# DENSITY ESTIMATION USING REAL NVP

презентацию подготовил студент ВШЭ, Трошин Сергей.
2019

# Что будет рассказано?

1. Задача генеративной модели
2. Формула замены переменной
3. Affine Coupling layer
4. Правило композиции
5. Ранняя факторизация
6. Batch Normalisation
7. Результаты

# Общая постановка задачи для генеративных моделей

Задача генеративных моделей - генерировать новые данные по имеющимся обучающим примерам.

$X, y$ - выборка

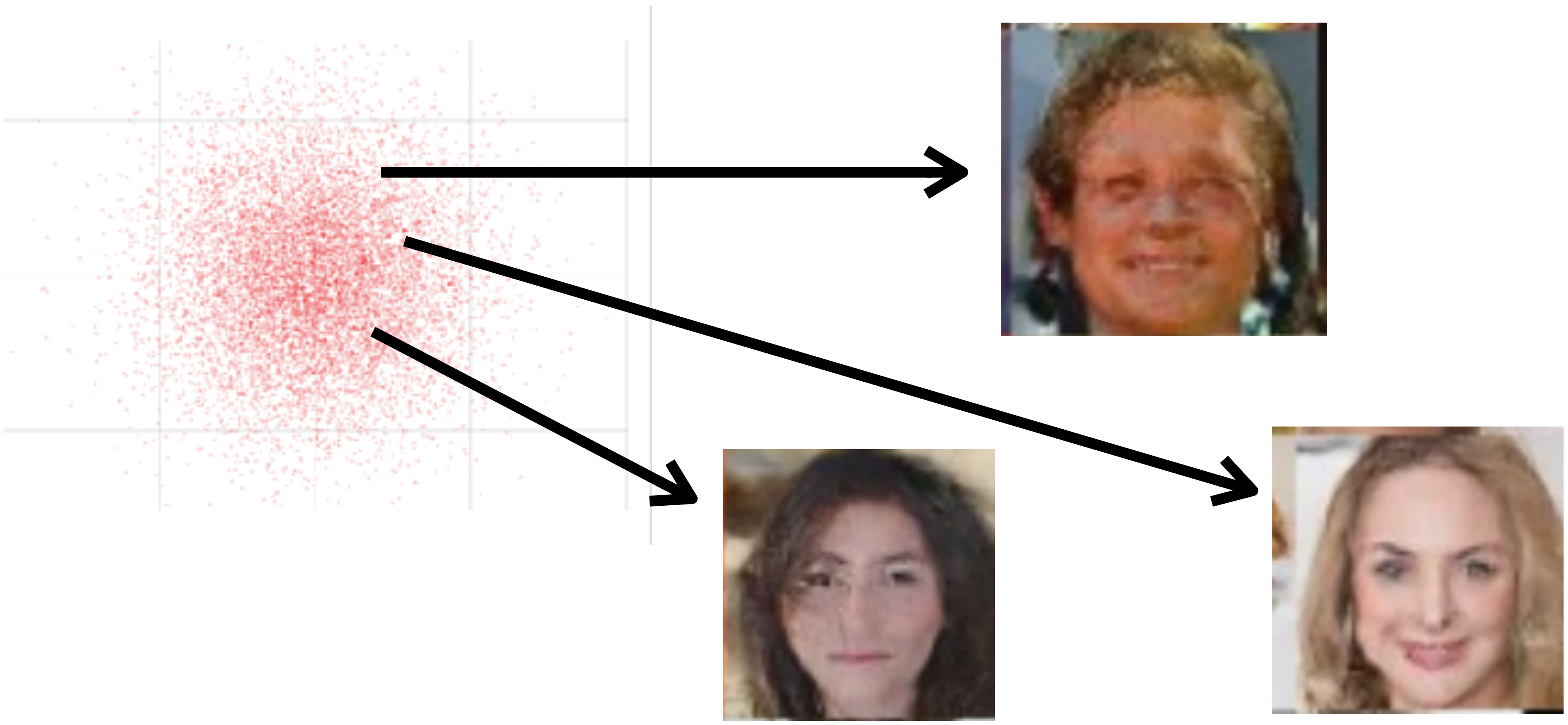
$$X, y \sim p(X, y)$$

$$\hat{p}(X, y) - ?$$

Хотим новые объекты из распределения $\hat{p}(X \mid y)$

# Общая постановка задачи для генеративных моделей

**Зачастую, генеративные модели учатся, принимая на вход шум (точки из латентного пространства) и, возможно, другую мета-информацию, порождать объект, например, картинку.**



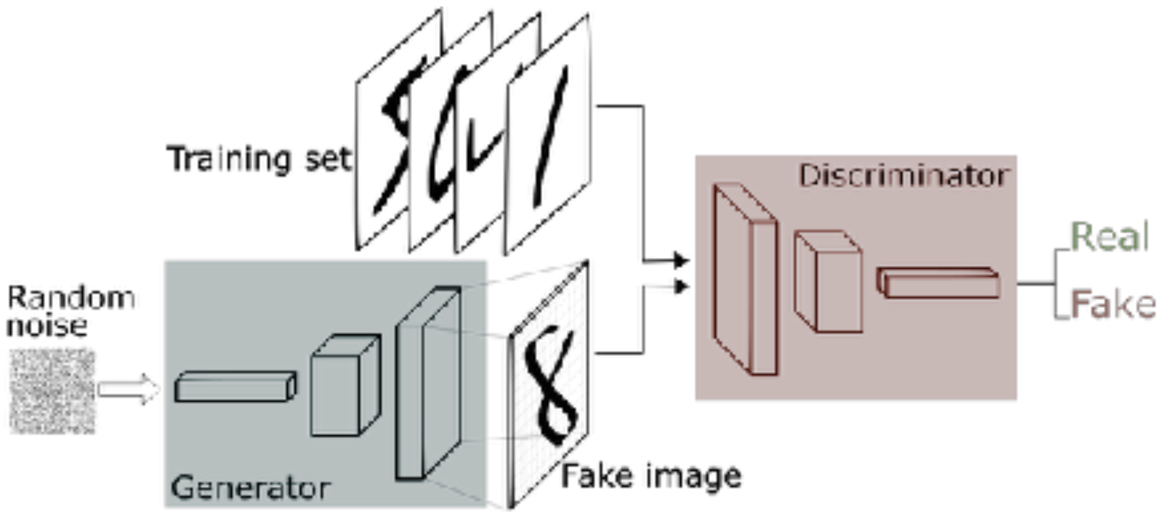
# Примеры задач

Генерация музыки и речи

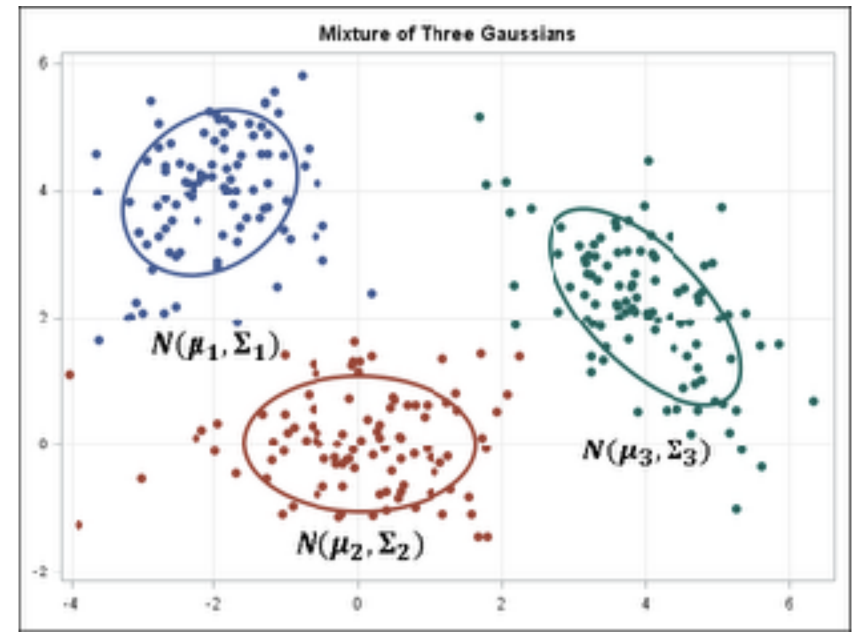
Ренедеринг в реальном времени

**Работая с реальными системами, мы хотим быструю генерацию, точную вычислимость, разнообразие.**

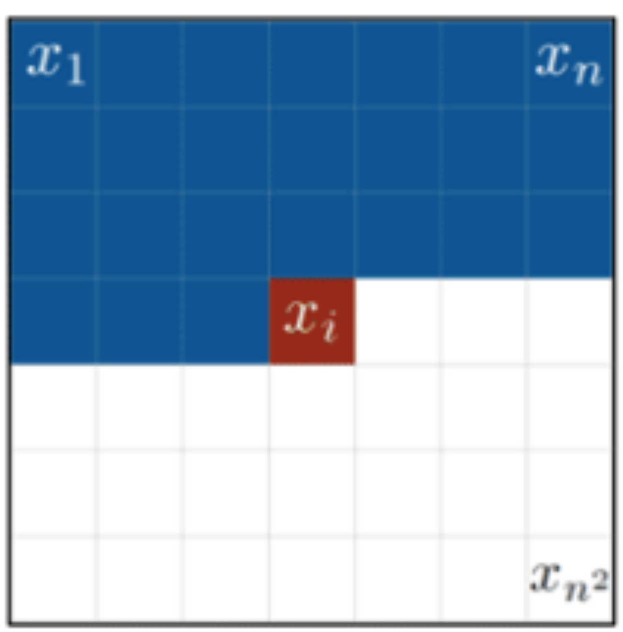
# Подходы

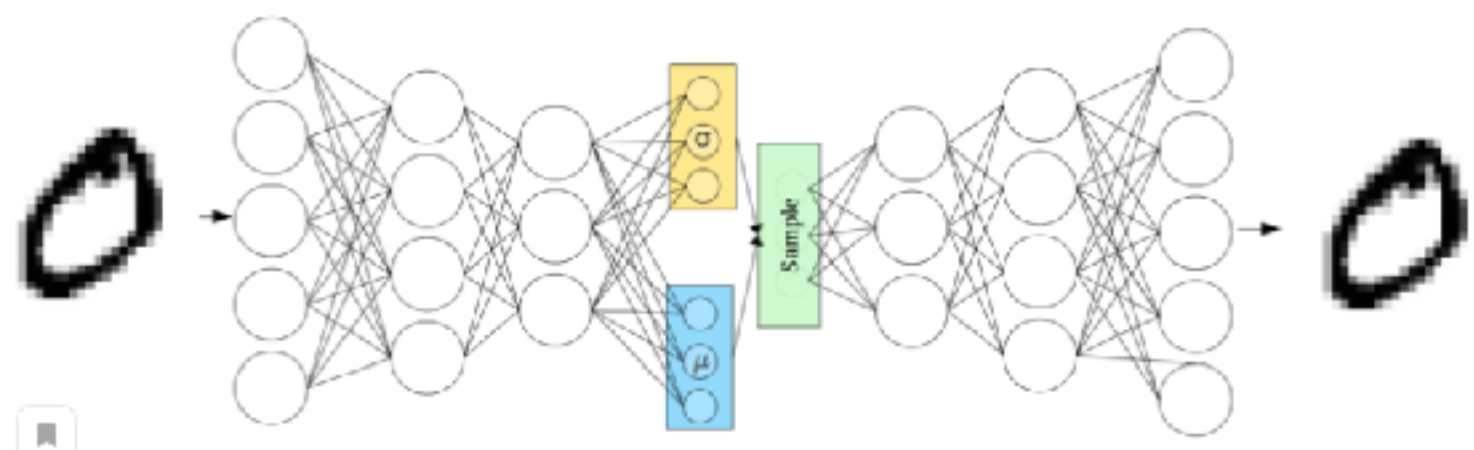



GANs (Goodfellow et al. 2014)




GMM + EM



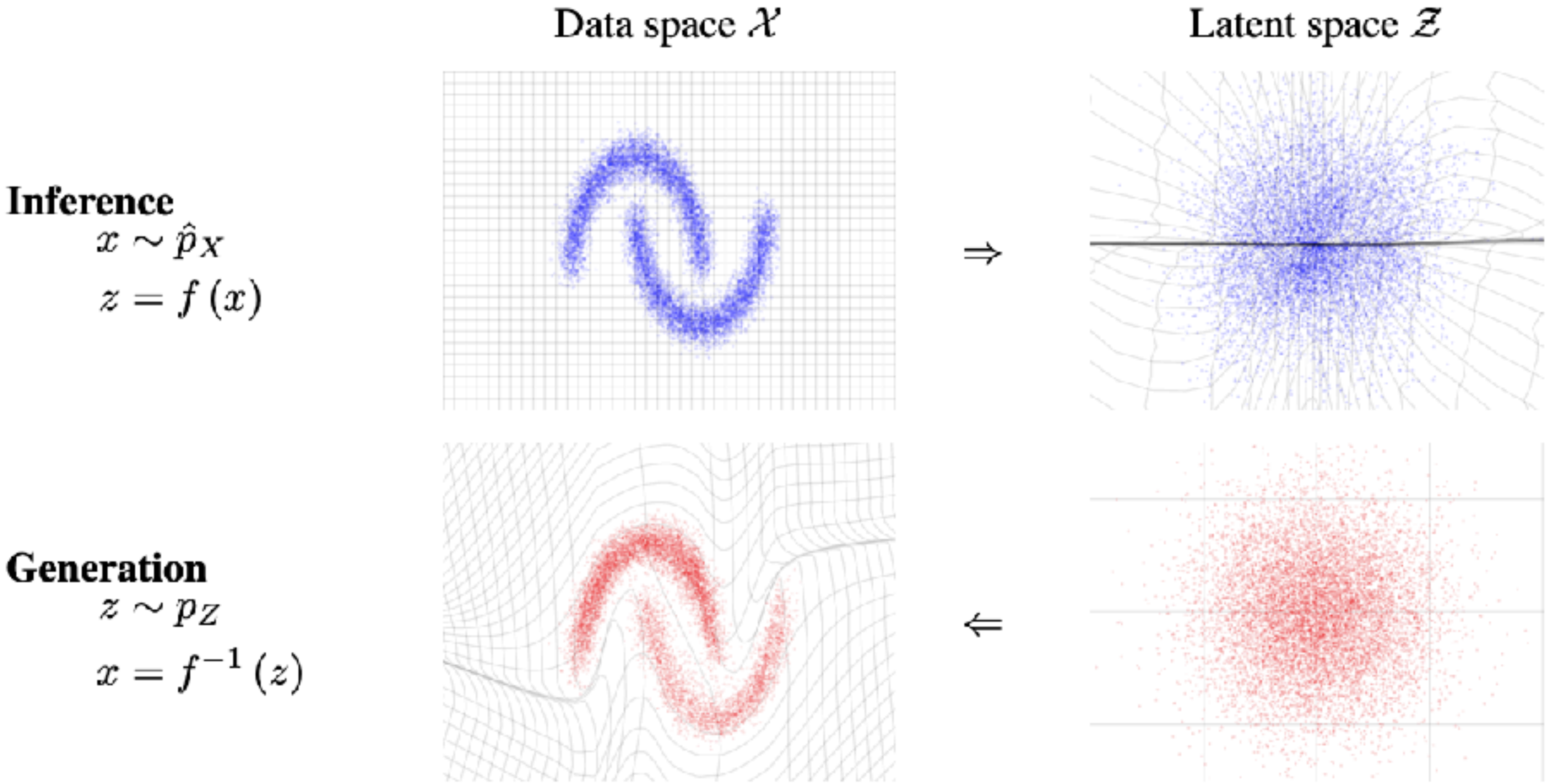




Variational autoencoder (Kingma et al. 2013)

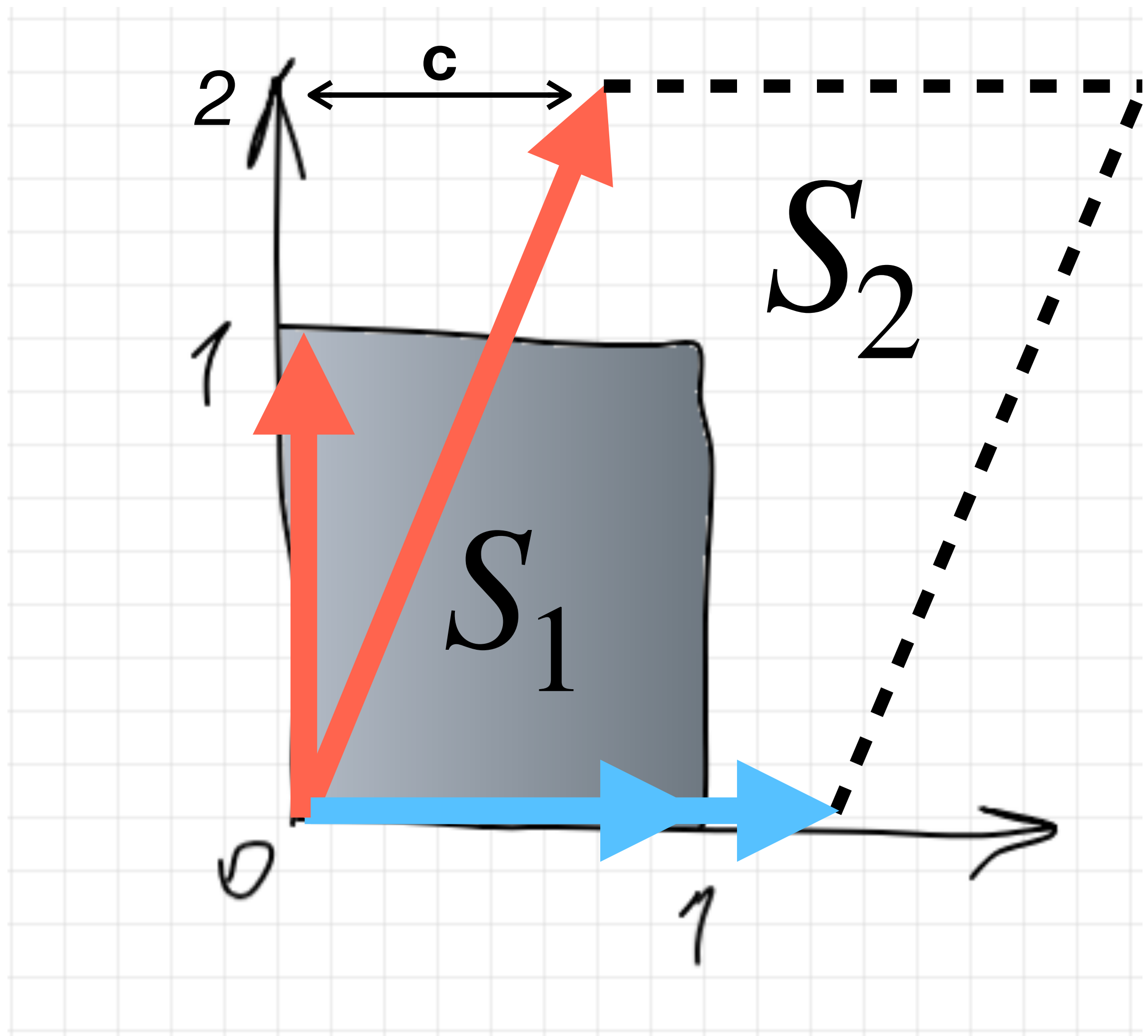
Autoregressive GM (Aaron et al. 2016)

# Real-valued non-volume preserving (real NVP) transformations

# Идея Real NVP



Вспомним, как меняется площадь при смене базиса.




$$e_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$e_1' = \begin{bmatrix} 2 \\ c \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

$$S_1 = 1 \cdot 1 = 1$$

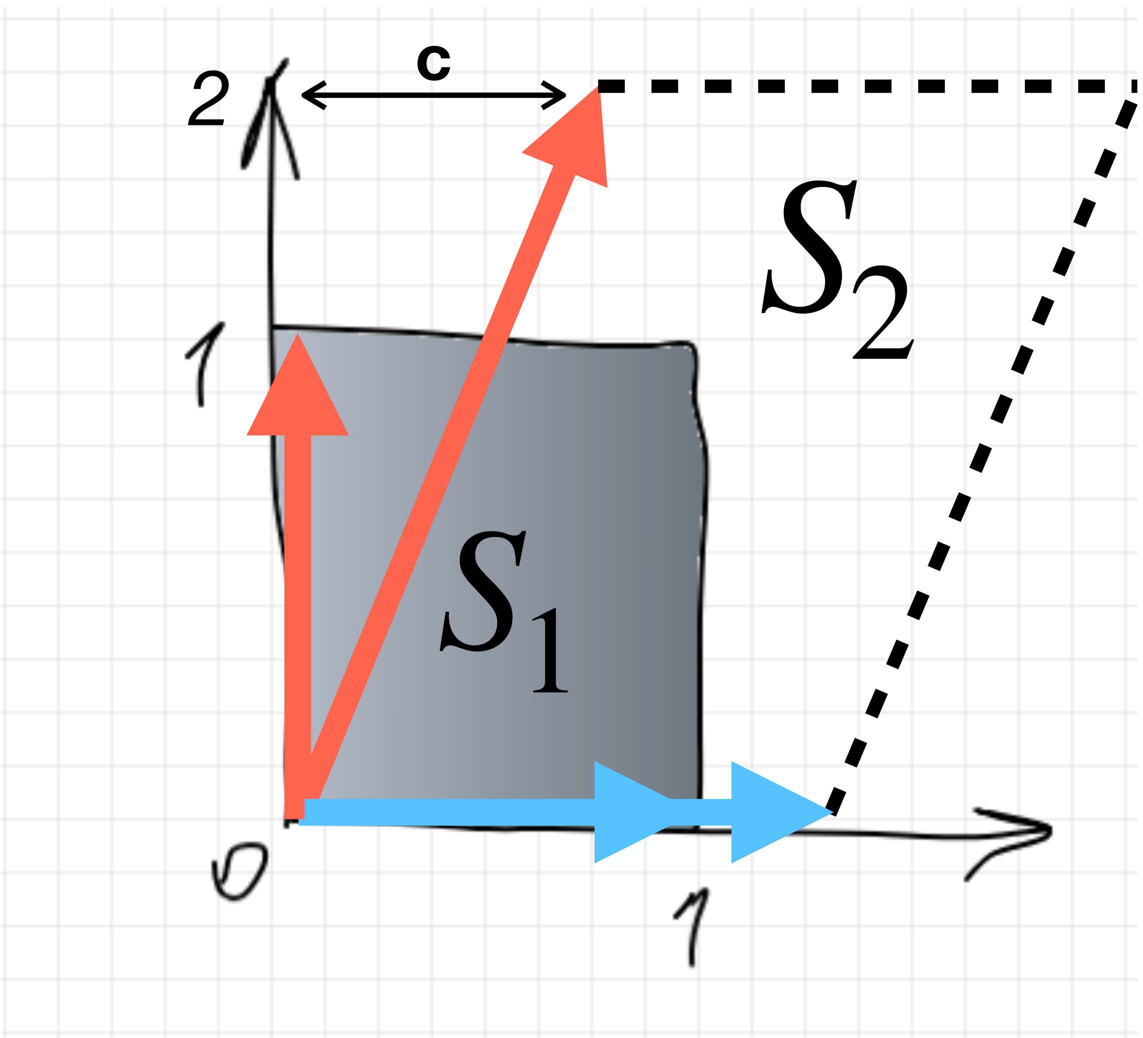
$$S_2 = b \cdot \langle e_1', e_1 \rangle = 2b$$

**не зависит от с**

$$C_{e \to e'} = \begin{bmatrix} 2 & 0 \\ c & b \end{bmatrix} \qquad S_2 = |\, C_{e \to e'} | = 2b$$

**диагональная**

# Замена переменной




$$z \sim U([0,1]^2) \quad y = g(z) = C_{e' \to e} z$$

$$p_z(z) = \frac{1}{S_1} \qquad p_y(y) = ?$$

$$p_y(y) = p_z(z) \left| det \left( \frac{\partial g(z)}{\partial z^T} \right) \right|^{-1}$$

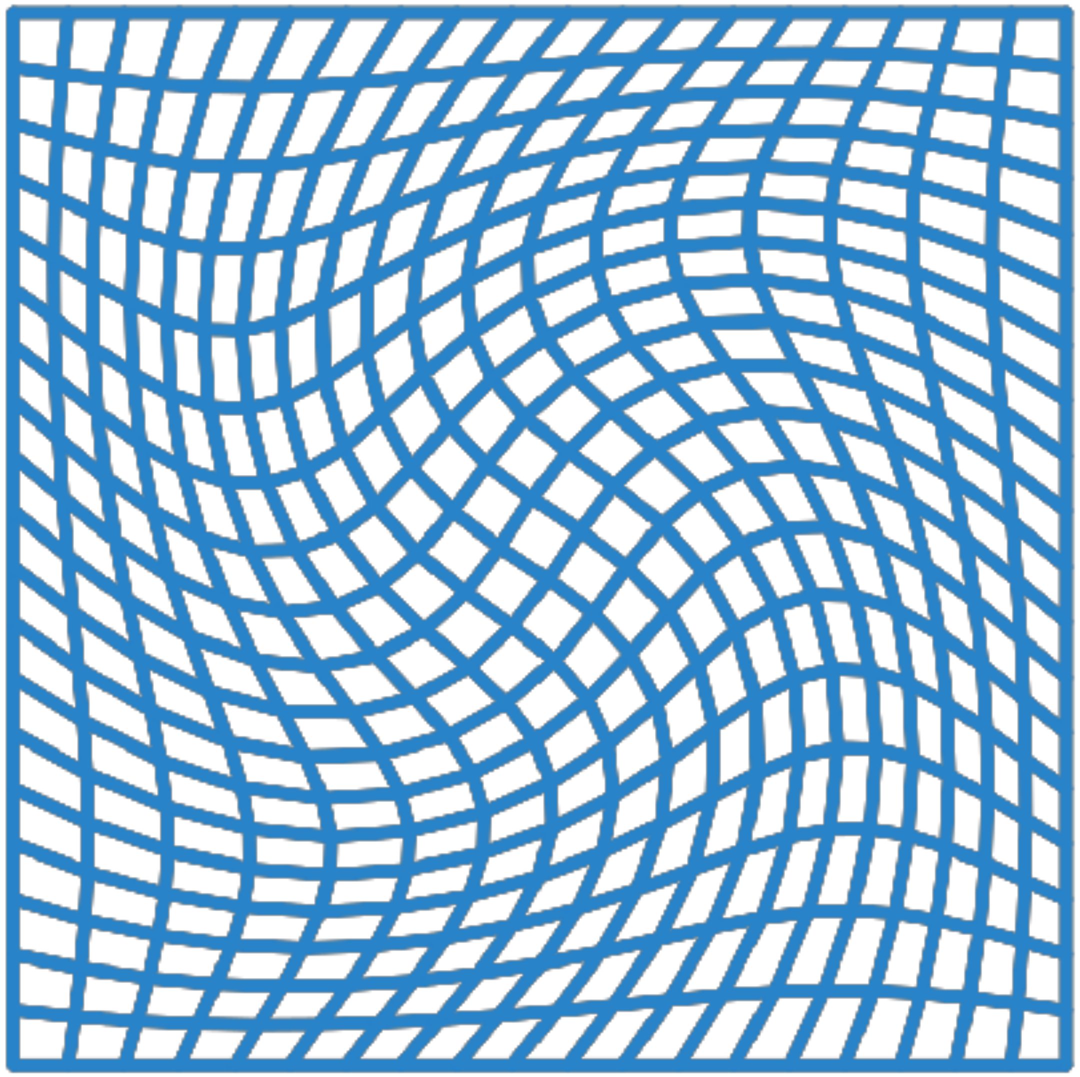
$$C_{e \to e'} = \begin{bmatrix} 2 & 0 \\ c & b \end{bmatrix}$$

**считаем Якобиан за O(n)!**
**вместо O(n^3)**

$$p_y(y) = p_z(z) S_2^{-1}$$

**‘c’ может быть сложной функцией!**

# Замена переменной



**Пример хорошего отображения (диффеоморфизм)**

**При соблюдении условий**

1. Отображение *g* - взаимно-однозначно.

2. g диффререренцируемо.

3. Якобиан отображения $J_g(x) \neq 0$:

**Справедлива теорема**

$$p_y(y) = p_z(z) \left| det\left(\frac{\partial g(z)}{\partial z^T}\right) \right|^{-1}$$

**то, что будем максимизировать (максимизация правдоподобия)**

# Affine coupling layer

$$y_{1:d} = x_{1:d}$$

$$y_{d+1:D} = x_{d+1:D} \odot \exp\big(s(x_{1:d})\big) + t(x_{1:d}),$$

**Как будет выглядеть матрица Якоби такой функции?**

$$\frac{\partial y}{\partial x^T} = \begin{bmatrix} \mathbb{I}_d & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \operatorname{diag}\left(\exp\left[s\left(x_{1:d}\right)\right]\right) \end{bmatrix}$$

**s, t - одни и те же пусть это будет нейросети!**

**Как посчитать обратную функцию?**

$$\begin{cases} x_{1:d} &= y_{1:d} \\ x_{d+1:D} &= \big(y_{d+1:D} - t(y_{1:d})\big) \odot \exp\big(-s(y_{1:d})\big) \end{cases}$$

# Правило композиции

$$\frac{\partial (f_b \circ f_a)}{\partial x_a^T}(x_a) = \frac{\partial f_a}{\partial x_a^T}(x_a) \cdot \frac{\partial f_b}{\partial x_b^T}\left(x_b = f_a(x_a)\right)$$

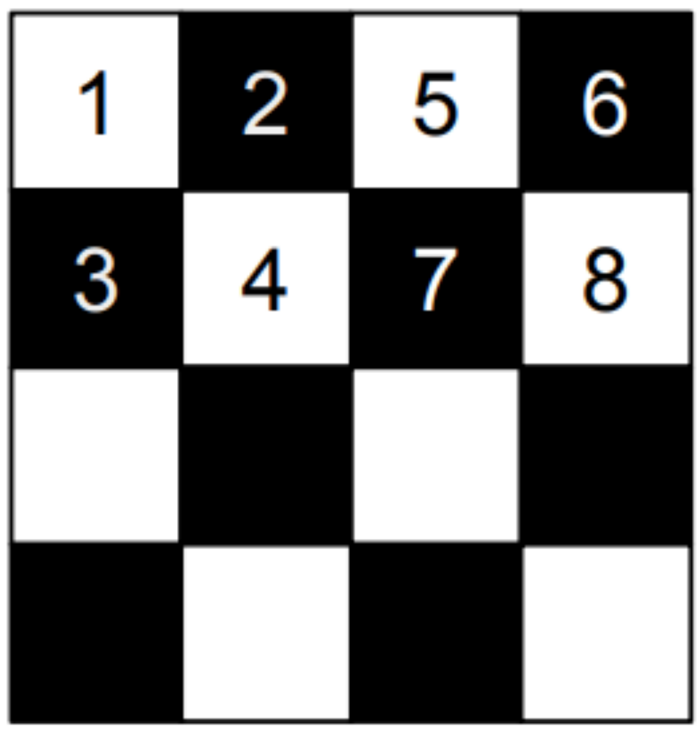
$$\det(A \cdot B) = \det(A)\det(B).$$

$$(f_b \circ f_a)^{-1} = f_a^{-1} \circ f_b^{-1}.$$

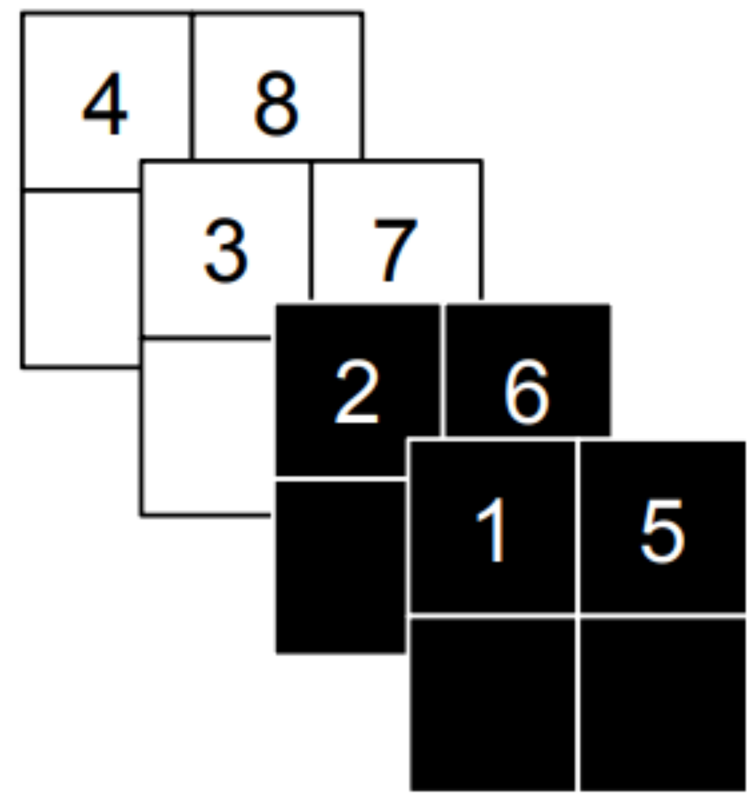
**Позволяет нам комбинировать простые преобразования в сложные композиции.**

# Применение к изображениям

**Маскирование для affine coupling layer**

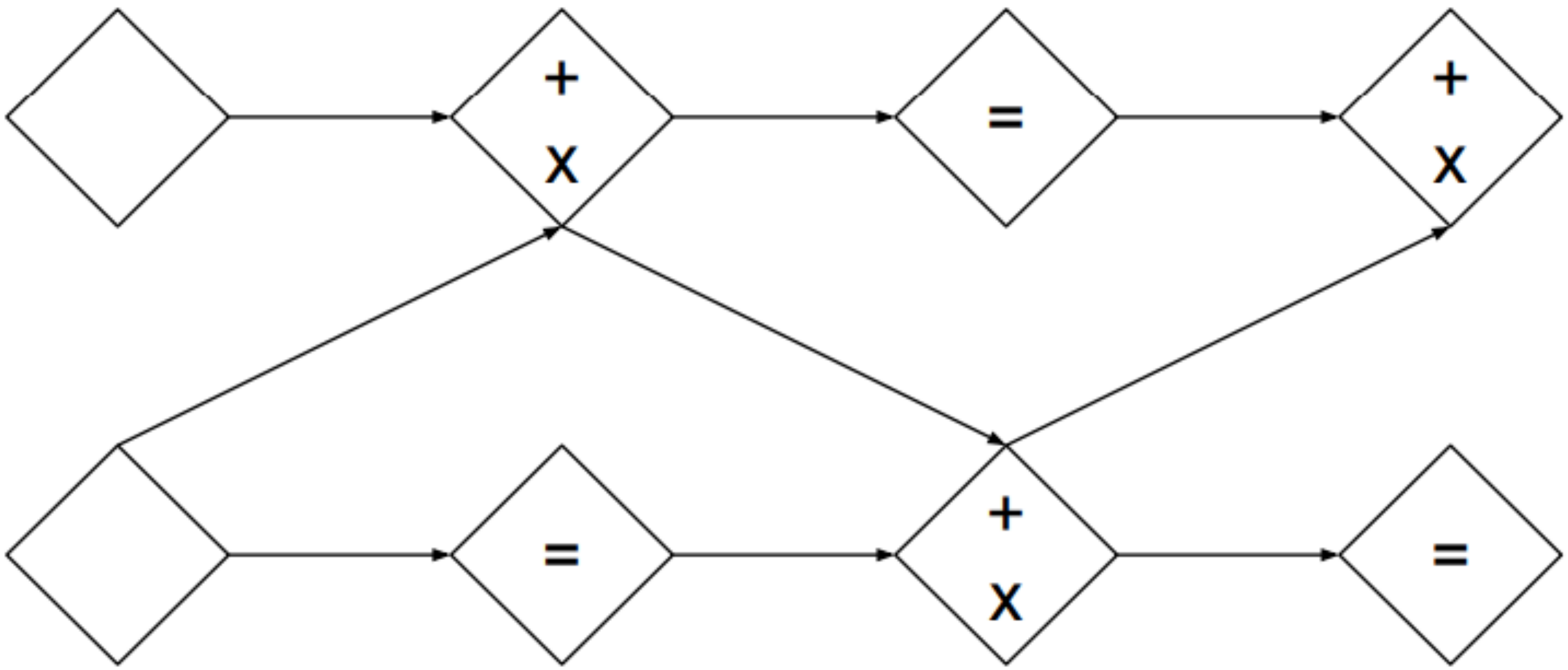



**шахматное маскирование**



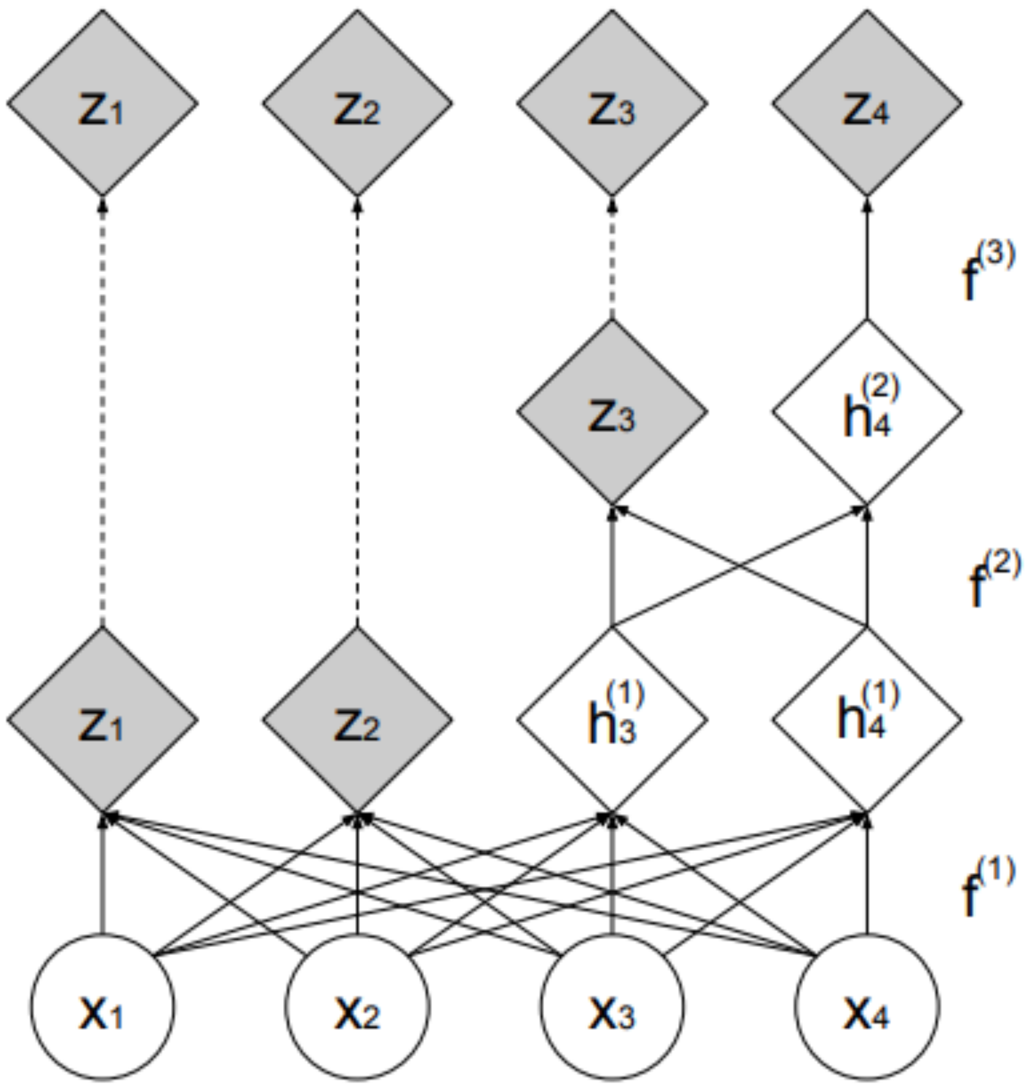

**маскирование по каналам**

**Клетками одного цвета обозначены x[1:d], а другого x[d+1:D]**




**фиксируем одни переменные, обновляем другие**

# Ранняя гауссинизация




**быстрее**

**эффективно по памяти**

**половина переменных на каждом этапе фиксируется**
**(считаем, что это уже гаусовские латентные переменные)**

# Ключевые преимущества

- **Скорость. В отличие от Pixel RNN.**
- **Вычислимое правдоподобие. В отличие от GAN.**

# Модель для больших датасетов

- s, t - сверхточные резидьюальные сети
- batch normalisation
- weight normalisation

# Batch Normalisation

- считаем моменты $\sigma^2$, $\mu$

$$x \to \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \qquad J = (\sqrt{\sigma^2 + \epsilon})^{-1}$$

Аппроксимация

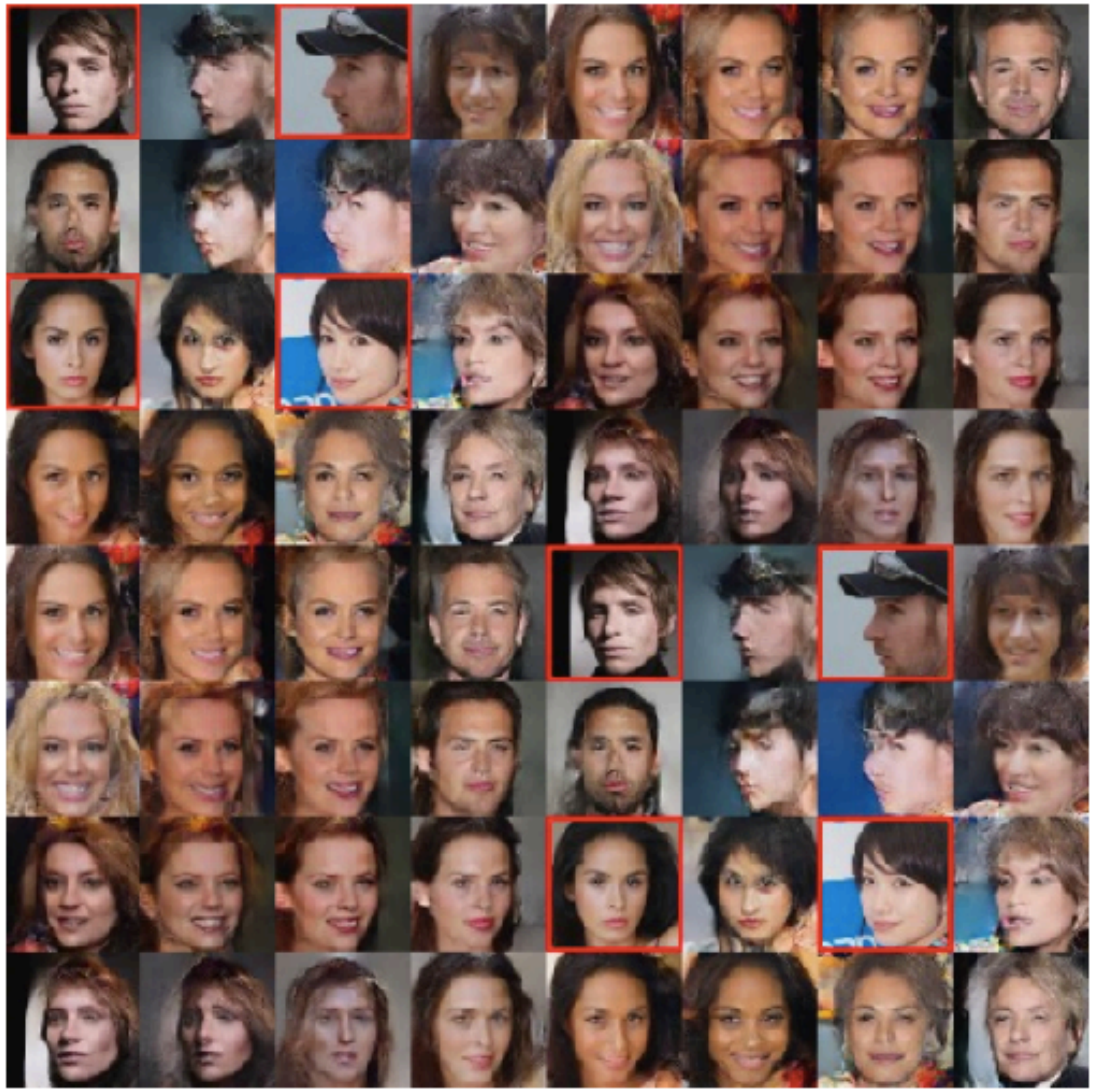
# Результаты

| Dataset | PixelRNN [46] | Real NVP | Conv DRAW [22] | IAF-VAE [34] |
|---|---|---|---|---|
| **CIFAR-10** | 3.00 | 3.49 | < 3.59 | < 3.28 |
| **Imagenet** $(32 \times 32)$ | 3.86 (3.83) | 4.28 (4.26) | < 4.40 (4.35) | |
| **Imagenet** $(64 \times 64)$ | 3.63 (3.57) | 3.98 (3.75) | < 4.10 (4.04) | |
| **LSUN (bedroom)** | | 2.72 (2.70) | | |
| **LSUN (tower)** | | 2.81 (2.78) | | |
| **LSUN (church outdoor)** | | 3.08 (2.94) | | |
| **CelebA** | | 3.02 (2.97) | | |

**Bits/dim (чем меньше, тем лучше)**

# Примеры генерации изображений.

# Интер/Экстра-поляция



На рисунке представлены изображения,
полученные путем интерполяции на сфере в пространстве латентных переменных.

# Summary

- Вычислимое оценивание плотности с помощью DNN и формулы замены переменной

- Быстрый вывод и сэмплирование

- качество на уровне других моделей и связные изображения

# Материалы

- Оригинальная статья. https://arxiv.org/pdf/1605.08803.pdf

- Презентация автора. https://www.periscope.tv/hugo_larochelle/1ypKdAVmbEpGW