



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

ГЕНЕРАТИВНЫЕ МОДЕЛИ

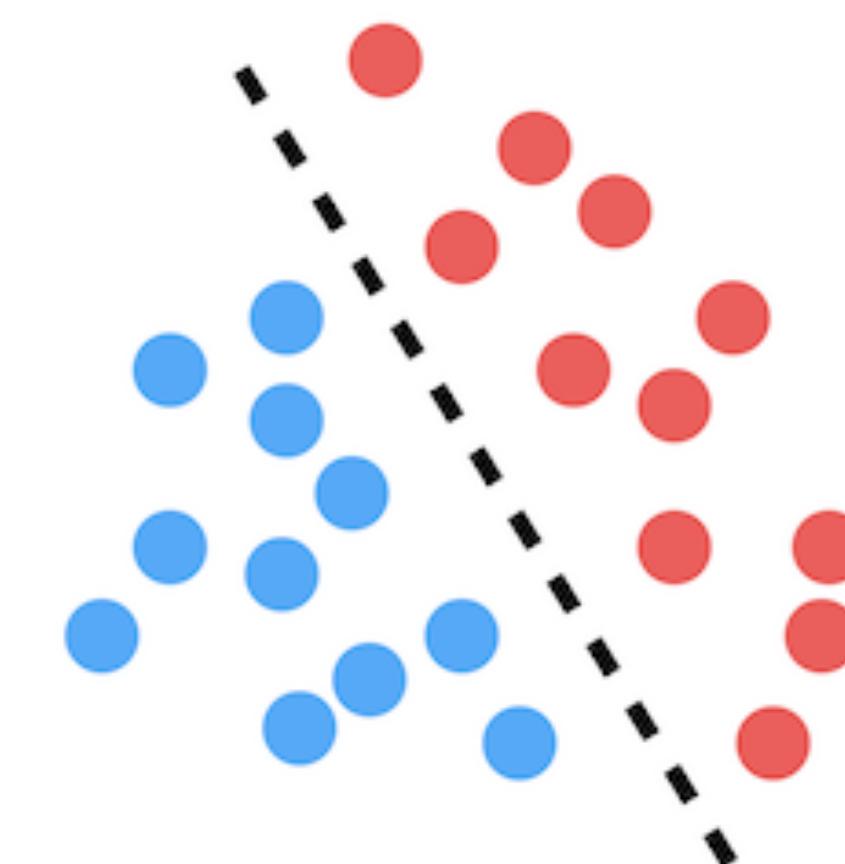
Токкожин Арсен
БПМИ191



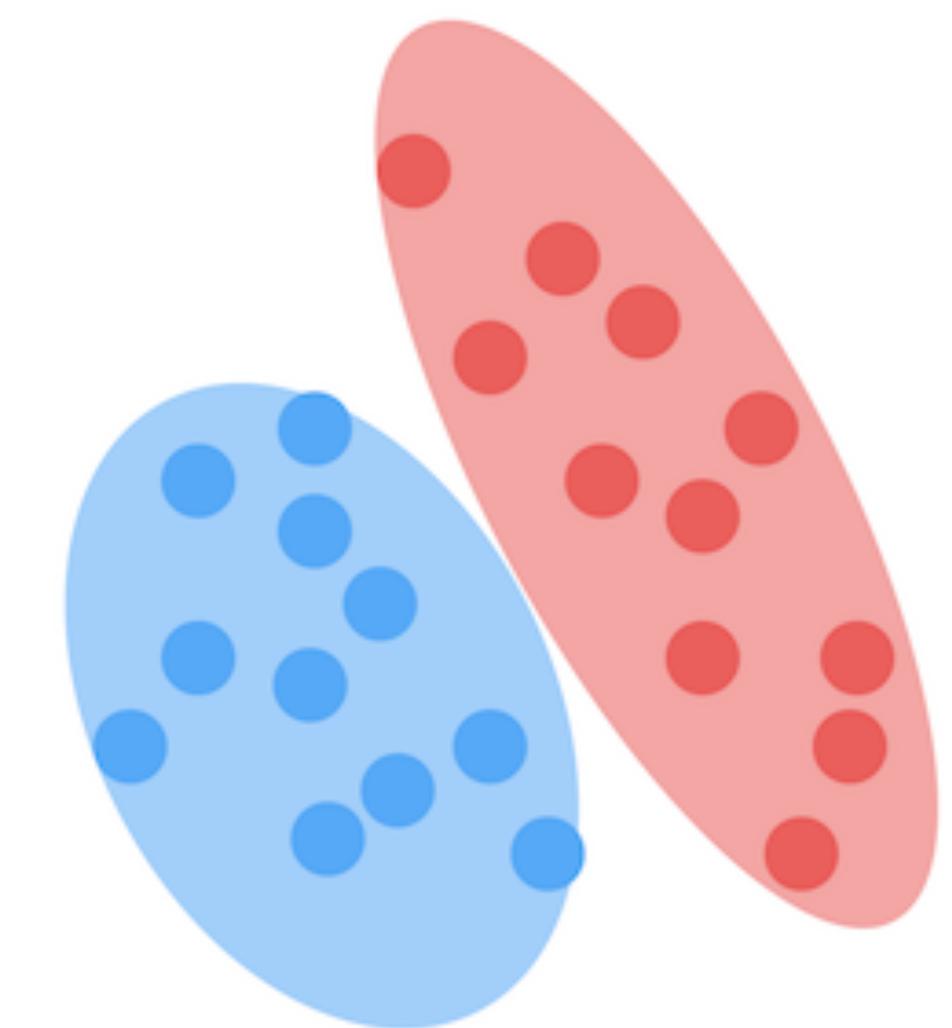
Два типа моделей

- Discriminative
(дискриминативные или условные)
- Generative
(генеративные)

Discriminative



Generative





Discriminative models

- Модель условной вероятности Y по наблюдаемыми переменным X
- С помощью обучающих данных X оцениваем параметры $P(Y|X)$



Generative models

- Модель условной вероятности X , по целевой переменной Y
- С помощью обучающих данных X мы оцениваем параметры $P(X|Y)$, $P(Y)$

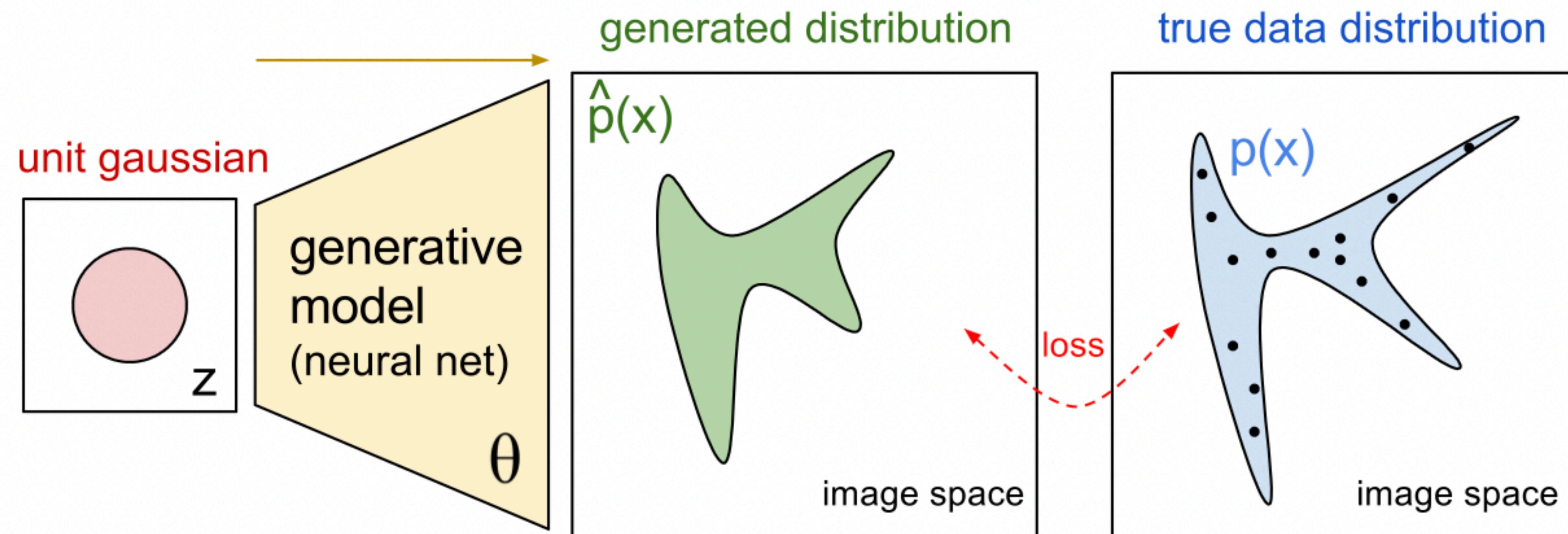


Generative models





Generative models



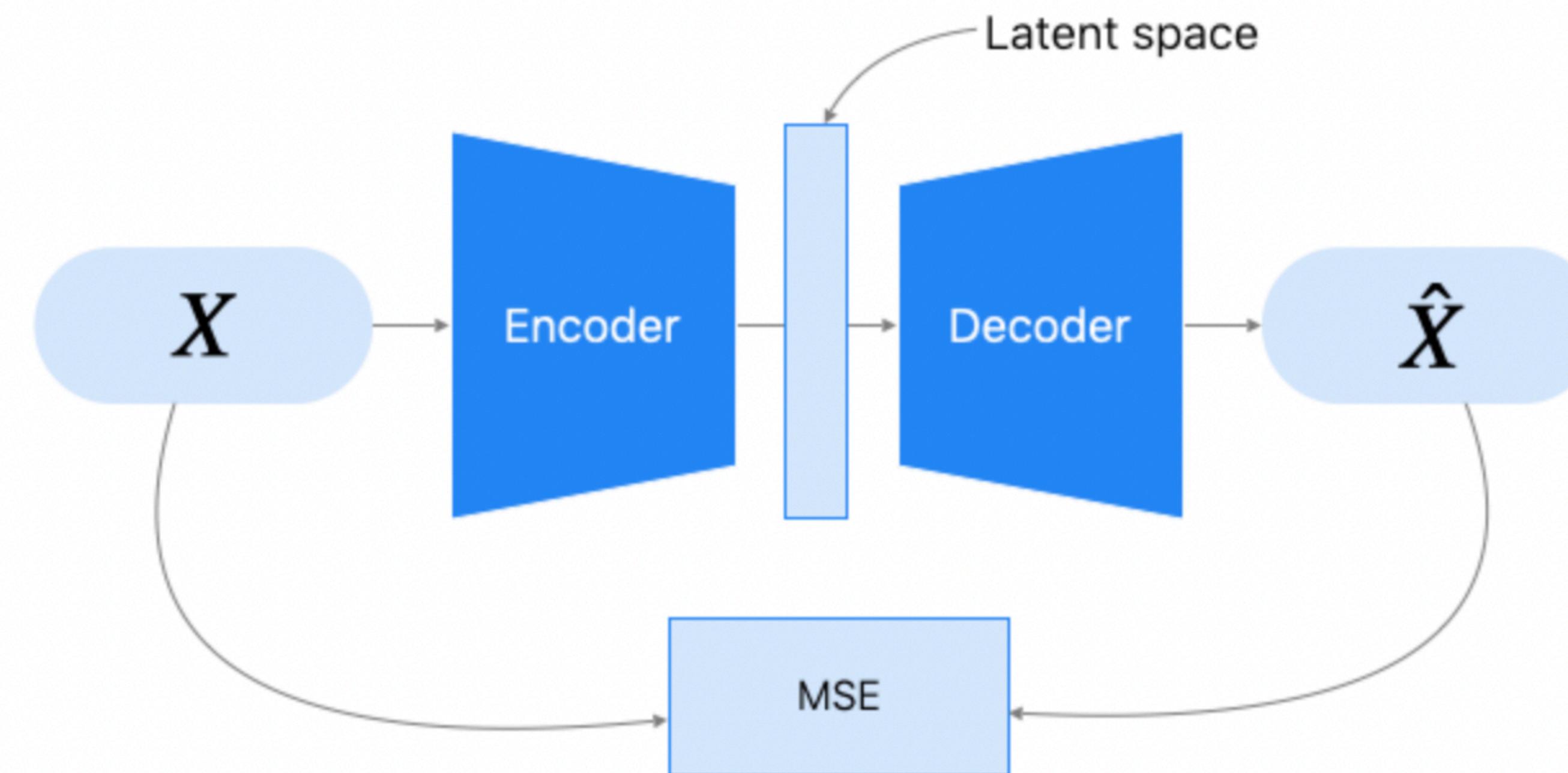


Примеры генеративных моделей

- Generative Adversarial Networks (GANs)
- Variational Autoencoders (VAEs)
- Auto-Regressive (PixelRNN, PixelCNN)



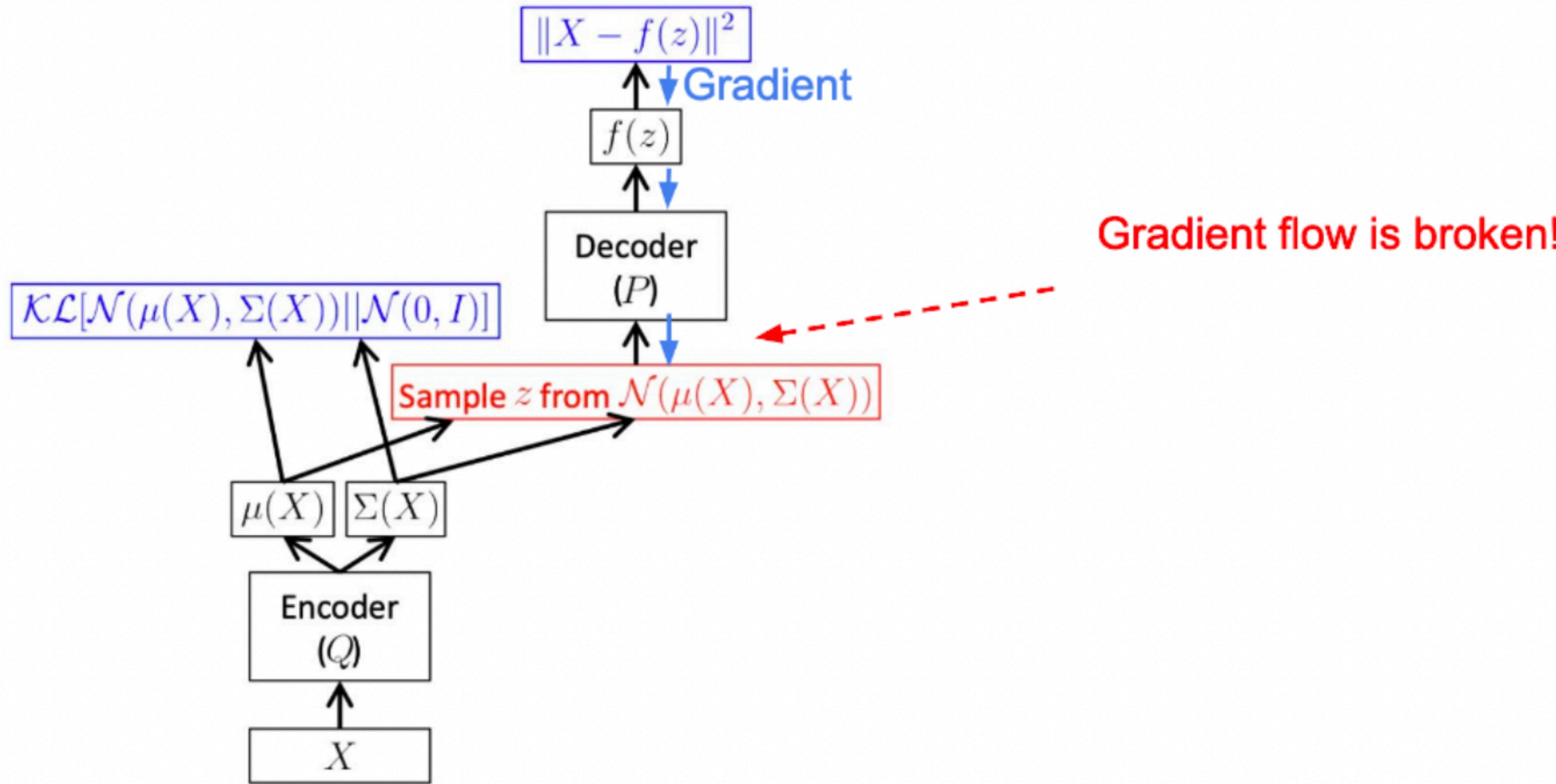
VAE (Variational Autoencoder)

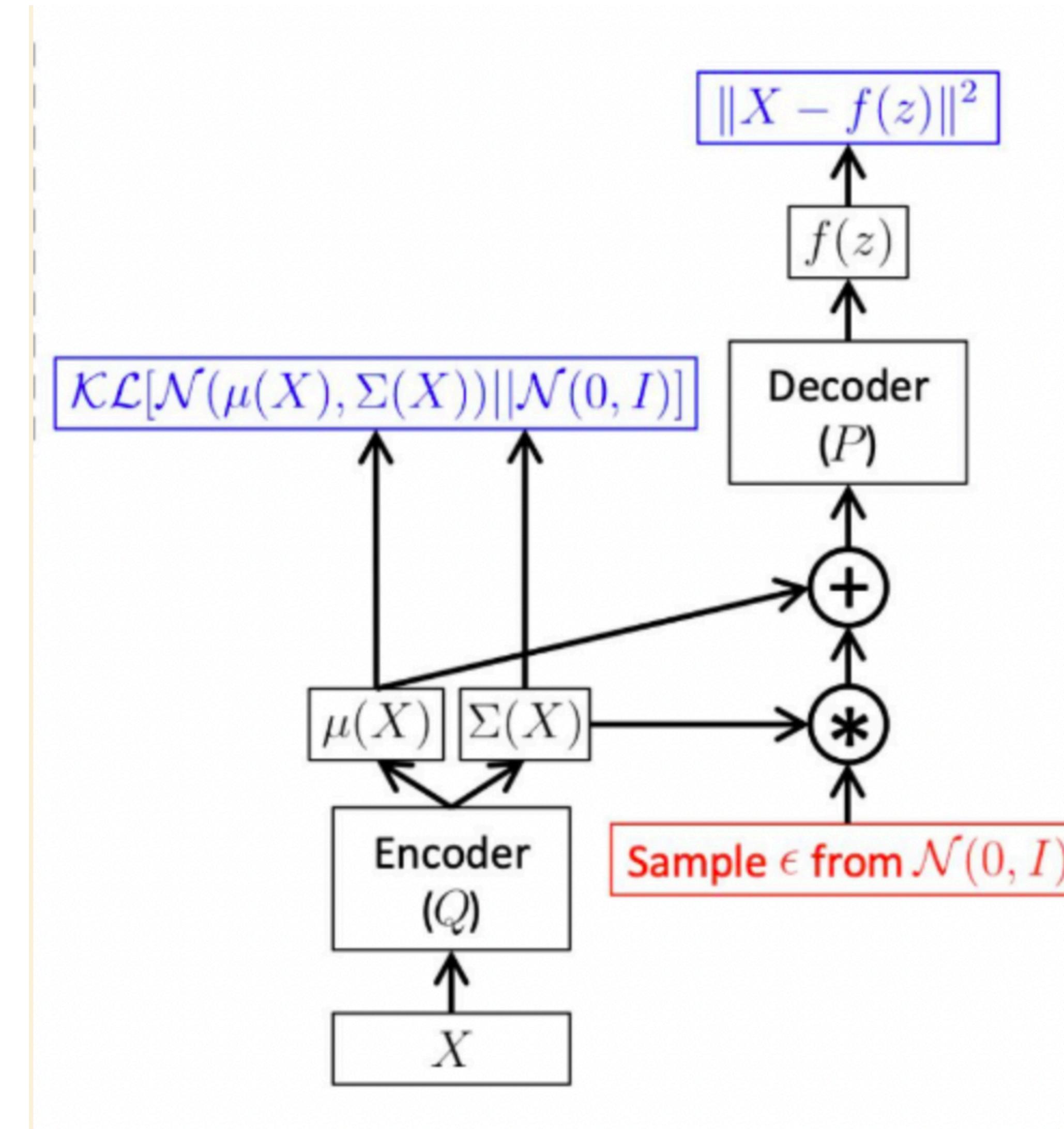




Алгоритм VAE

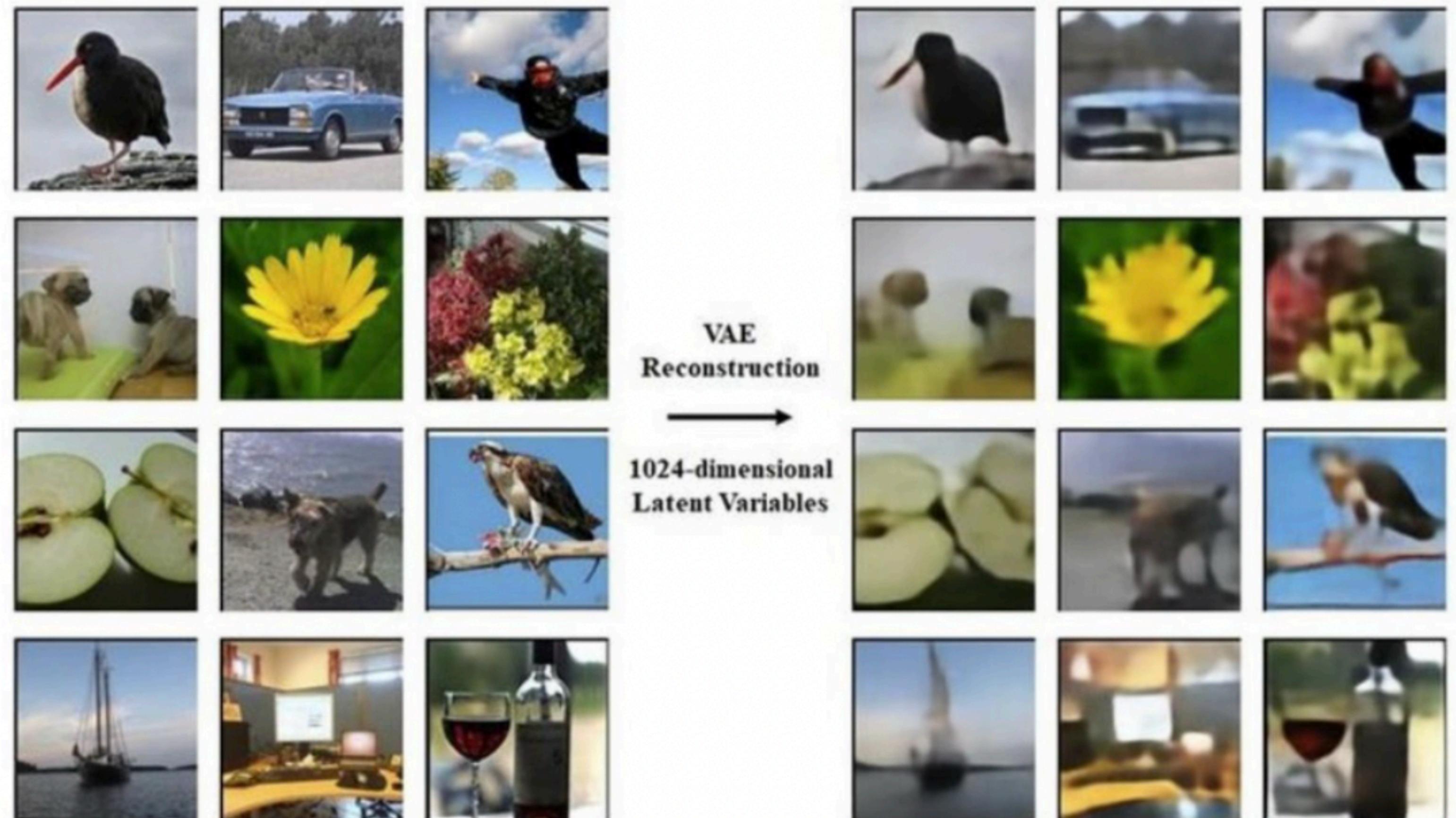
1. Функция, которая использована в основе encoder-а $f(X)$ выдает параметры нормального распределения (μ, Σ) , которое должно быть максимально похоже на стандартное нормальное распределение $N(0, I)$
2. Берем sample x' из распределения $N(\mu, \Sigma)$
3. Пропускаем наш x' через decoder и получаем







VAE





ПЛЮСЫ И МИНУСЫ

Плюсы:

- Просто обучить
- Модель умеет сжимать данные

Минусы:

- Итоговое изображение нечеткое



VAE

ПРИМЕНЕНИЕ

- Уменьшение шума в данных
- Уменьшение размерности данных
- Генерация новых объектов(например картинок)



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

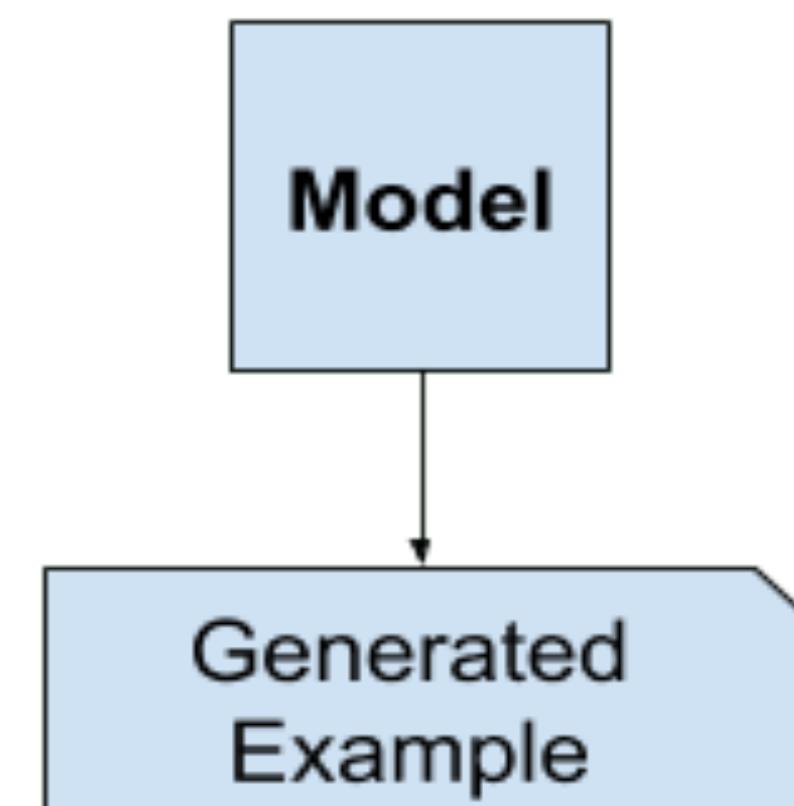
ГЕНЕРАТИВНО-СОСТАЗАТЕЛЬНЫЕ НЕЙРОННЫЙ СЕТИ (GANS)

Василевская Юля
БПМИ191

GAN - Generative Adversarial Network

Генеративно состязательная нейронная сеть

- Генерируем данные из заданного распределения
- Принцип таких моделей построен на соревновании (состязании) двух нейронных сетей





Идея работы

Генеративно состязательных нейронных сетей

- Генератор (порождает данные) - фальшивомонетчик
 - ✓ цель - научиться генерировать правдоподобные данные
- Дискриминатор (отличает реальные данные от сгенерированных) - банкир
 - ✓ цель - не дать фальшивомонетчику себя обмануть и распознать фальшивки



Формализм, которого так не хватало

- Генератор - G (от generator)

вход: шум z

выход: $G(z)$ - сгенерированные данные из распределения

- Дискриминатор - D (от discriminator)

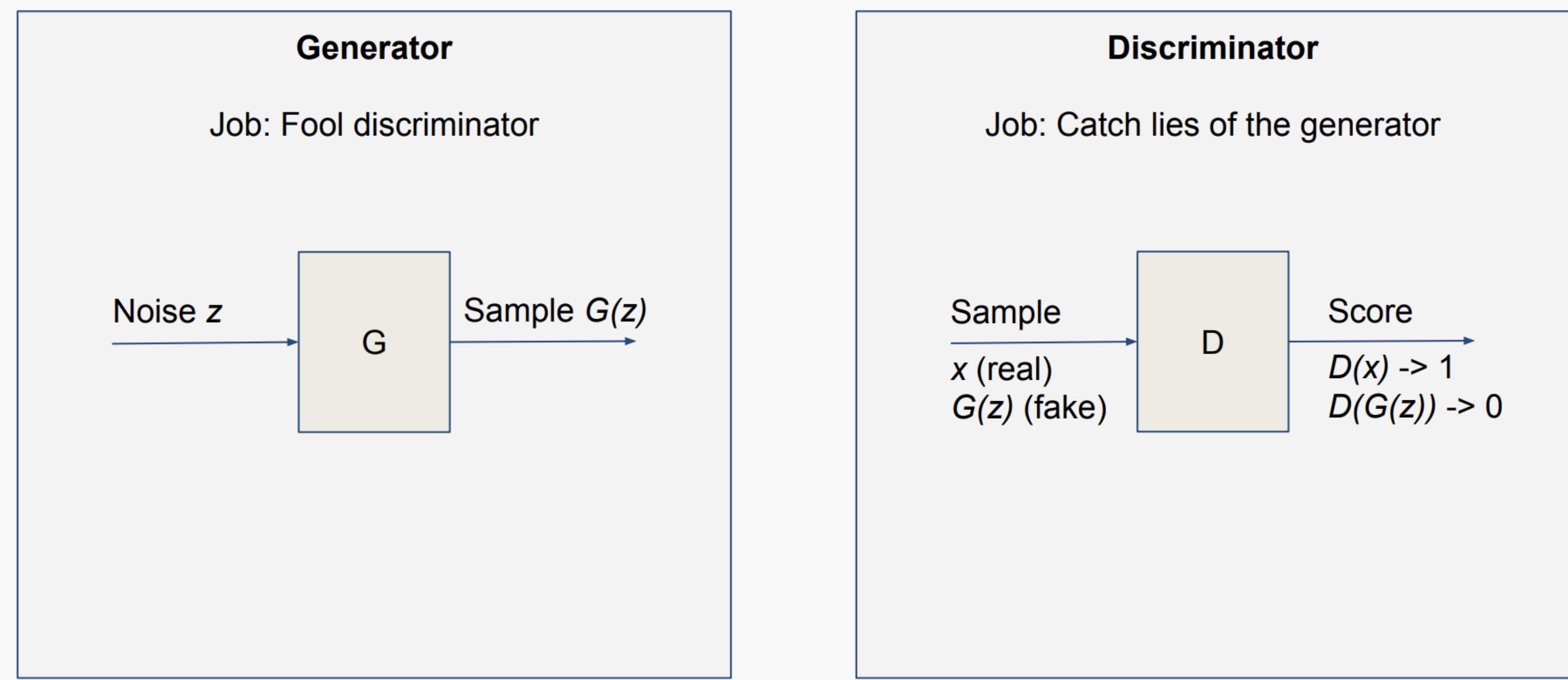
вход: пример данных из распределения (реальный или сгенерированный)

выход: вероятность принадлежности к реальным данным



Формализм, которого так не хватало

- Генератор
функция успеха, которую хотим максимизировать: $D(G(z))$
- Дискриминатор - D (от discriminator)
функция успеха, которую хотим максимизировать: $D(x)(1 - D(G(z)))$,
здесь x - объект из реальных данных, G(z) - сгенерированный объект



!! Веса обеих моделей обновляем в сторону градиента функции успеха !!



Алгоритм обучения

1. n настоящих и m сгенерированных объектов.
2. Фиксируем веса генератора, обновляем веса дискриминатора.
3. Фиксируем веса дискриминатора и обновляем веса генератора.
4. Повторяем 1-3, пока дискриминатор и генератор не придут в равновесие*.

* состояние, когда каждая из моделей не может “обмануть” другую

Примеры

Инженер Филипп Ван

[This Cat Does Not Exist](#)



[This Horse Does Not Exist](#)

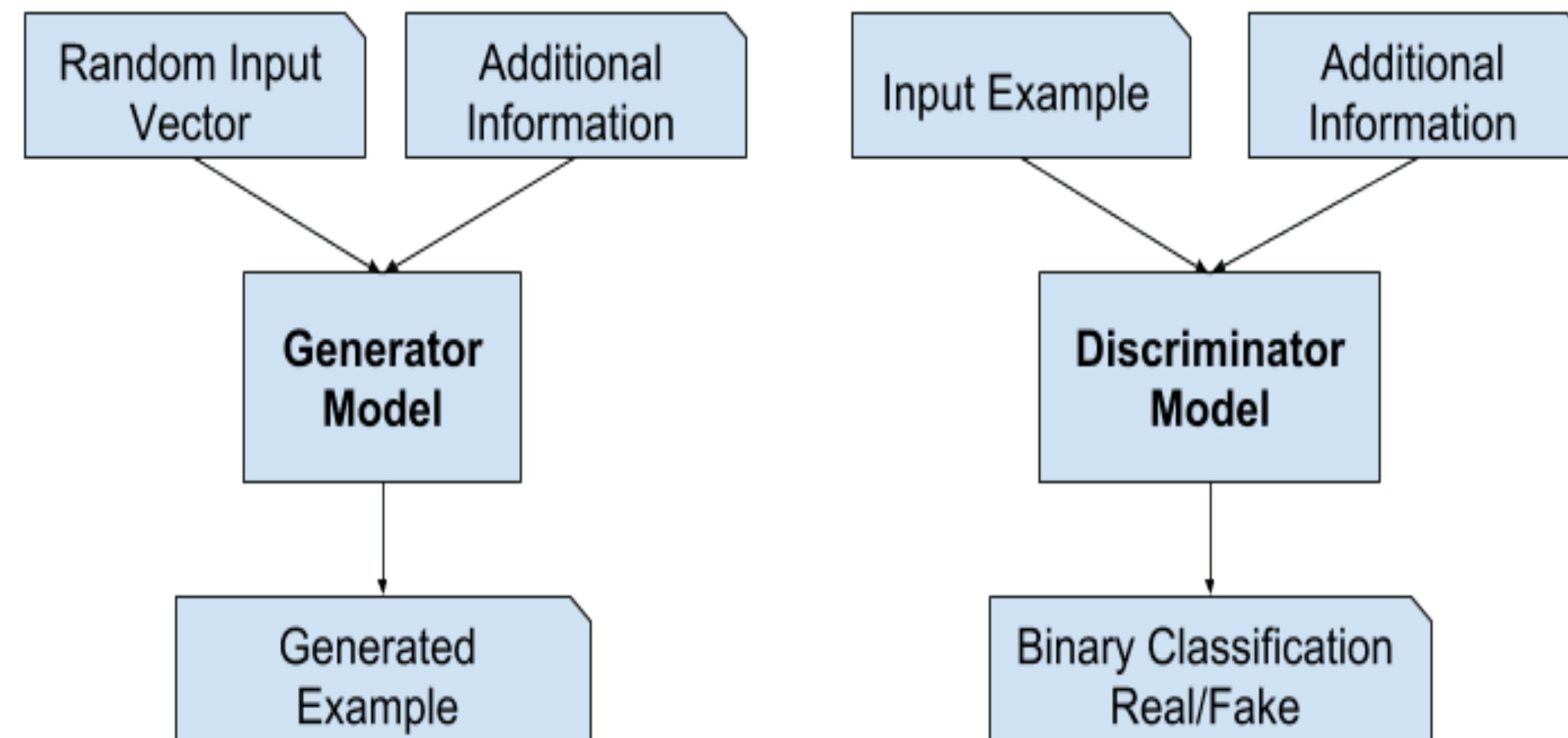


[This Person Does Not Exist](#)



Условные ганы

На вход соревнующимся моделям подаётся дополнительная информация





Streamlit Face-GAN Demo

Features

Show advanced options

Control which features?

Young Smiling
Male Brown_Hair

Young: 45

Smiling: 65

Male: 60

Brown_Hair: 50



Проблемы

- Затухание градиента
- Колебания ф. успеха из-за независимости обновления параметров G и D
- Модальный коллапс (генерация выглядящих одинаково картинок)
- Высокая чувствительность к гиперпараметрам
- Проблема запутывания

* Может помочь Вассерштейн ган (расстояние Вассерштейна)



Применение

- [CycleGAN](#): меняет изображения с одного домена на другой, например, лошадей на зебр;
- [SRGAN](#): создает изображения с высоким разрешением из более низкого разрешения;
- [Pix2Pix](#): создает изображения по семантической окраске;
- [StackGAN](#): создает изображения по заданному тексту;
- [MidiNet](#): генерирует последовательность нот, таким образом, создает мелодию.



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

NORMALIZING FLOWS

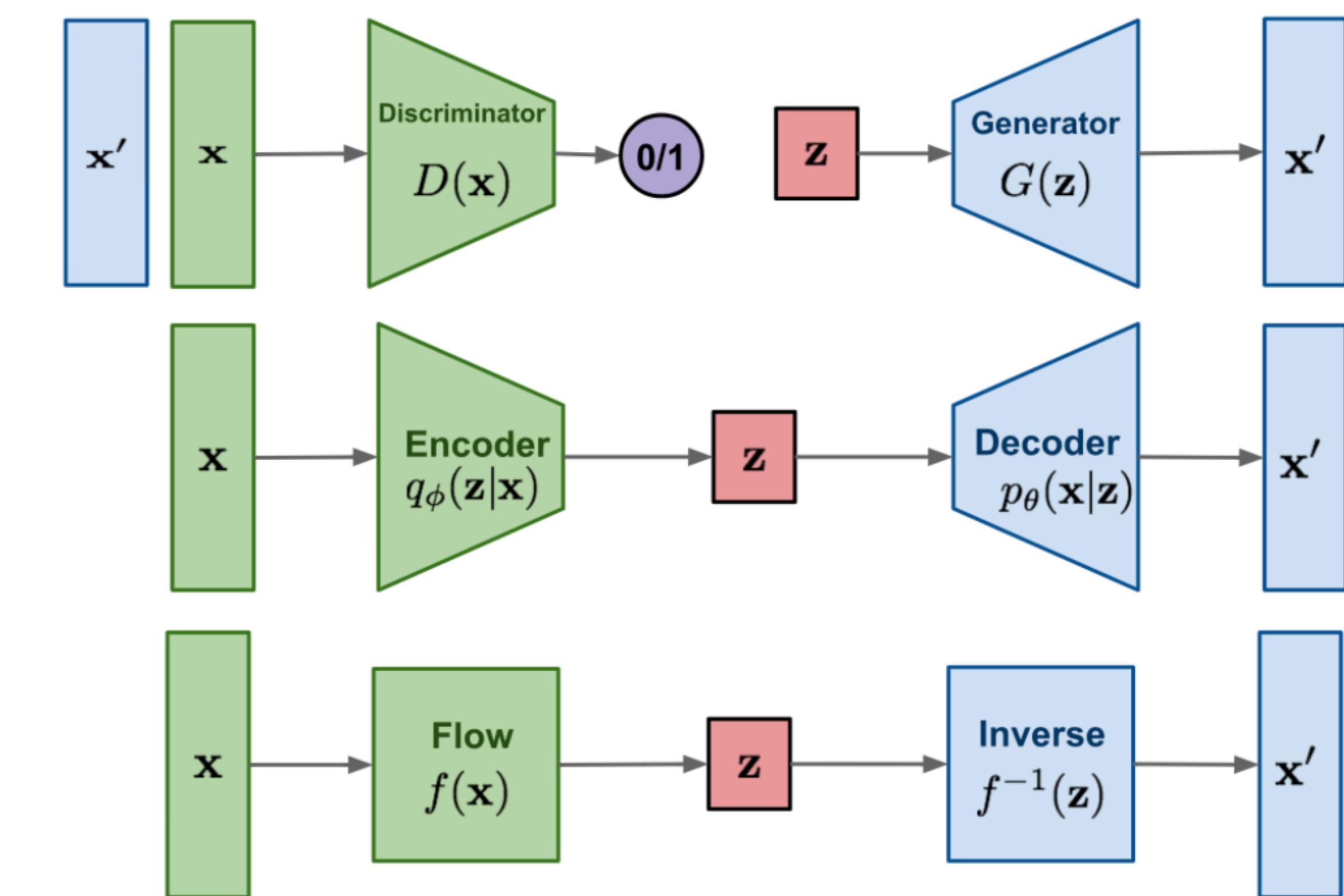
Федоров Григорий
БПМИ191



VAE и GAN - не аппроксимируют функцию плотности распределения данных

приближение плотности весьма непростая задача

генеративные модели решают эту задачу с помощью нормализации потоков



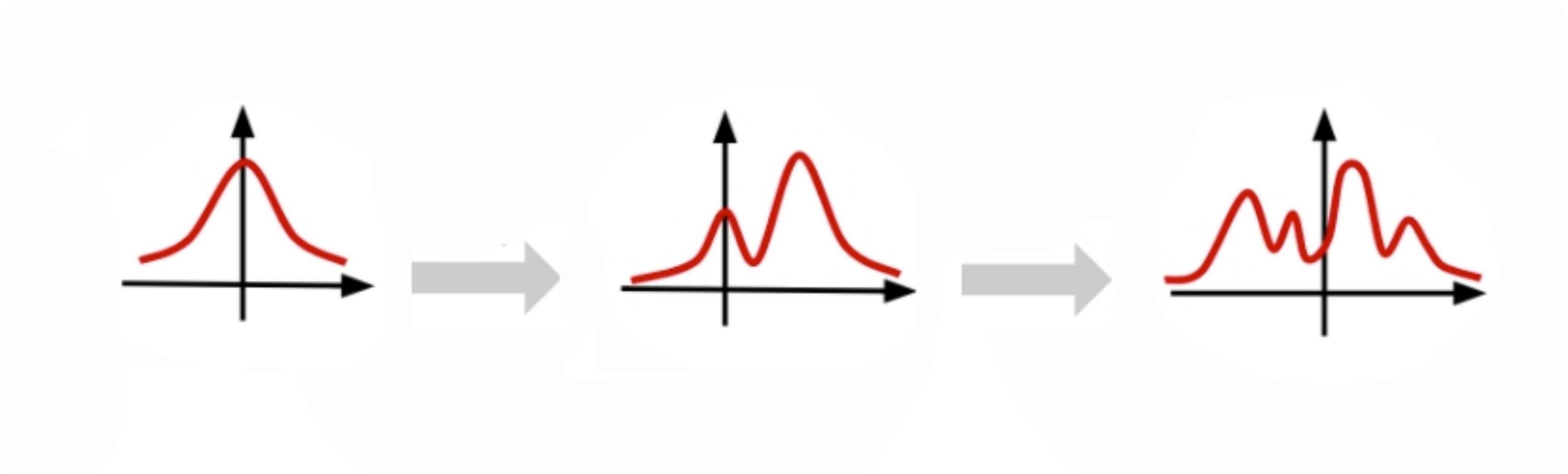


Какие задачи сможем решать, зная плотность распределения данных?

- Выбираем не наблюдаемые, но реалистичные данные - генерация
- Оценка плотности - прогнозирование
- Заполнение неполных выборок
- Нахождение выбросов
- Кластеризация
- И многие другие задачи, где необходимо знать распределение данных



Как приблизить распределение?





Формула замены переменной

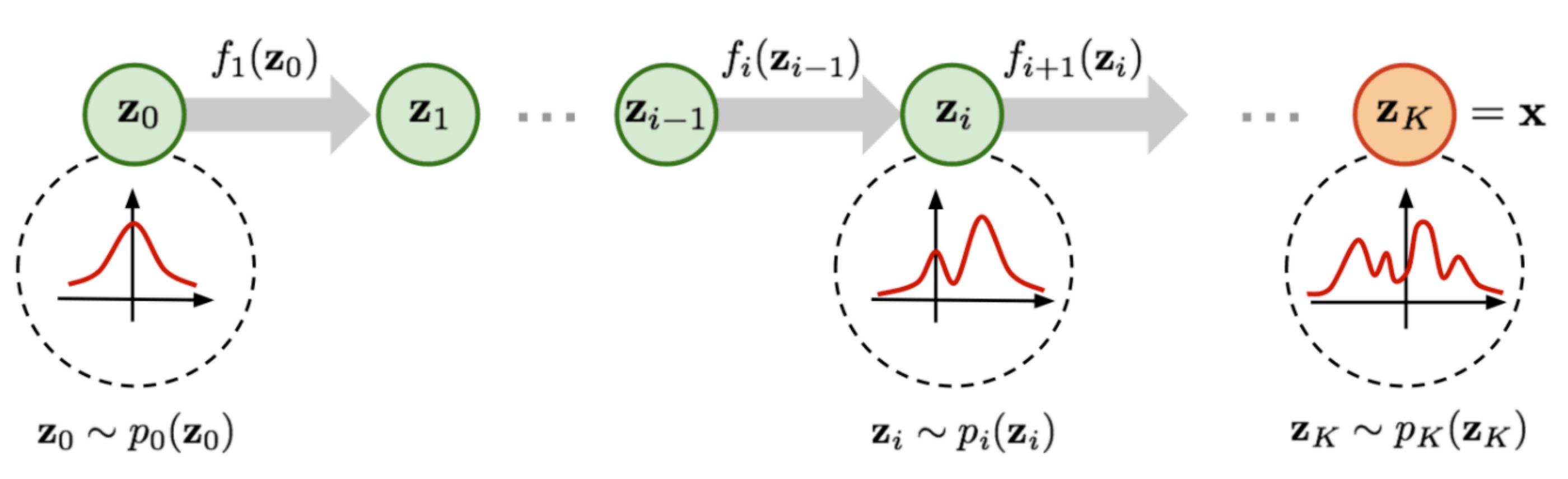
$$z \sim p_z(t)$$

$$x = f(z) \quad z = f^{-1}(x)$$

$$p_x(t) = p_z(f^{-1}(x)) \cdot \left| \det \frac{df^{-1}(x)}{dx} \right| = p_z(f^{-1}(x)) \cdot \left| \det^{-1} \frac{df(x)}{dx} \right|$$



Что такое поток?



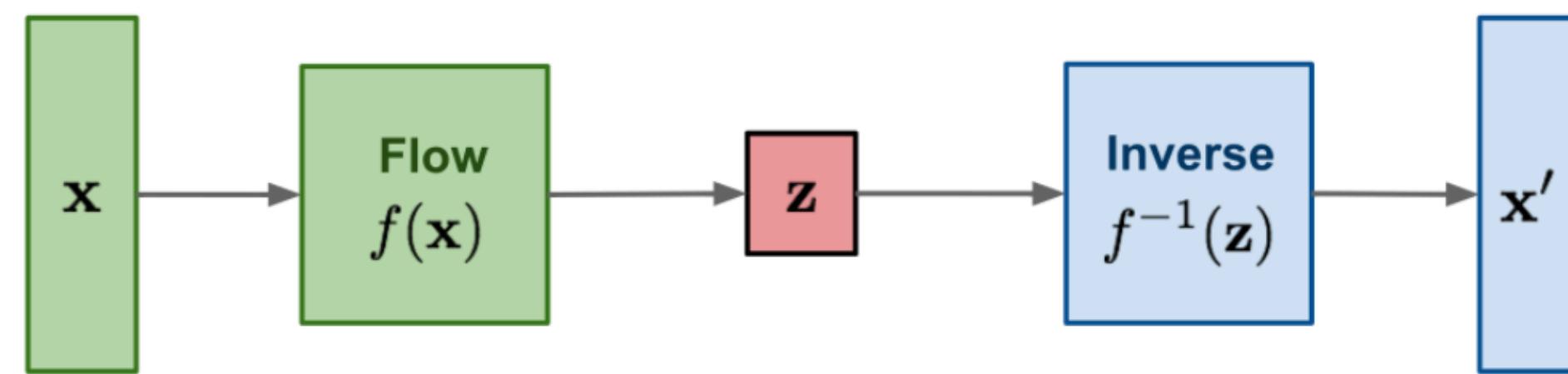
$$\mathbf{z}_{i-1} \sim p_{i-i}(\mathbf{z}_{i-1})$$

$$\mathbf{z}_i = f_i(\mathbf{z}_{i-1})$$

$$\mathbf{z}_{i-1} = f_i^{-1}(\mathbf{z}_i)$$



Как выучить плотность?



$$\mathbf{x} = \mathbf{z}_K = f_K \circ f_{K-1} \circ \cdots \circ f_1(\mathbf{z}_0)$$

$$\begin{aligned}\log p(\mathbf{x}) &= \log \pi_K(\mathbf{z}_K) = \log \pi_{K-1}(\mathbf{z}_{K-1}) - \log \left| \det \frac{df_K}{d\mathbf{z}_{K-1}} \right| \\ &= \log \pi_{K-2}(\mathbf{z}_{K-2}) - \log \left| \det \frac{df_{K-1}}{d\mathbf{z}_{K-2}} \right| - \log \left| \det \frac{df_K}{d\mathbf{z}_{K-1}} \right| \\ &= \dots \\ &= \log \pi_0(\mathbf{z}_0) - \sum_{i=1}^K \log \left| \det \frac{df_i}{d\mathbf{z}_{i-1}} \right|\end{aligned}$$



Как выучить плотность?

$$\mathcal{L}(D) = -\frac{1}{|D|} \sum_{\mathbf{x} \in D} \log p(\mathbf{x})$$



RealNVP

$$f : \mathbf{x} \mapsto \mathbf{y}$$

$$\begin{cases} \mathbf{y}_{1:d} &= \mathbf{x}_{1:d} \\ \mathbf{y}_{d+1:D} &= \mathbf{x}_{d+1:D} \odot \exp(s(\mathbf{x}_{1:d})) + t(\mathbf{x}_{1:d}) \end{cases}$$

$$\begin{cases} \mathbf{x}_{1:d} &= \mathbf{y}_{1:d} \\ \mathbf{x}_{d+1:D} &= (\mathbf{y}_{d+1:D} - t(\mathbf{y}_{1:d})) \odot \exp(-s(\mathbf{y}_{1:d})) \end{cases}$$

$$\mathbf{J} = \begin{bmatrix} \mathbb{I}_d & \mathbf{0}_{d \times (D-d)} \\ \frac{\partial \mathbf{y}_{d+1:D}}{\partial \mathbf{x}_{1:d}} & \text{diag}(\exp(s(\mathbf{x}_{1:d}))) \end{bmatrix}$$



NICE

$$f : \mathbf{x} \mapsto \mathbf{y}$$

$$\begin{cases} \mathbf{y}_{1:d} &= \mathbf{x}_{1:d} \\ \mathbf{y}_{d+1:D} &= \mathbf{x}_{d+1:D} + m(\mathbf{x}_{1:d}) \end{cases} \Leftrightarrow \begin{cases} \mathbf{x}_{1:d} &= \mathbf{y}_{1:d} \\ \mathbf{x}_{d+1:D} &= \mathbf{y}_{d+1:D} - m(\mathbf{y}_{1:d}) \end{cases}$$

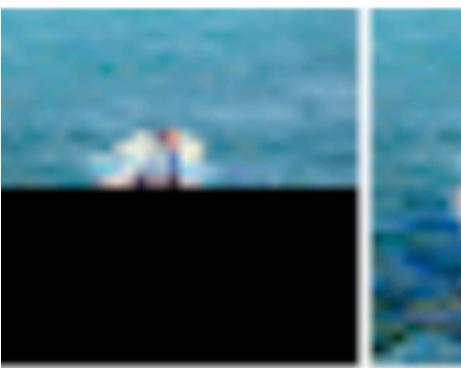


PixelRNN

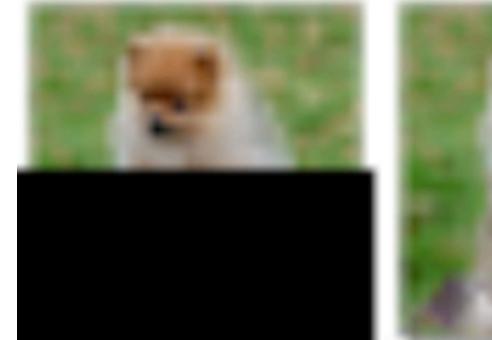
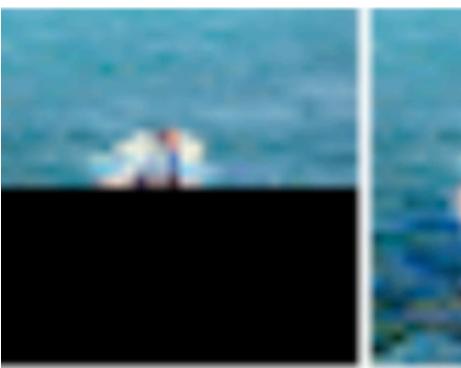
occluded



completions

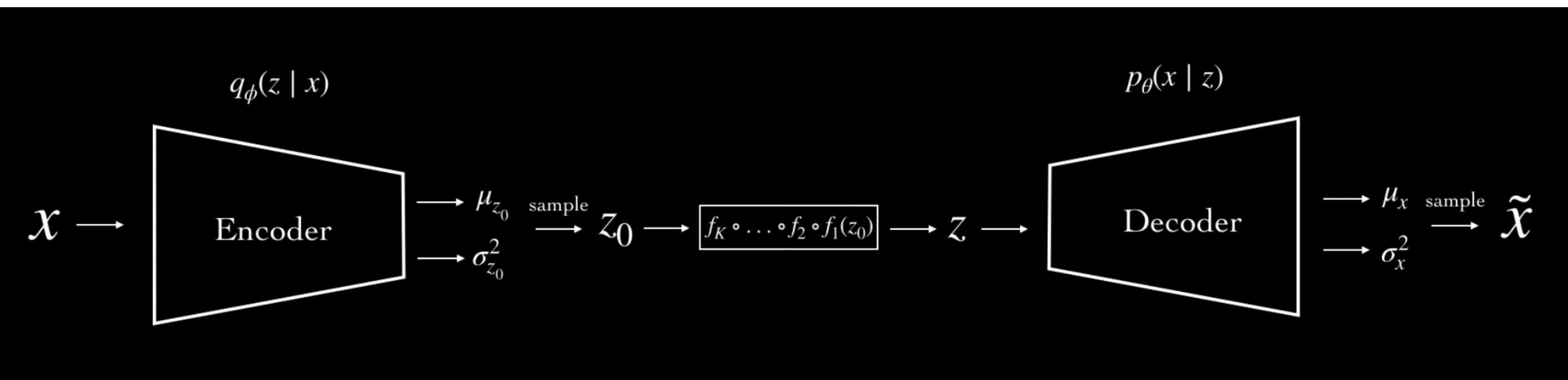
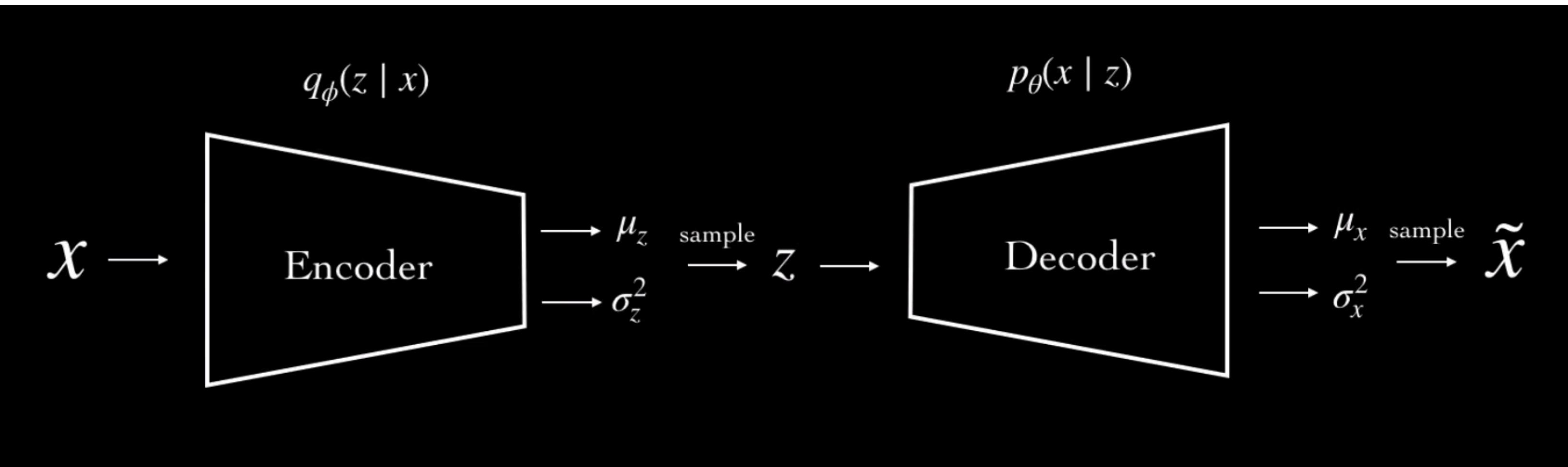


original





VAE + NF





+

- Стабильное обучение
- Гораздо проще сходится

-

- Сложное латентное пространство
- Сравнительно низкое качество генерации



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Спасибо за внимание!