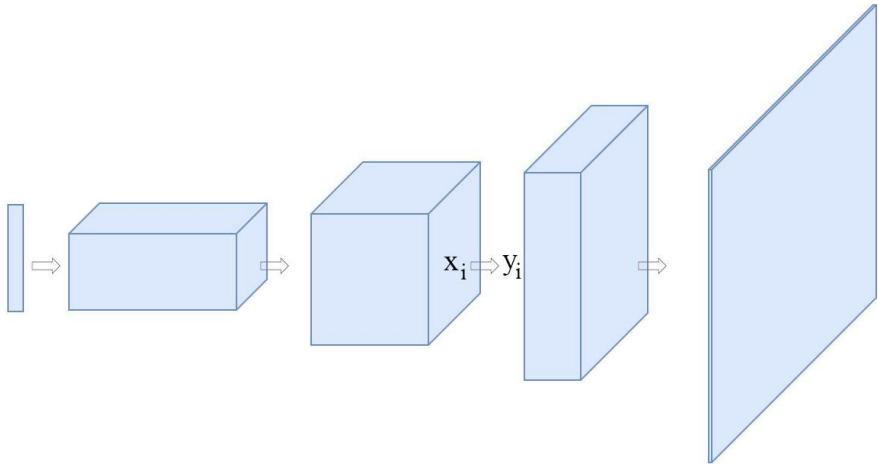
# Архитектура и обучение ProGAN

- Новые слои добавляются одновременно в генератор и дискриминатор
- Чтобы смягчить добавление, сигнал с нового слоя идет с постепенно увеличивающейся со временем константой  $\alpha$
- Вместо батчевой нормализации пиксельная нормализация

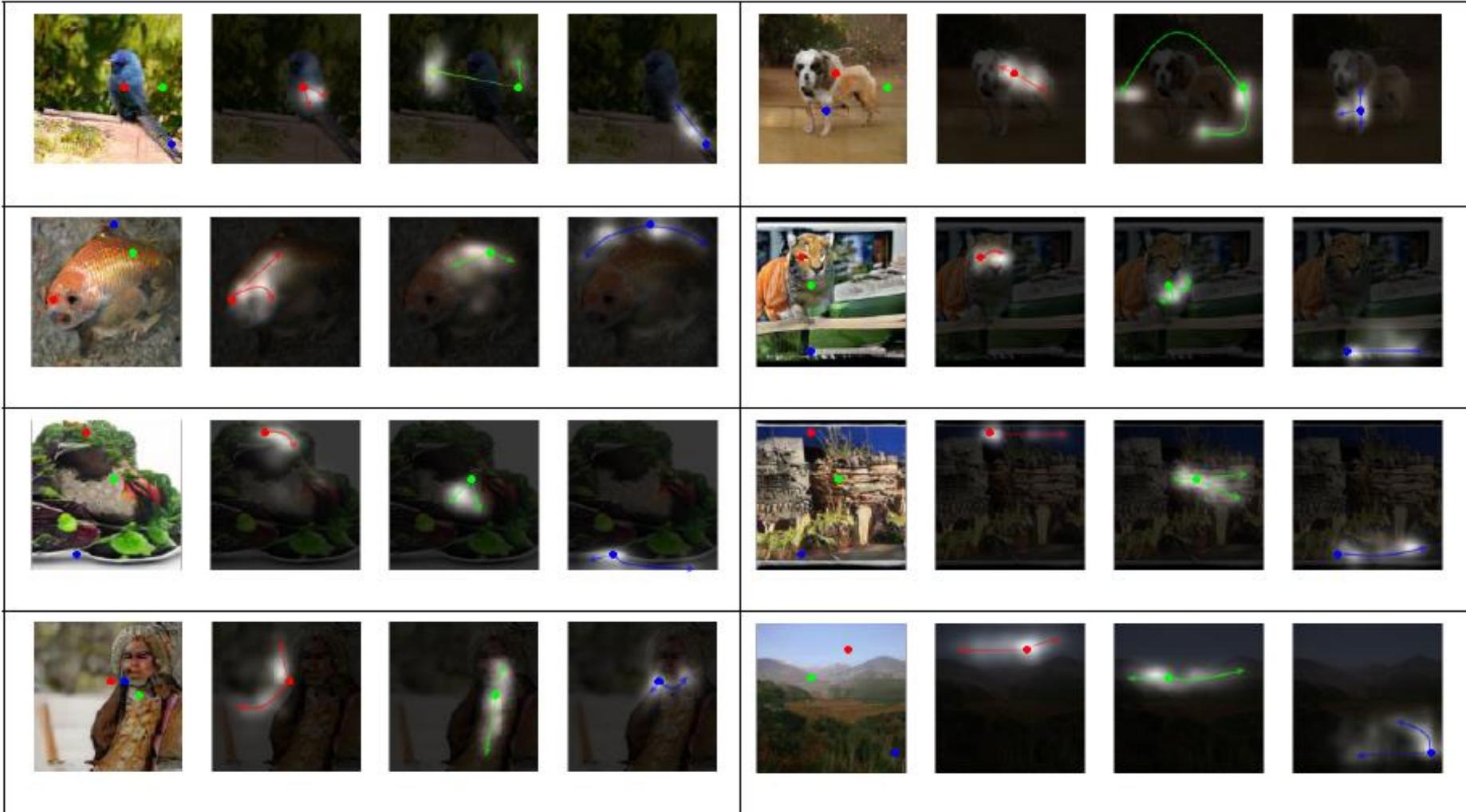


# SAGAN

- Добавим self-attention к каждому сверточному слою генератора



# Работа SAGAN



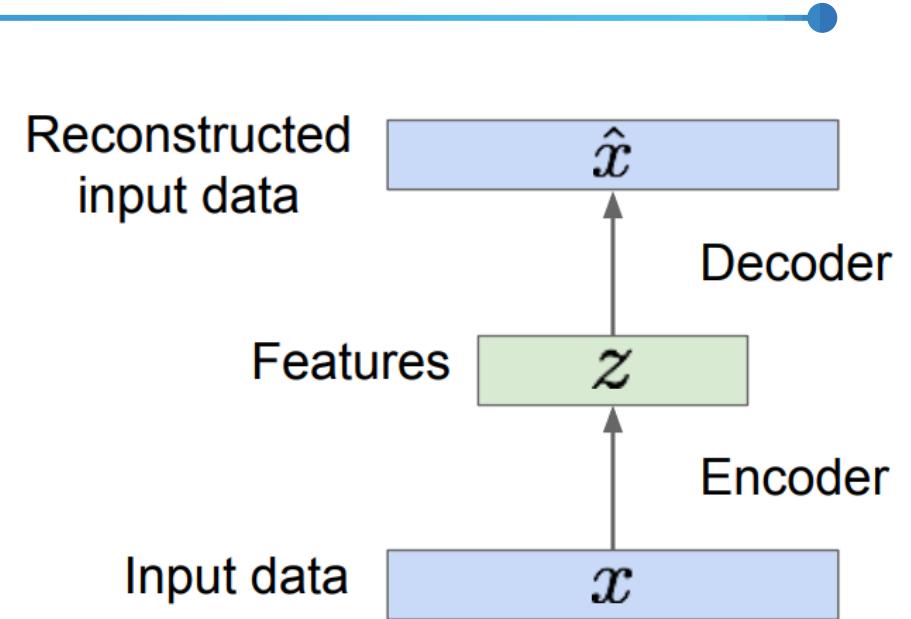
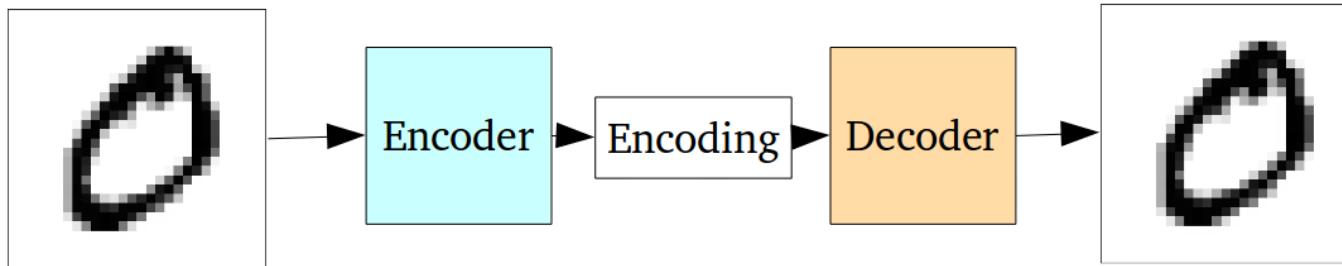
# План выступления

---

- Задачи дискриминации и генерации
- Задача генерация с точки зрения машинного обучения
- Как измерять качество генерации (изображений)
- Генеративно-состязательные сети
- Проблемы и решения
- Задача оптимальной транспортировки и функции потерь
- Интересные идеи в GANах
- **Вариационные кодировщики**
- Интересные идеи в VAE
- Немного примеров из жизни

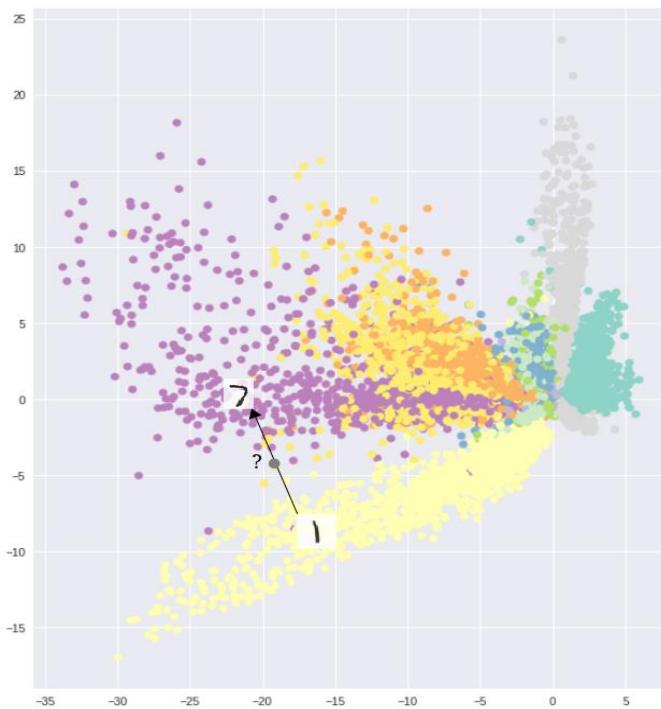
# Обычные автоэнкодеры

- Обучение без учителя (иногда говорят про самообучение), при котором обучается низкоразмерное представление объекта



# Главный недостаток для генерации

- Пространство скрытых представление ведет себя неизвестно как:
  - может быть дискретно
  - может быть не непрерывно



# Исходное вероятностное распределение

---

Зададим вид распределения на данных, зависящий от некоторой скрытой переменной  $z$ :

$$p_{\theta}(x) = \int p_{\theta}(x|z)p_{\theta}(z)dz$$

# Основная идея

---

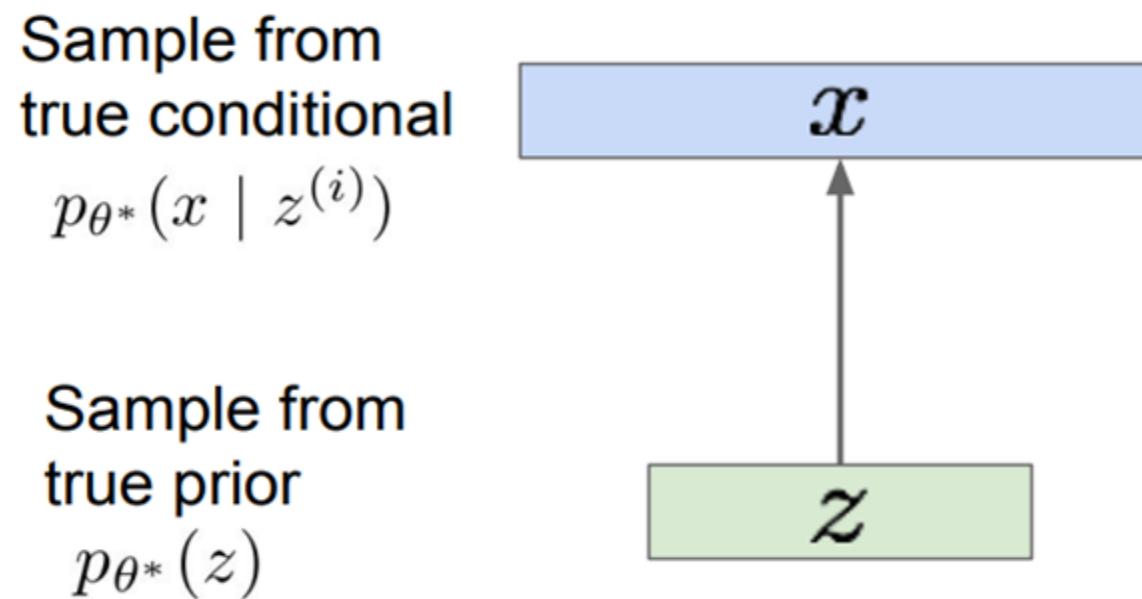
Вместо того, чтобы использовать некоторые предположения о том, как должна выглядеть структура вероятностной модели, мы определяем неразрешимую функцию плотности с некоторой скрытой переменной  $z$ :

$$p_{\theta}(x) = \int p_{\theta}(x|z)p_{\theta}(z)dz$$

# Скрытые переменные

---

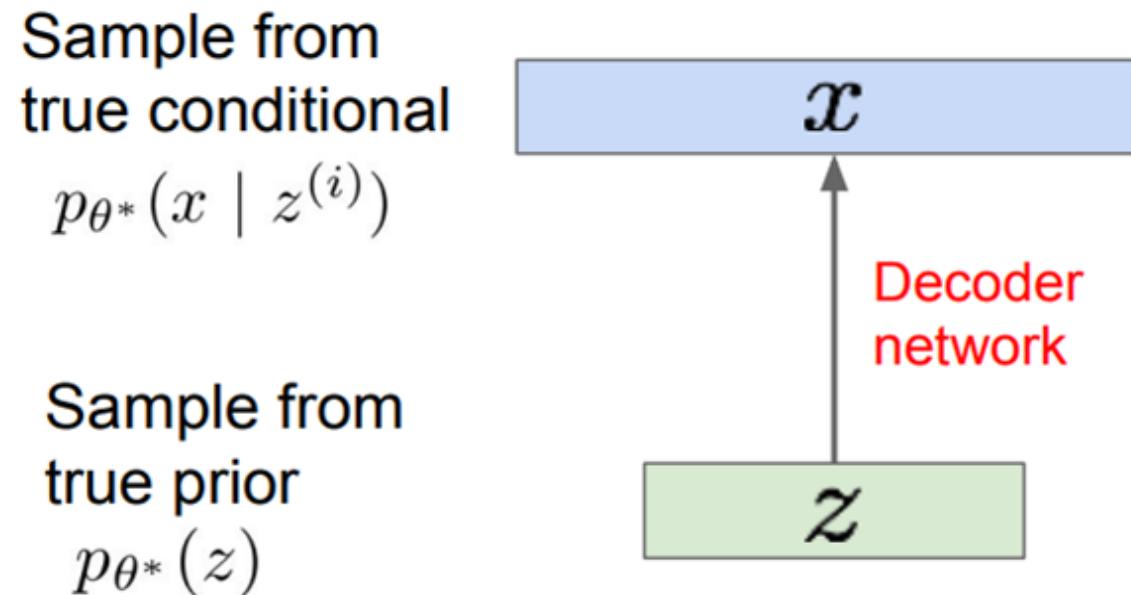
Предположим, что данные обучения генерируются в зависимости от некоторого латентного  $z$ , устроенного достаточно простоим образом



# Представление $p(x|z)$

---

Условное распределение  $p(x|z)$  является сложным, восстановим его при помощи нейронной сети



# Параметрическое обучение

---

$$p_{\theta}(x) = \int p_{\theta}(x|z)p_{\theta}(z)dz$$

$$\theta^* = \arg \min_{\theta} -\log \int p_{\theta}(x|z)p_{\theta}(z)dz$$

В чем здесь проблема?

# Параметрическое обучение

---

$$p_{\theta}(x) = \int p_{\theta}(x|z)p_{\theta}(z)dz$$

$$\theta^* = \arg \min_{\theta} -\log \int p_{\theta}(x|z)p_{\theta}(z)dz$$

$\int p_{\theta}(x|z)p_{\theta}(z)dz$  трудноразрешимо!

# Параметры обучения

---

$$p_{\theta}(x) = \int p_{\theta}(x|z)p_{\theta}(z)dz$$

$$\theta^* = \arg \max_{\theta} \log \int p_{\theta}(x|z)p_{\theta}(z)dz$$

$\int p_{\theta}(x|z)p_{\theta}(z)dz$  трудноразрешим!

Вероятность апостериорных данных также нельзя выразить!

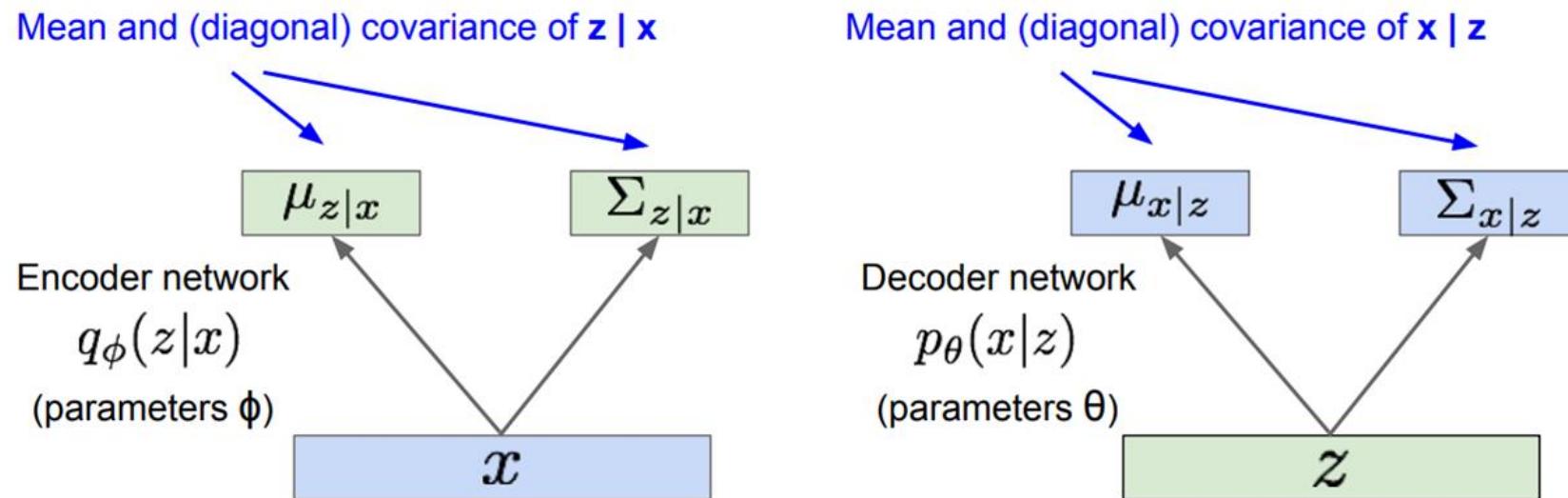
# Кодирующая сеть

---

Идея: добавить сеть кодировщика  $q_\phi(z|x)$ ,  
аппроксимирующую  $p_\phi(z|x)$

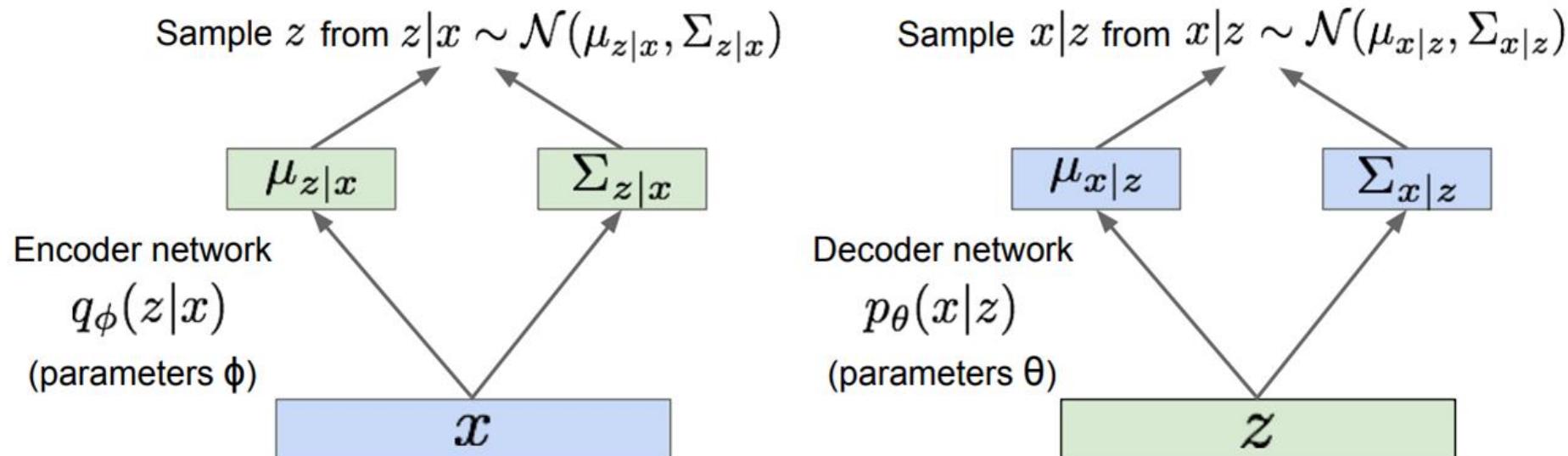
# Кодировщик и декодировщик

Кодировщик и декодировщик являются вероятностными, оба предполагают, что гиперпараметры распределены по Гауссу



# Сэмплирование с кодировщиком и декодировщиком

Мы можем выбрать  $z$  с помощью кодировщика и  $x|z$  с помощью декодировщика



# Возвращаясь к вероятности

---

$$\begin{aligned}\theta^* &= \arg \max_{\theta} \log p_{\theta}(x) \\ \log p_{\theta}(x^{(i)}) &= \mathbb{E}_{z \sim q_{\phi}(z|x^{(i)})} [\log p_{\theta}(x^{(i)})] = \\ &= \mathbb{E}_z \left[ \log \frac{p_{\theta}(x^{(i)}|z)p_{\theta}(z)}{p_{\theta}(z|x^{(i)})} \right] = \\ &= \mathbb{E}_z \left[ \log \frac{p_{\theta}(x^{(i)}|z)p_{\theta}(z)}{p_{\theta}(z|x^{(i)})} \frac{q_{\phi}(z|x^{(i)})}{q_{\phi}(z|x^{(i)})} \right] = \\ &= \mathbb{E}_z [\log p_{\theta}(x^{(i)}|z)] - \mathbb{E}_z \left[ \log \frac{q_{\phi}(z|x^{(i)})}{p_{\theta}(z)} \right] + \mathbb{E}_z \left[ \log \frac{q_{\phi}(z|x^{(i)})}{p_{\theta}(z|x^{(i)})} \right] = \\ &= \mathbb{E}_z [\log p_{\theta}(x^{(i)}|z)] - D_{KL} (q_{\phi}(z|x^{(i)}) || p_{\theta}(z)) + D_{KL} (q_{\phi}(z|x^{(i)}) || p_{\theta}(z|x^{(i)}))\end{aligned}$$

# Нижние границы

$$E_z \left[ \log p_\theta(x^{(i)} | z) \right] - D_{KL} \left( q_\phi(z | x^{(i)}) || p_\theta(z) \right) + D_{KL} \left( q_\phi(z | x^{(i)}) || p_\theta(z | x^{(i)}) \right)$$

↑  
Decoder network gives  $p_\theta(x|z)$ , can compute estimate of this term through sampling.

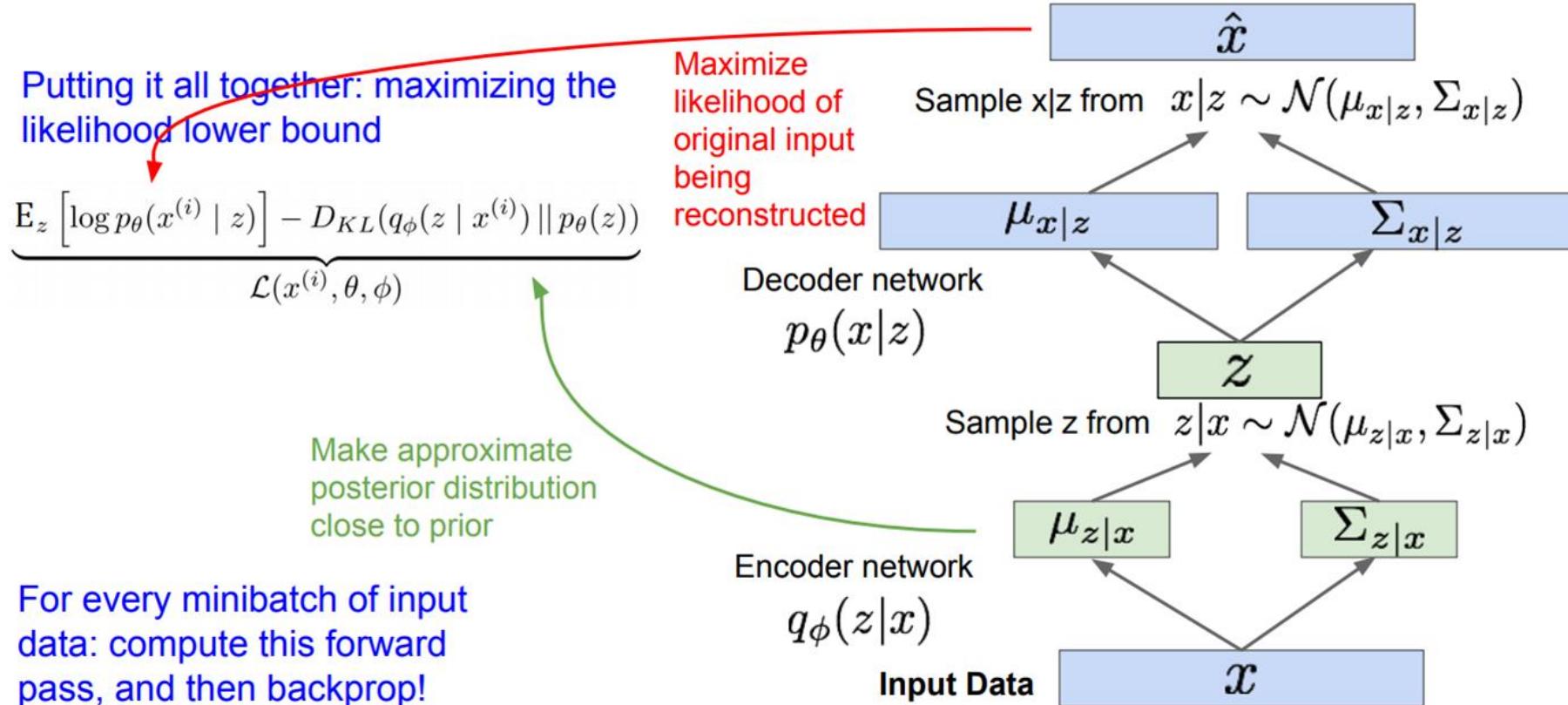
↑  
This KL term (between Gaussians for encoder and z prior) has nice closed-form solution!

↑  
 $p_\theta(z|x)$  intractable (saw earlier), can't compute this KL term :( But we know KL divergence always  $\geq 0$ .

$$\underbrace{E_z \left[ \log p_\theta(x^{(i)} | z) \right] - D_{KL}(q_\phi(z | x^{(i)}) || p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)} + \underbrace{D_{KL}(q_\phi(z | x^{(i)}) || p_\theta(z | x^{(i)}))}_{\geq 0} \geq 0$$

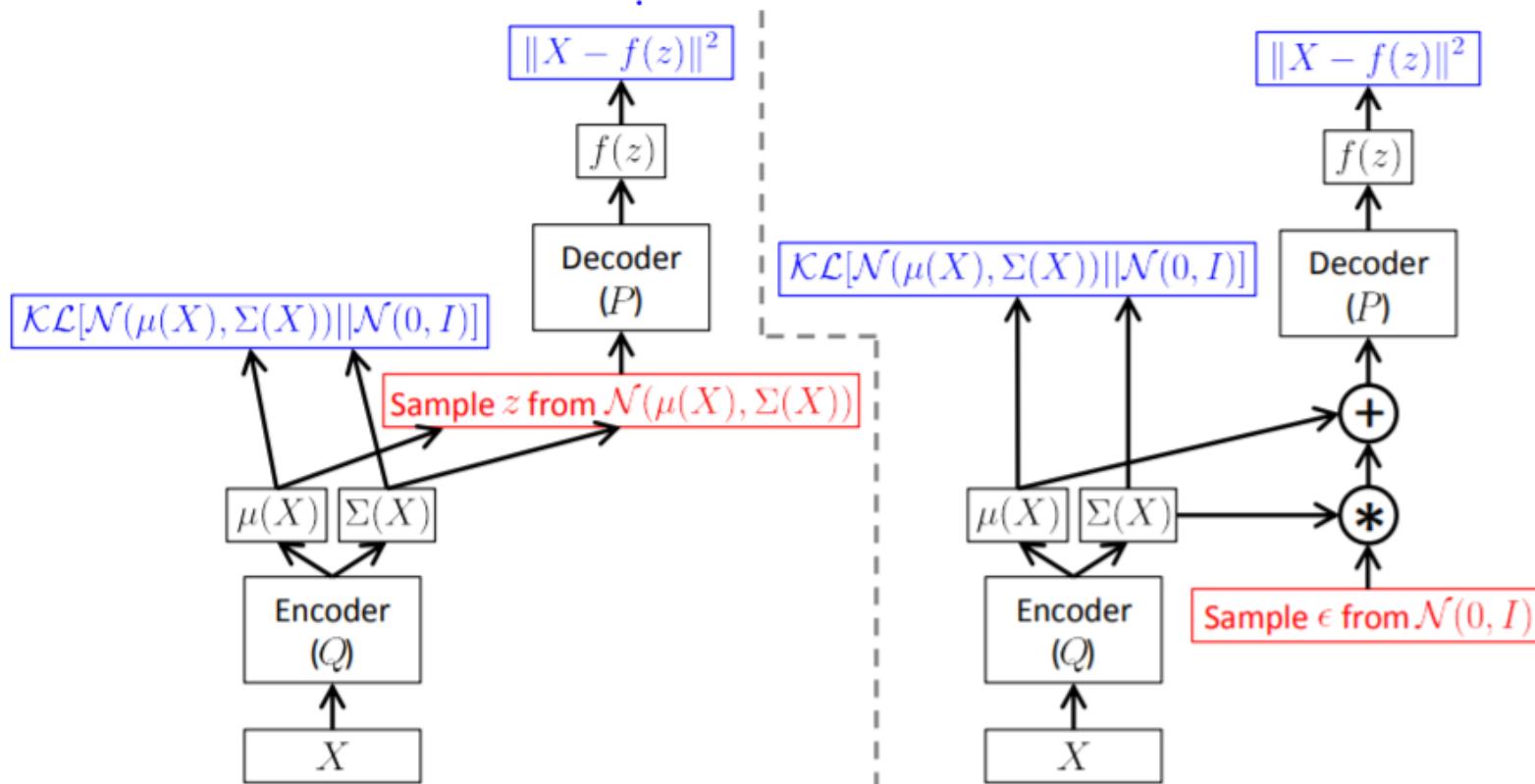
**Tractable lower bound** which we can take gradient of and optimize! ( $p_\theta(x|z)$  differentiable, KL term differentiable)

# Обучение VAE



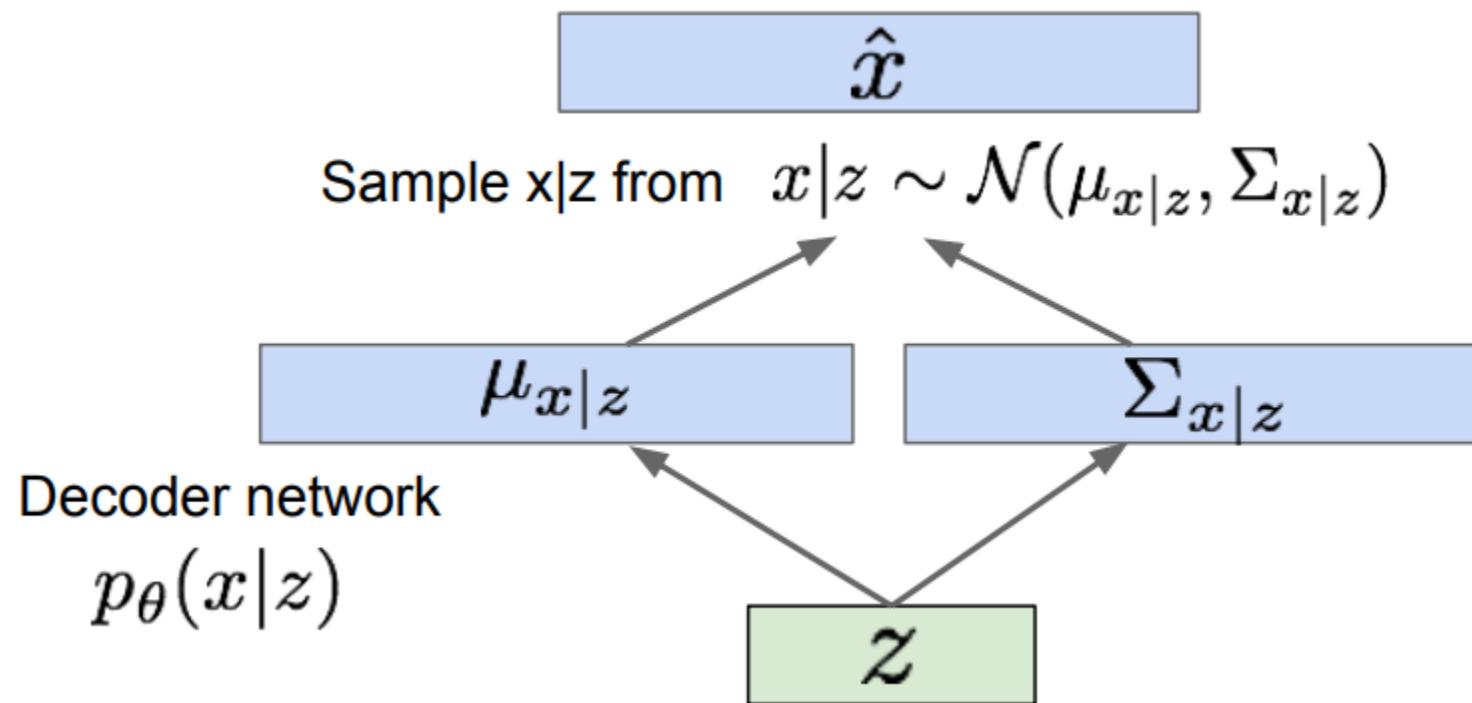
# Репараметризационный трюк

- Заменим переменные, добавив случайный коэффициент



# Создание данных с помощью VAE

Просто пример с декодером



# Анализ VAE

---

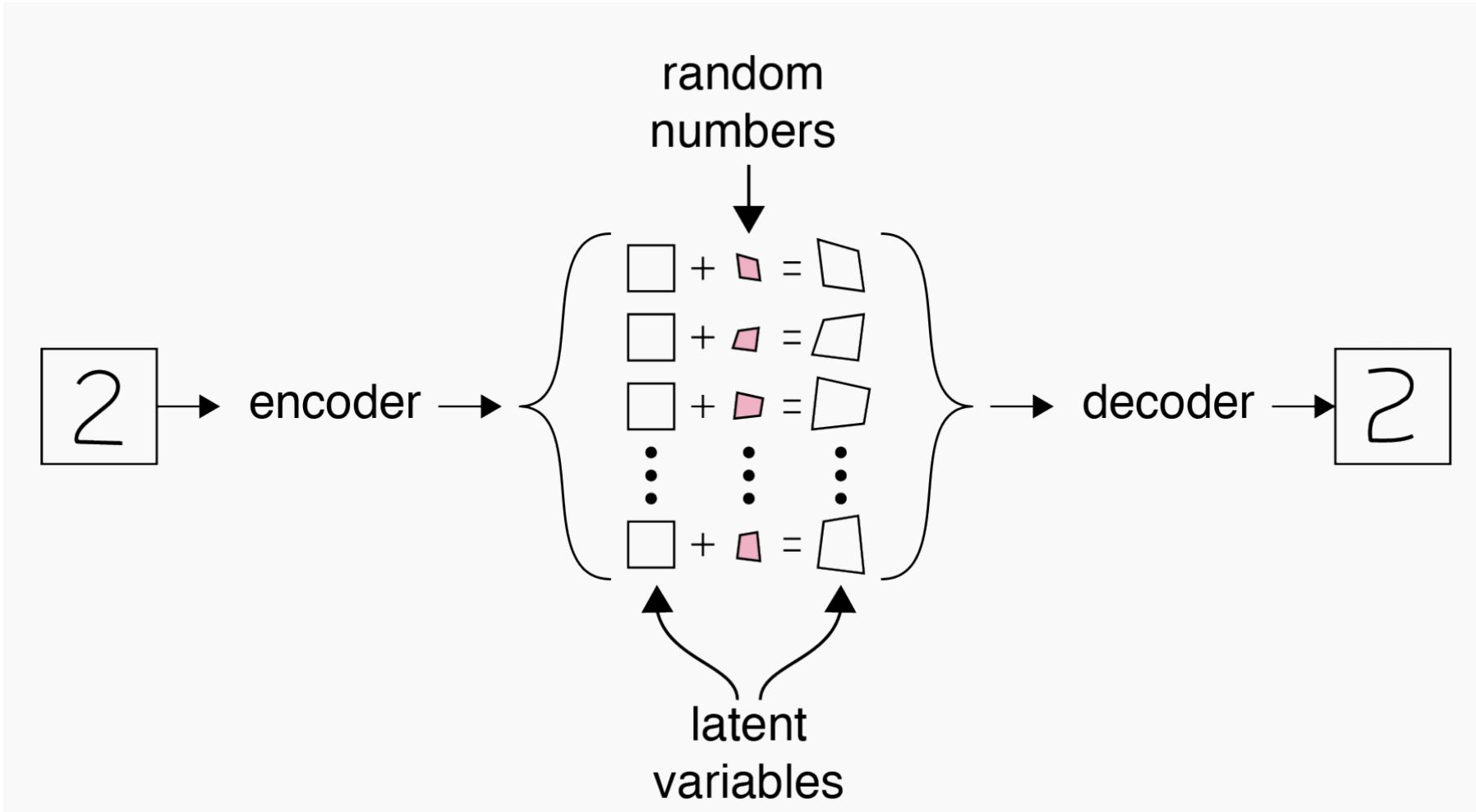
Преимущества:

- Принципиальный подход к генеративным моделям
- Позволяет сделать вывод  $q(z | x)$ , может быть полезным представлением функции для других задач

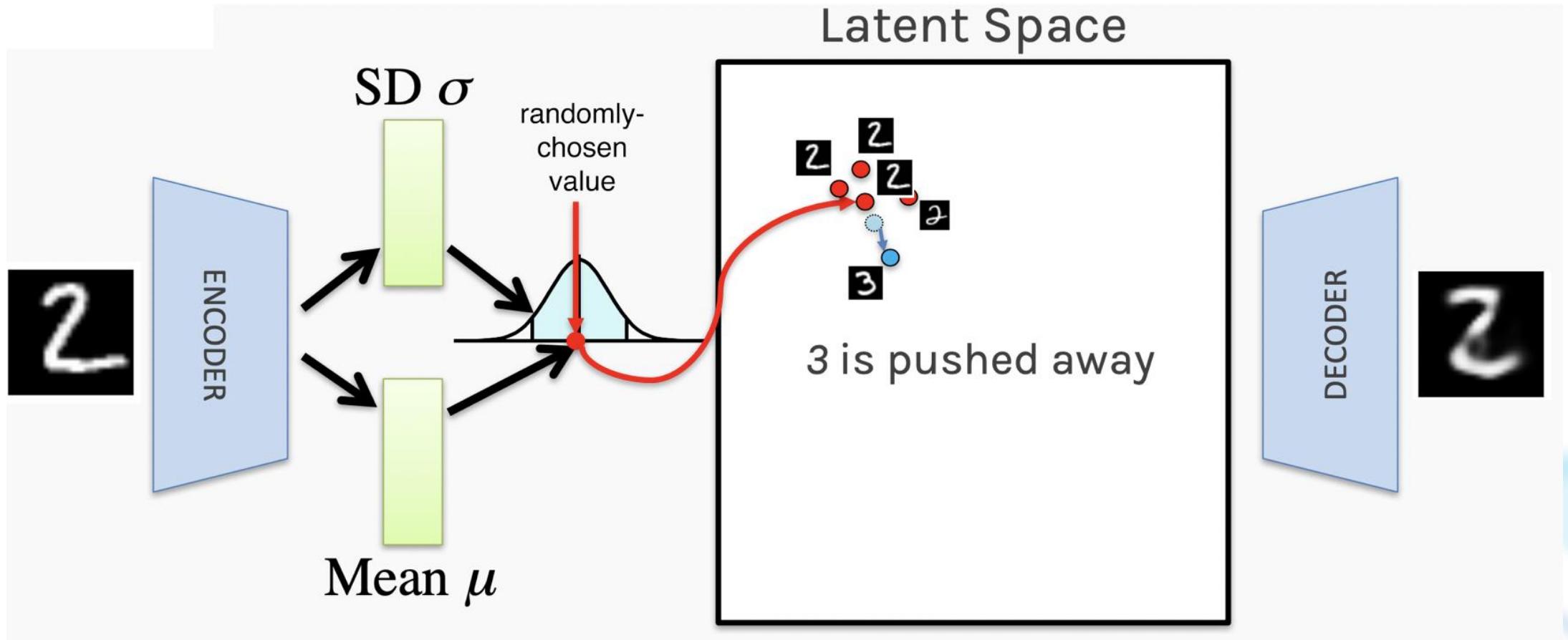
Недостатки:

- Максимизация нижней границы вероятности - не такая хорошая оценка, как PixelRNN / PixelCNN.
- Образцы более размытые и более низкого качества по сравнению с современными (GAN)

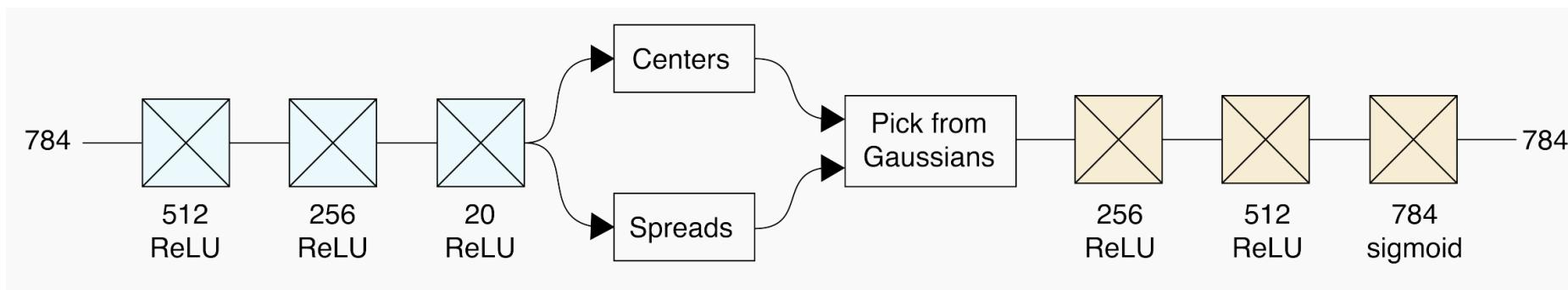
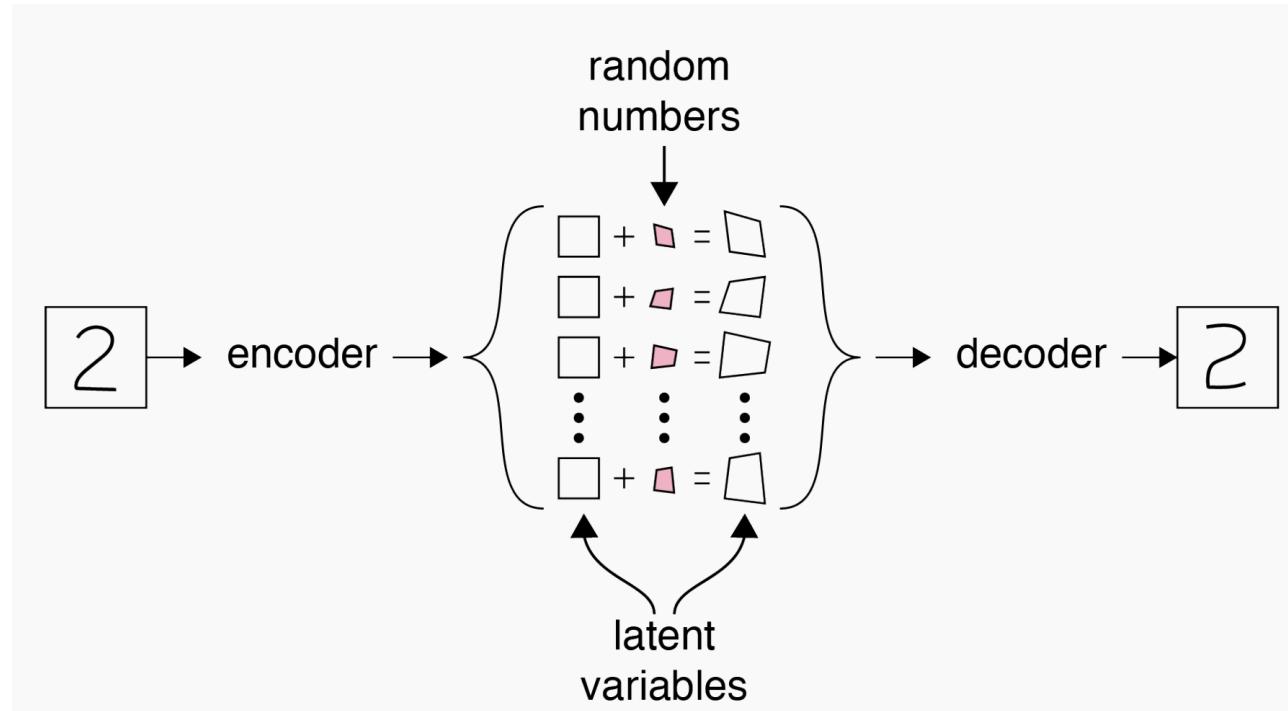
# Пример генерации



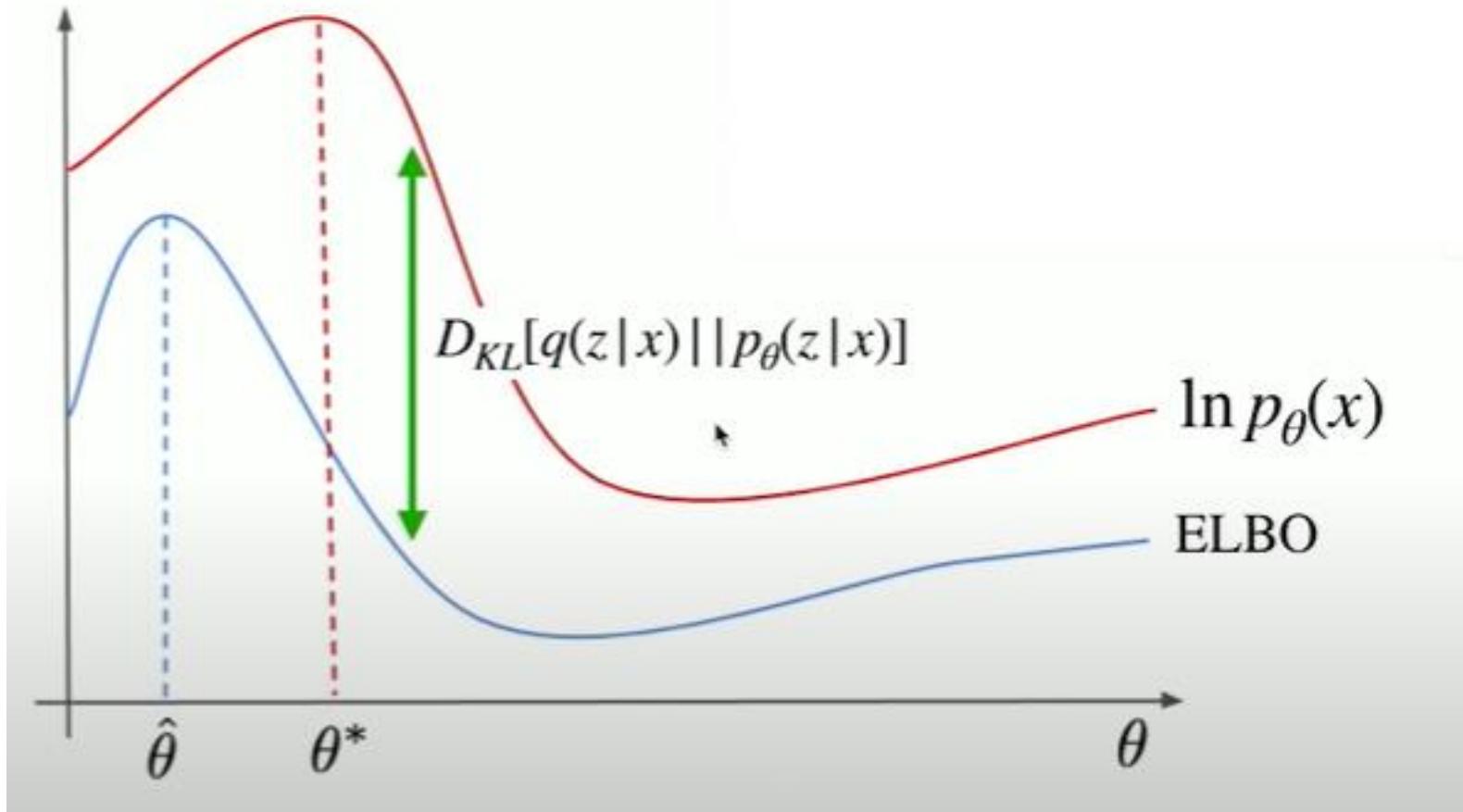
# Иллюстрация обучения



# Генерация чисел



# Некоторые проблемы



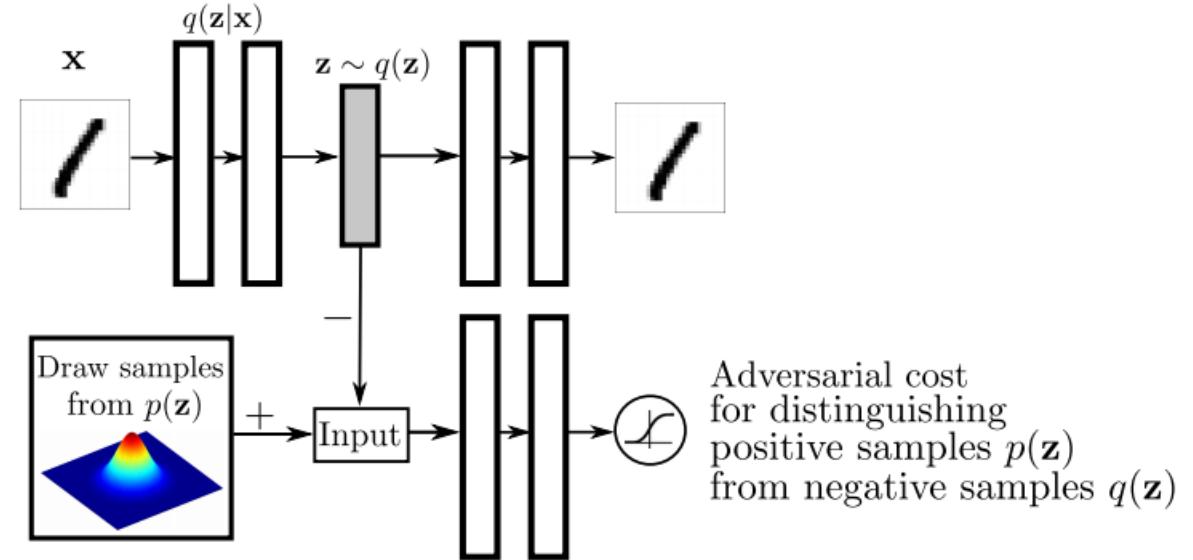
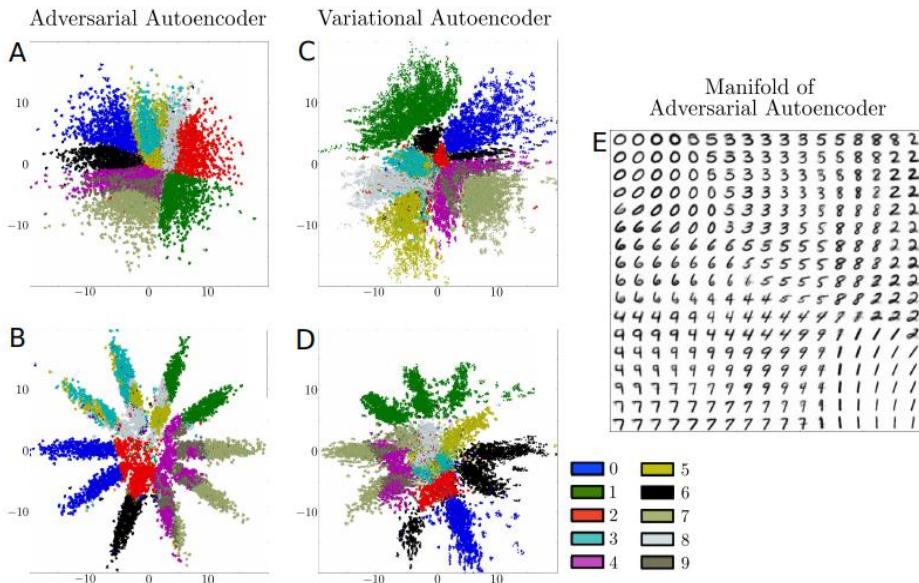
# План выступления

---

- Задачи дискриминации и генерации
- Задача генерация с точки зрения машинного обучения
- Как измерять качество генерации (изображений)
- Генеративно-состязательные сети
- Проблемы и решения
- Задача оптимальной транспортировки и функции потерь
- Интересные идеи в GANах
- Вариационные кодировщики
- **Интересные идеи в VAE**
- Немного примеров из жизни

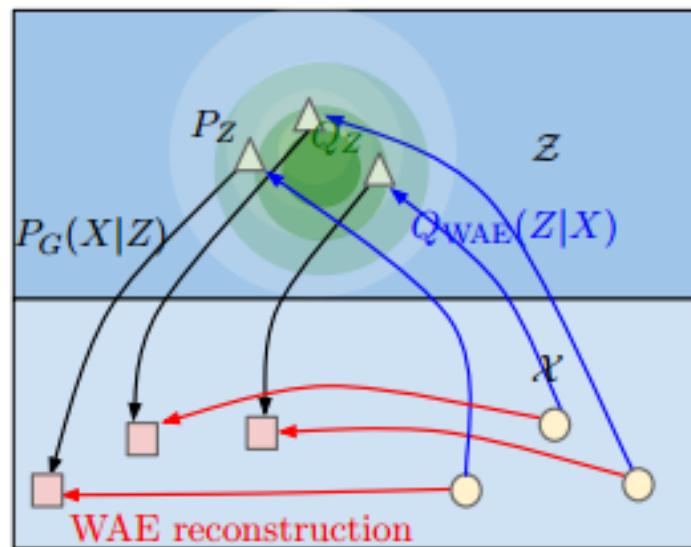
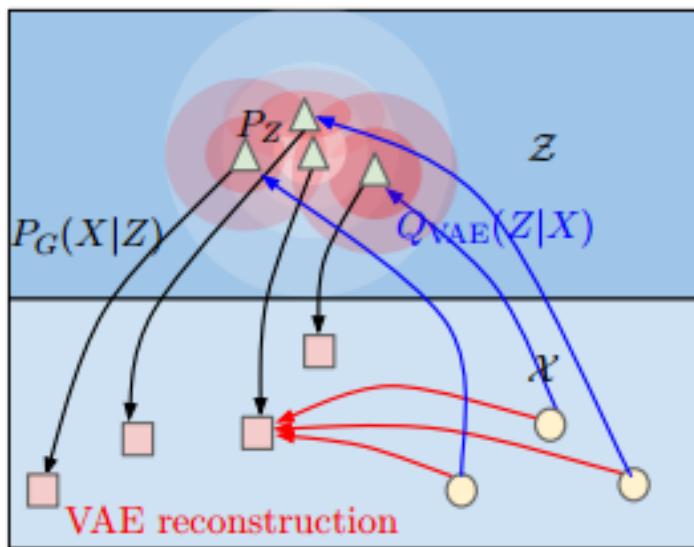
# Adversarial autoencoder

- Будем добавлять лосс за то, что распределение скрытой переменной не похоже на то, которое мы закладываем



# Wasserstein autoencoder

- Вместо похожести с исходным распределением по KL, будем сравнивать то, насколько сильно различаются



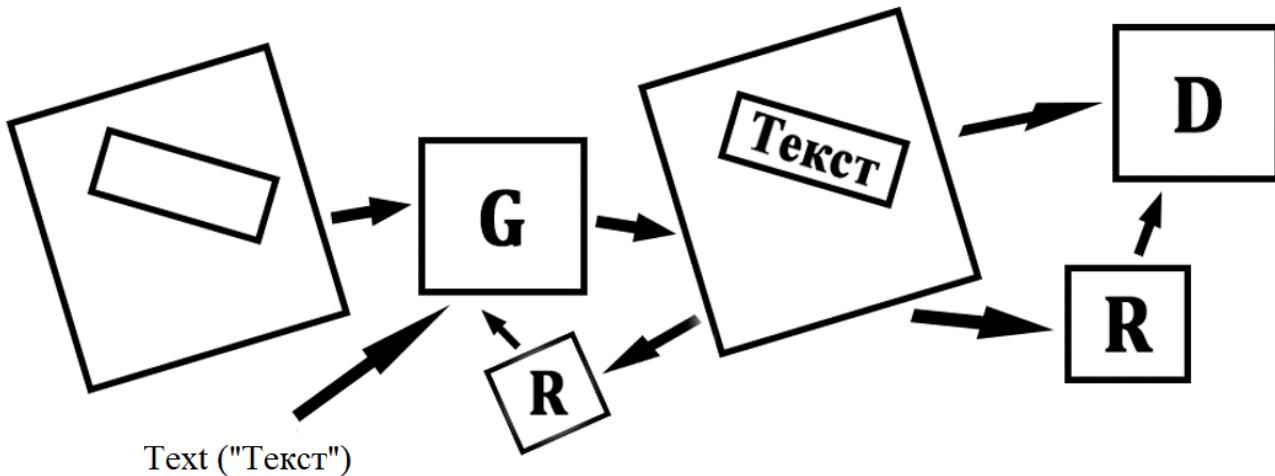
# План выступления

---

- Задачи дискриминации и генерации
- Задача генерация с точки зрения машинного обучения
- Как измерять качество генерации (изображений)
- Генеративно-состязательные сети
- Проблемы и решения
- Задача оптимальной транспортировки и функции потерь
- Интересные идеи в GANах
- Вариационные кодировщики
- Интересные идеи в VAE
- **Немного примеров из жизни**

# Аугментация данных

- Аугментация данных для задачи распознавания символов с целью увеличить качество обученной модели распознавания

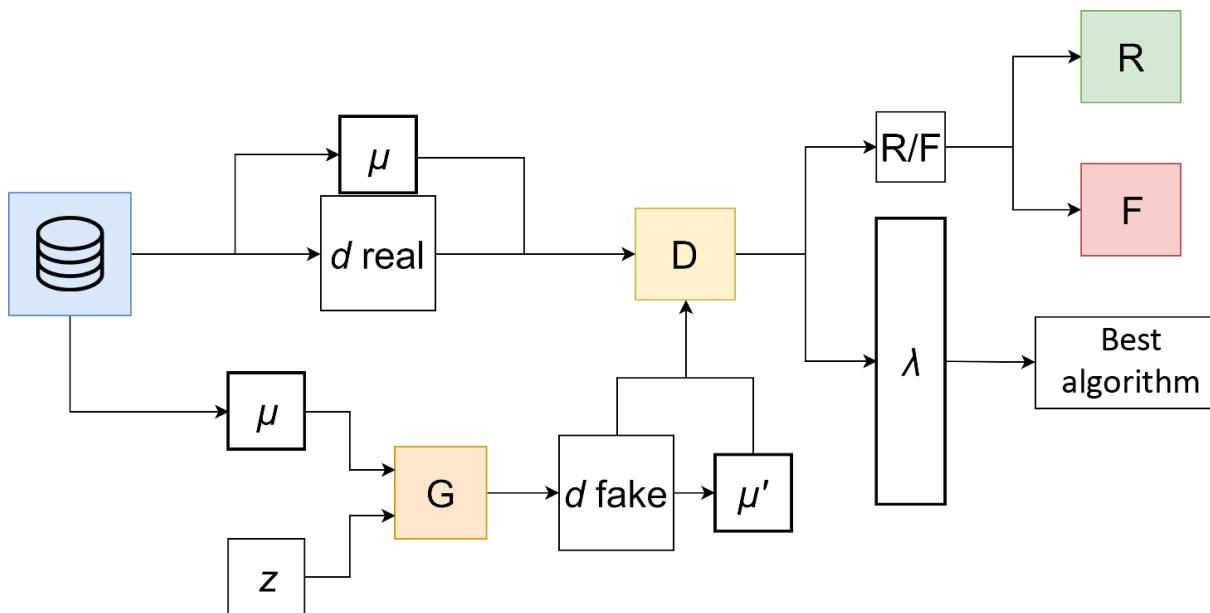


Валерия Ефимова  
ML Lab  
CV DL Senior Researcher  
PhD Student



# Генерация наборов данных

- Часто данные представлены в таблицах
- Наборы данных нужны для систем мета-обучения



A screenshot of a Microsoft Word document showing a table of order data. The table has columns for Order Date, Region, Rep, Item, Units, Unit Cost, and Total.

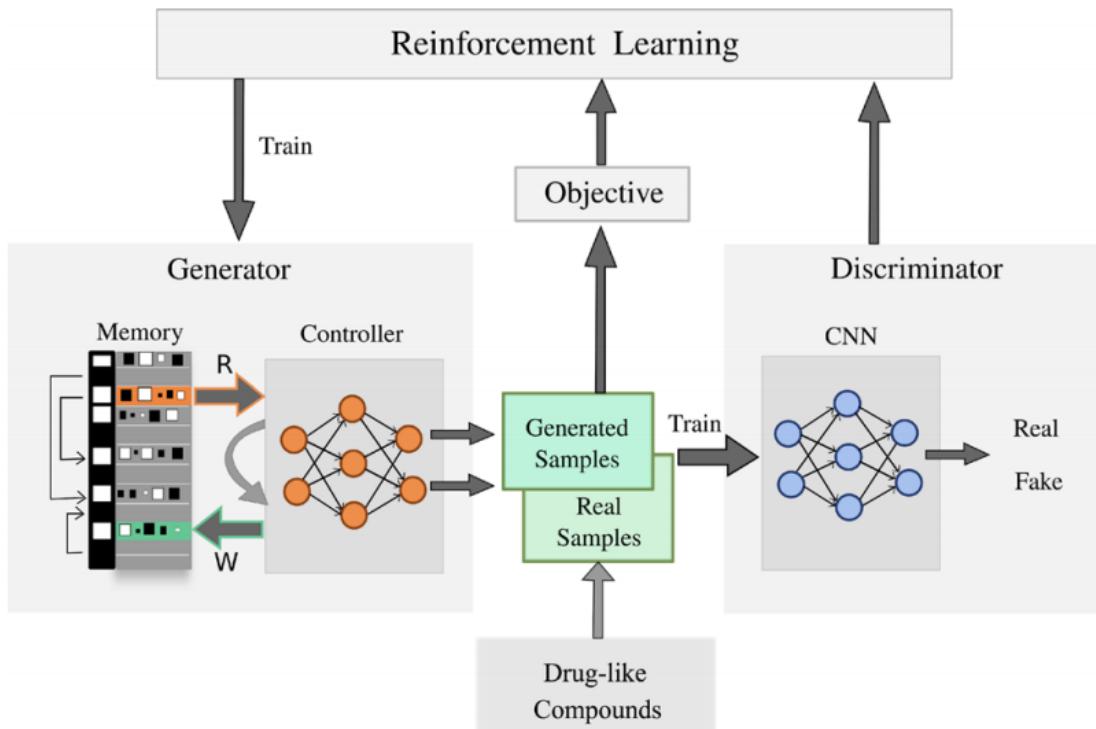
Order Date	Region	Rep	Item	Units	Unit Cost	Total
1/23/10	Ontario	Kivell	Binder	50	\$19.99	\$999.50
2/9/10	Ontario	Jardine	Pencil	36	\$4.99	\$179.64
2/26/10	Ontario	Gill	Pen	27	\$19.99	\$539.73
3/15/10	Alberta	Sorvino	Pencil	56	\$2.99	\$167.44
4/1/10	Quebec	Jones	Binder	60	\$4.99	\$299.40
4/18/10	Ontario	Andrews	Pencil	75	\$1.99	\$149.25
5/5/10	Ontario	Jardine	Pencil	90	\$4.99	\$449.10
5/22/10	Alberta	Thompson	Pencil	32	\$1.99	\$63.68

Алексей Забашта  
ML Lab  
AutoML Senior Researcher  
PhD Student



# Генерация молекул лекарств

- Поиск наиболее перспективных молекул среди всего пространства молекул



Арип Асадулаев  
ML Lab  
Fundamental ML  
Senior Researcher  
PhD Student



# Генерация архитектур нейронных сетей

---

- Поиск подходящих архитектур нейронных сетей — одно из основных направлений в AutoML
- Выбор хорошего языка описания позволит работать в пространстве таких описаний и над распределениями в таком пространстве

Вячеслав Шаламов  
ML Lab  
AutoML Researcher  
PhD Student



# Сжатие изображений

---

- По сути, алгоритмы сжатия работают как кодер и декодер в поисках оптимального представления данных
- Генеративные модели могут помогать транслировать данные в более оптимальные представления

Татьяна Полевая  
ML Lab  
CV DL Researcher  
PhD Student



# Генерация кода для проверки компилятора

---

- Необходимо находить ошибки в работе компилятора
- Для этого необходимо находить код, который вызывает такие ошибки

Виктор Петухов  
JetBrains  
PhD Student



# СПАСИБО ЗА ВНИМАНИЕ

Андрей Фильченков



Темы дипломов

и курсовых

[cutt.ly/zgntuvf](https://cutt.ly/zgntuvf)

Лаборатория машинного обучения

[instagram.com/itmo.mllab](https://instagram.com/itmo.mllab)



[vk.com/itmo.mllab](https://vk.com/itmo.mllab)



[t.me/itmo\\_mllab](https://t.me/itmo_mllab)



[youtube.com/c/MLLabITMO](https://youtube.com/c/MLLabITMO)



# Заголовок

---

- Текст
  - Текст

