# Прикладные задачи анализа данных

Лекция 2

Нормализационные потоки

Михаил Гущин

mhushchyn@hse.ru

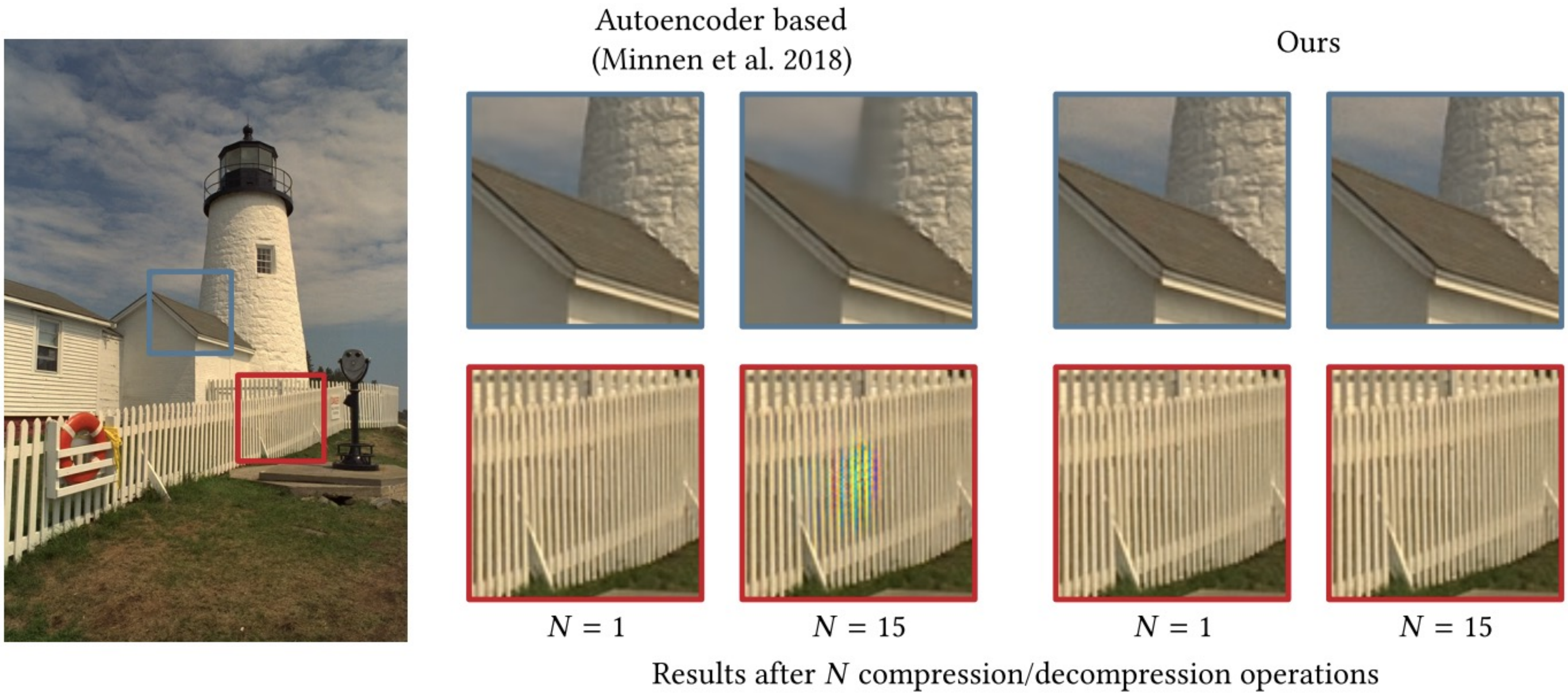
НИУ ВШЭ, 2023

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

# Приложения

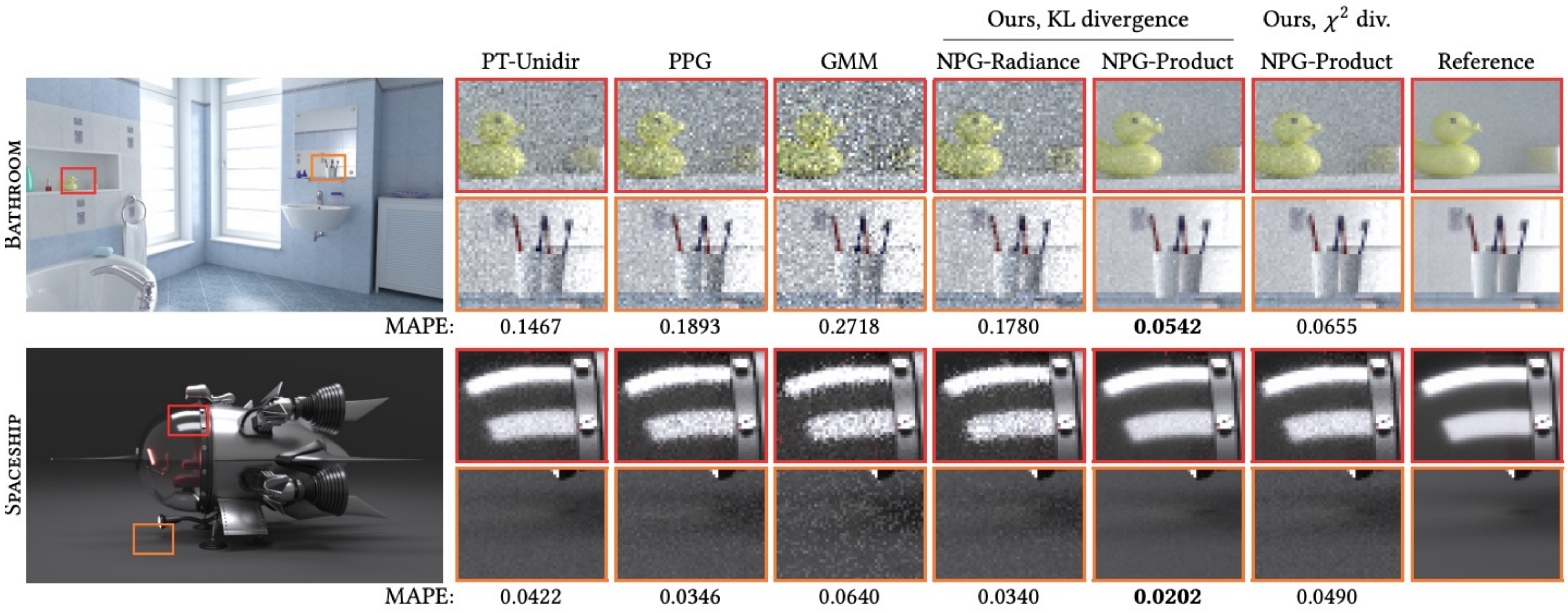
# Пример

# Пример




**Lossy Image Compression with Normalizing Flows, ICLR 2021**

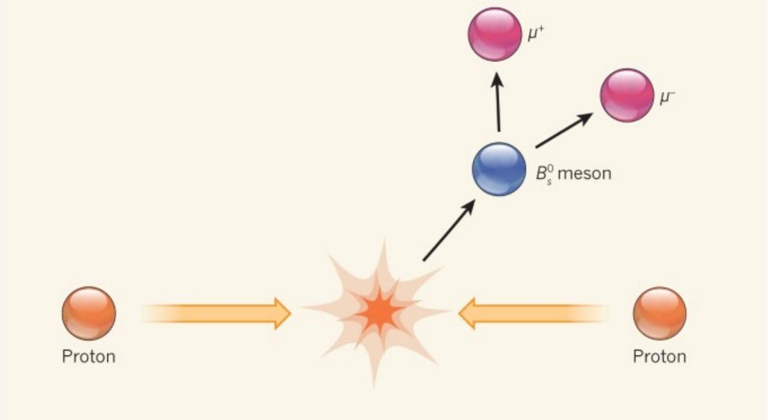
# Пример

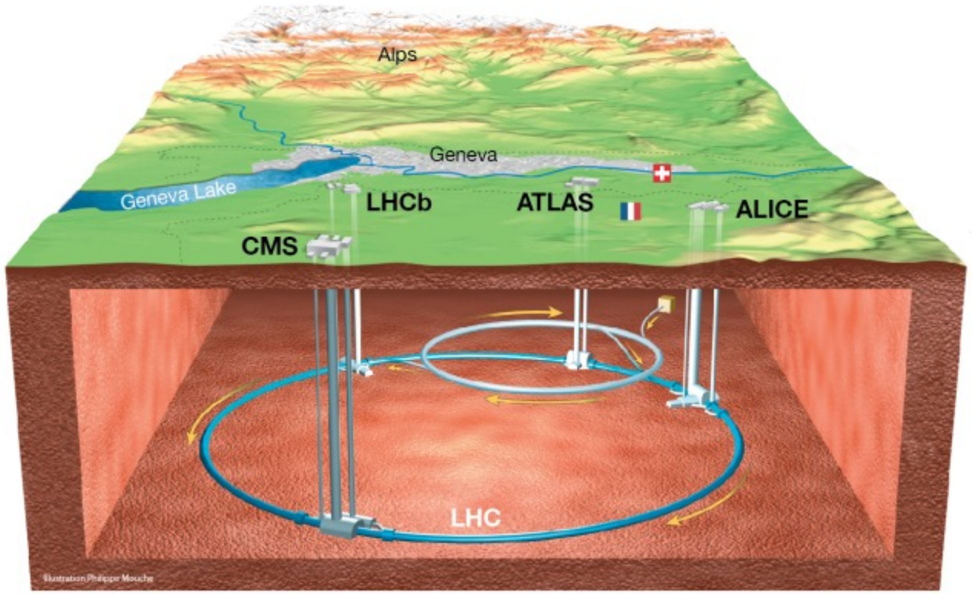


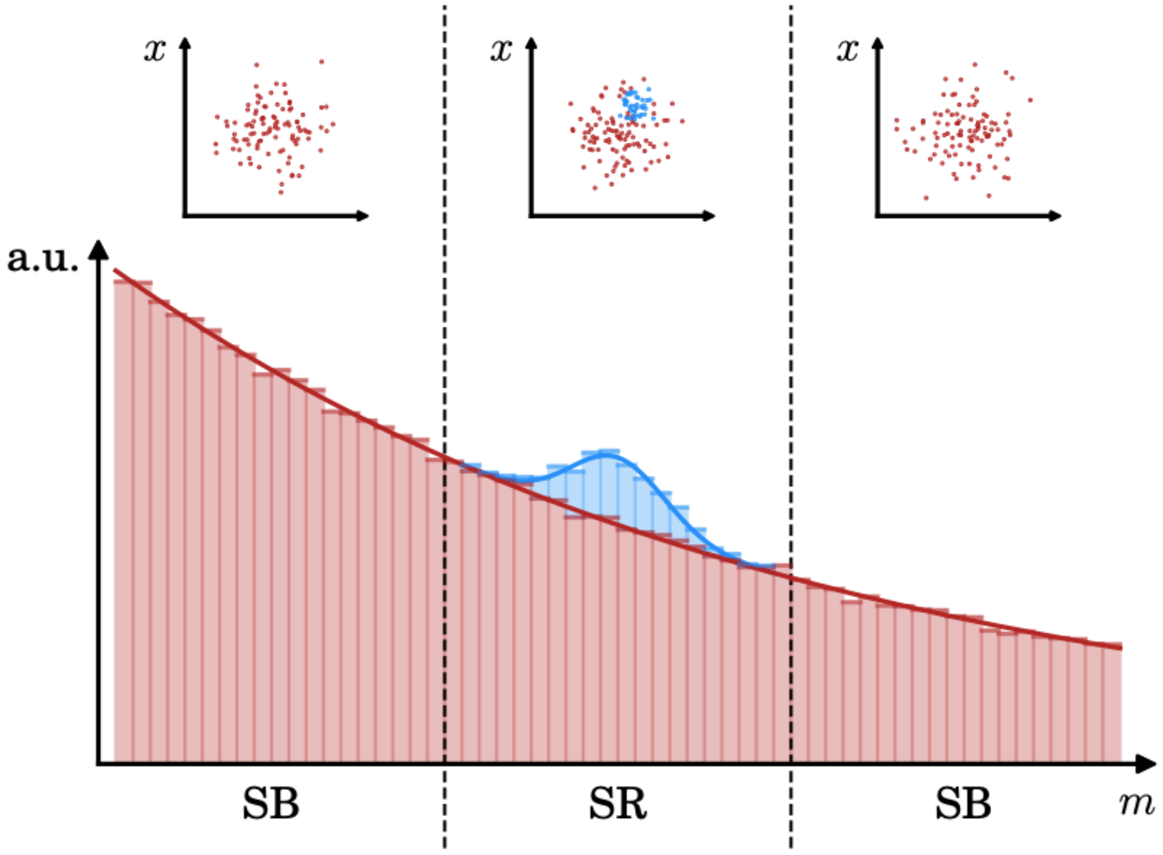

**Neural Importance Sampling**

# Поиск новой физики

*Nature* **522,** 42–43 (2015)



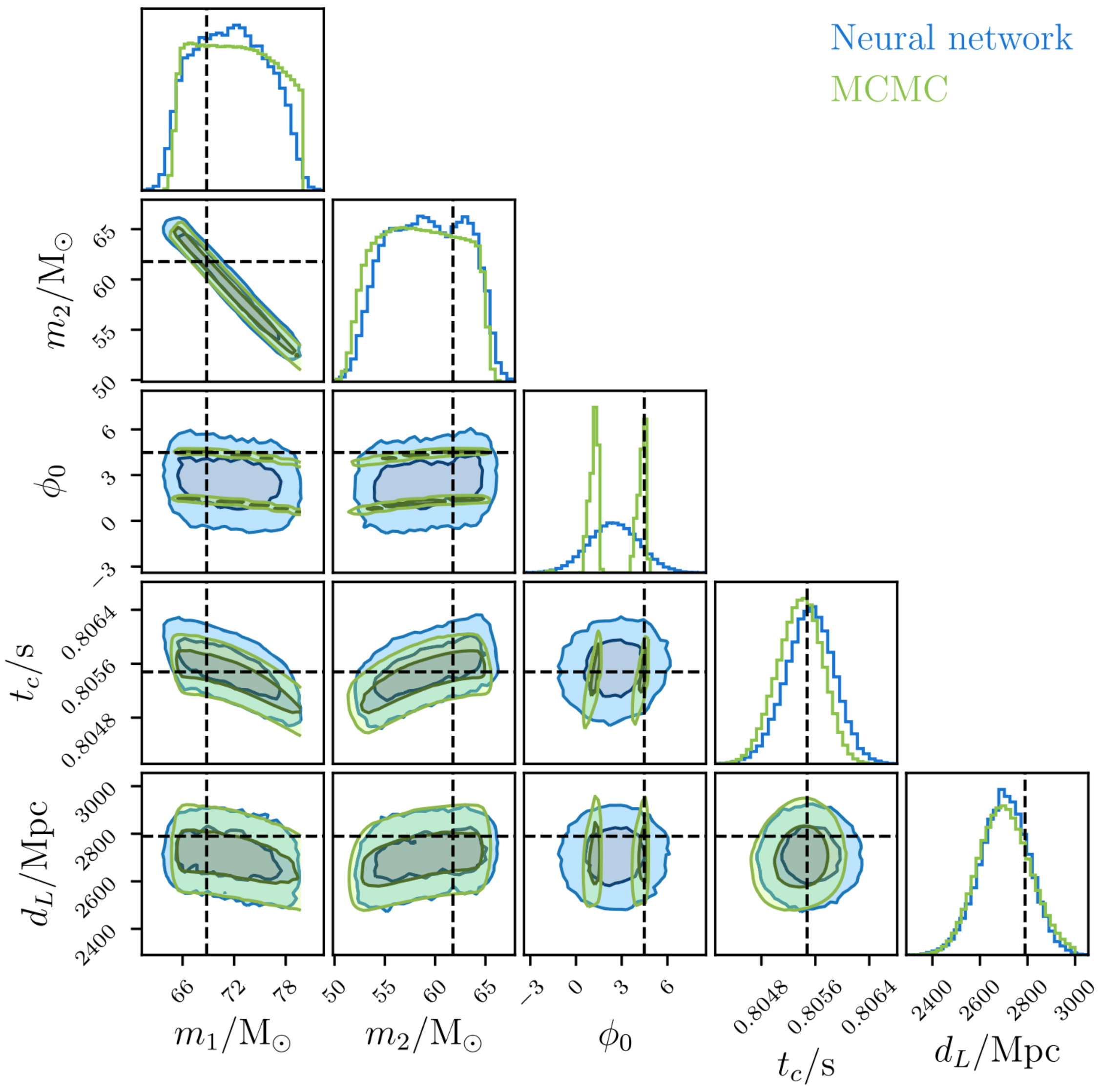

$$p_{\mathrm{data}}(x|m \in SB) = p_{\mathrm{bg}}(x|m \in SB)$$

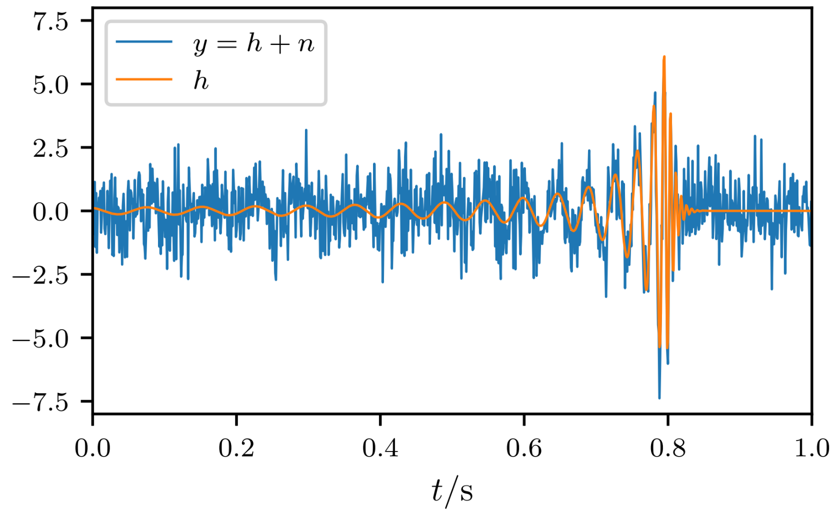
$$p_{\mathrm{data}}(x|m \in SR)$$

$$p_{\mathrm{data}}(x|m \in SB) = p_{\mathrm{bg}}(x|m \in SB)$$

FERMILAB-PUB-21-389-T

# Гравитационные волны

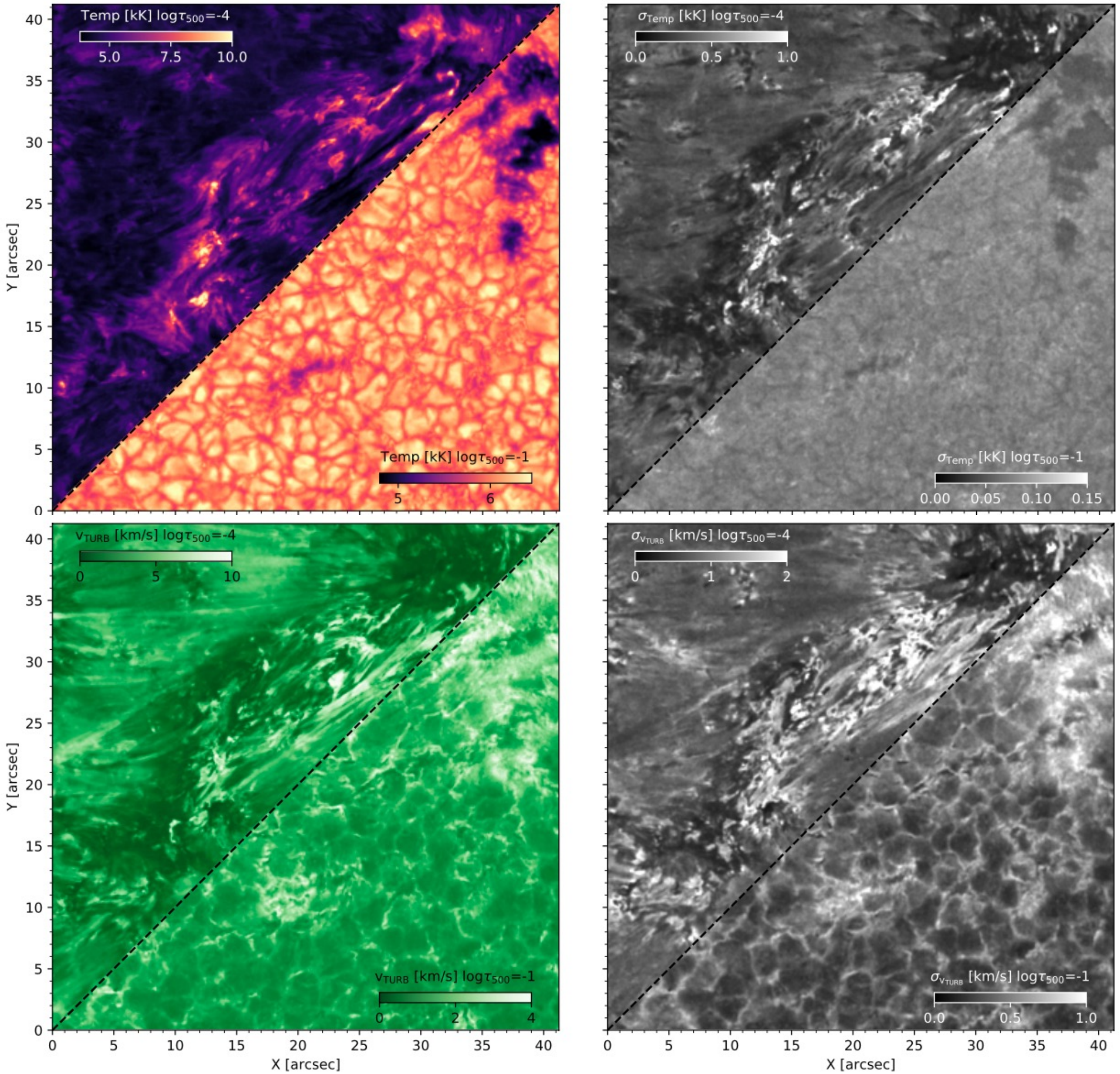



Gravitational-wave parameter estimation with autoregressive neural network flows, Phys. Rev. D **102**, 104057

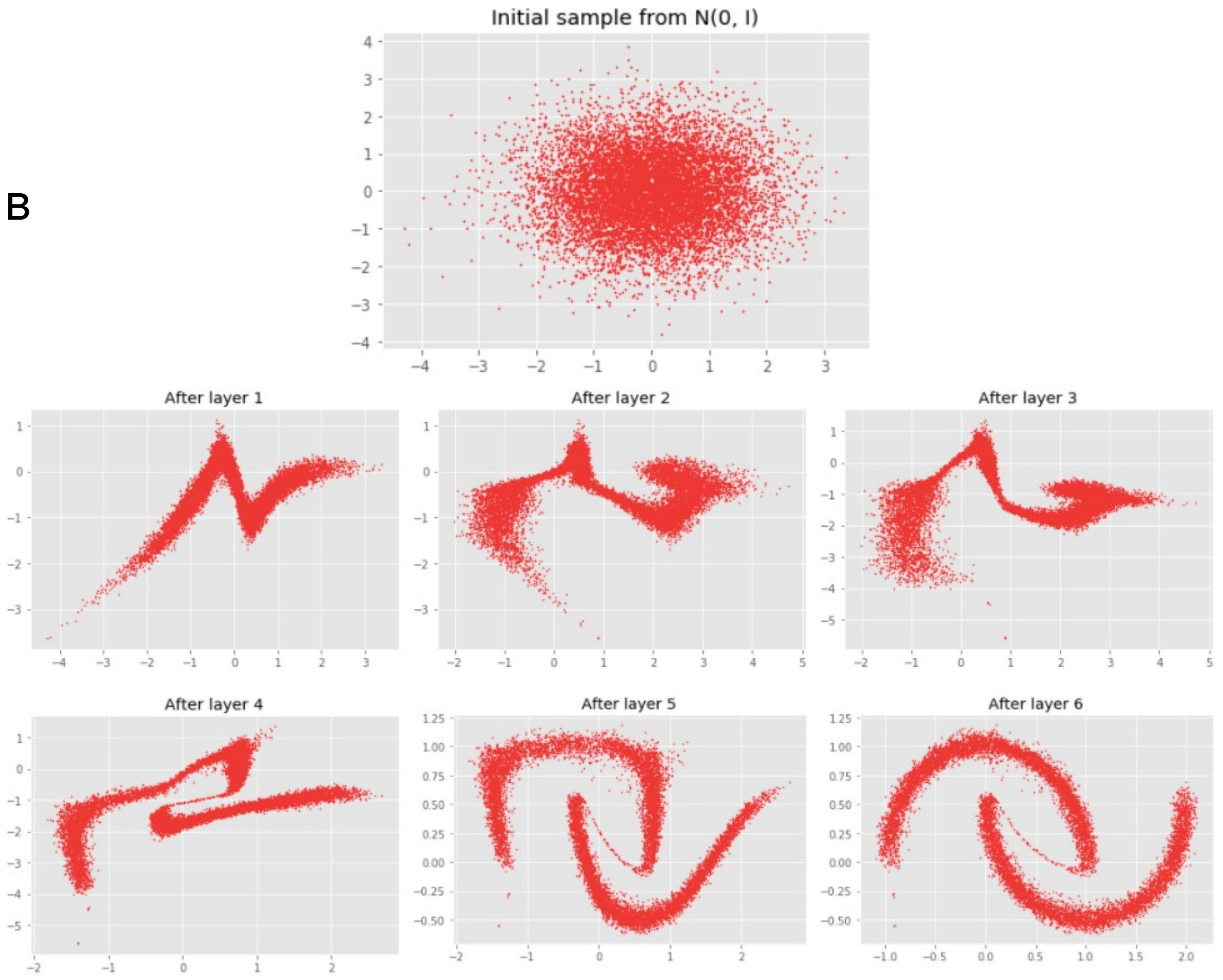
# Реконструкция солнечной атмосферы

Bayesian Stokes inversion
with normalizing flows,
A&A 659, A165 (2022)



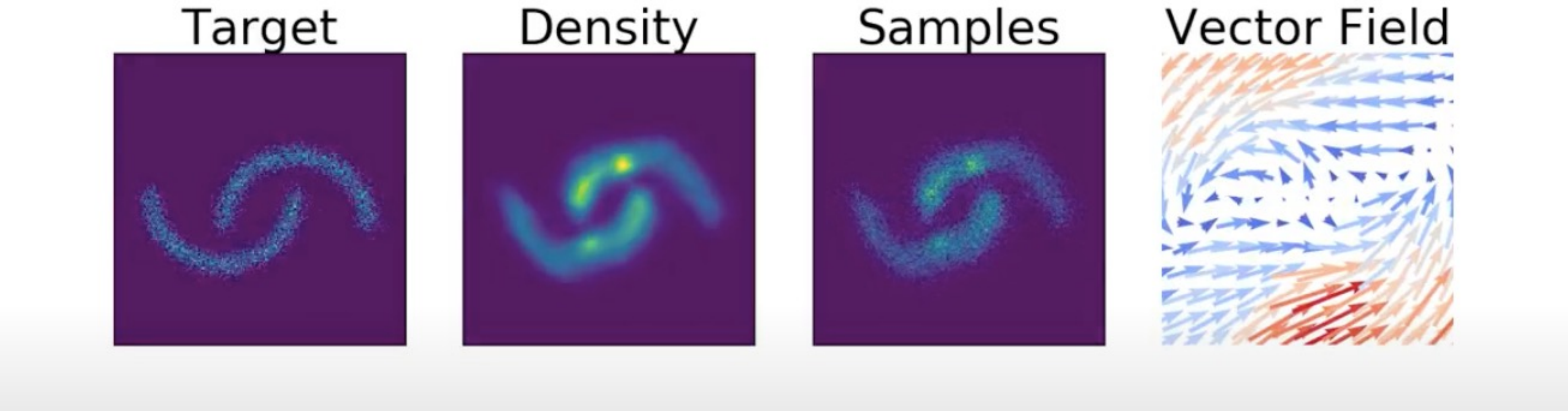
# Интуиция

Нормализационный поток – **последовательность** **обратимых** преобразований одного распределения в другое




https://engineering.papercup.com/posts/normalizing-flows-part-2/

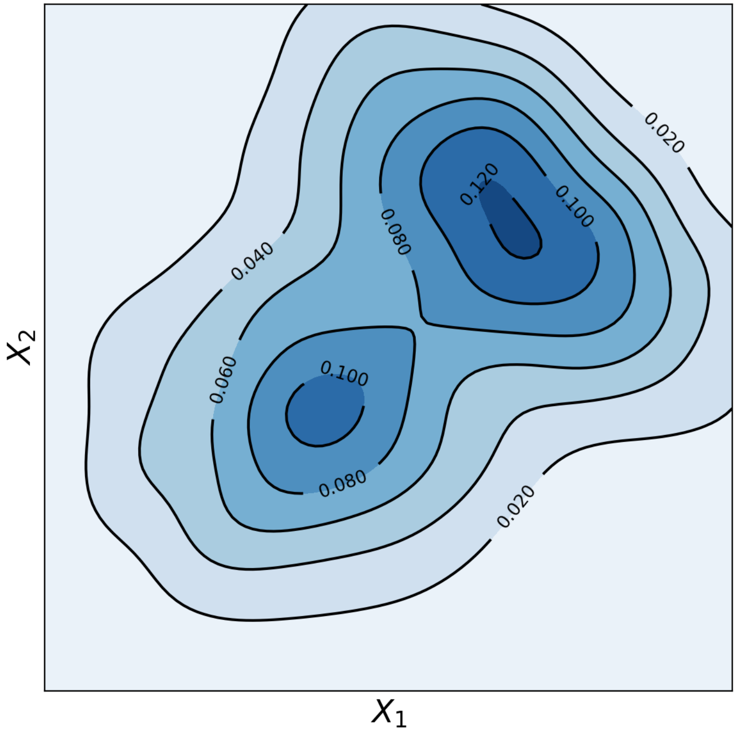
# Интуиция



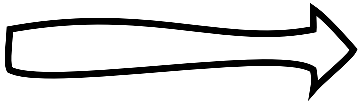

https://youtu.be/QQpz7V3OgFc

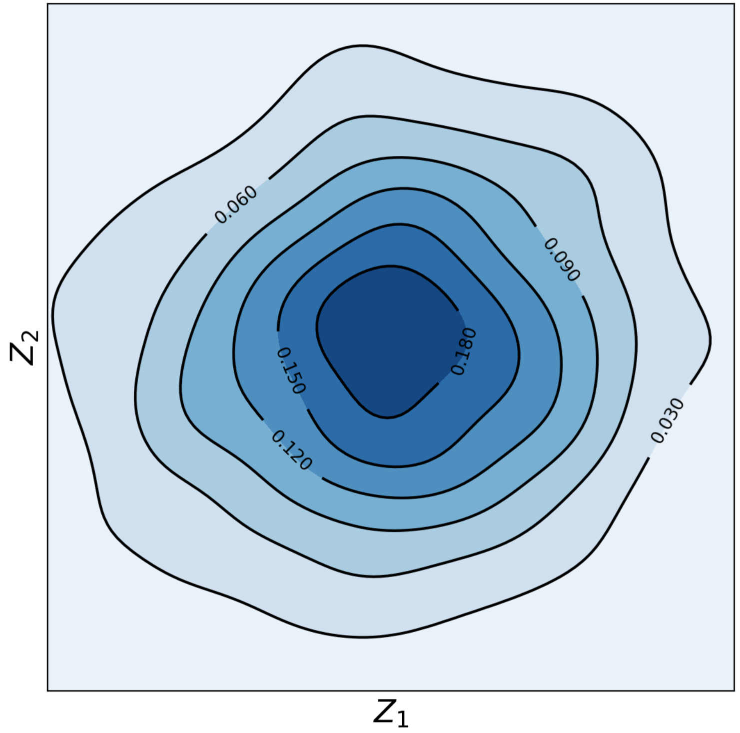
# Теорема о замене переменных

# Замена переменных




$$z = f(x)$$






$x_i \sim p_x(x)$

$p_x(x)$ - ?

$z_i \sim p_z(z)$

$p_z(z)$ - известно

# Теорема о замене переменных

Пусть даны $\boldsymbol{p_z}(z)$ и z $= \boldsymbol{f}(x)$, тогда $p_x(x)$ находим так:

Отношение объема $\partial z$ к новому объему $\partial x$

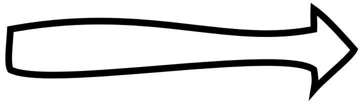
$$p_x(x_i) = \boldsymbol{p_z}\left(\boldsymbol{f}(x_i)\right)\left|\det\frac{\partial \boldsymbol{f}(x_i)}{\partial x_i}\right|,$$

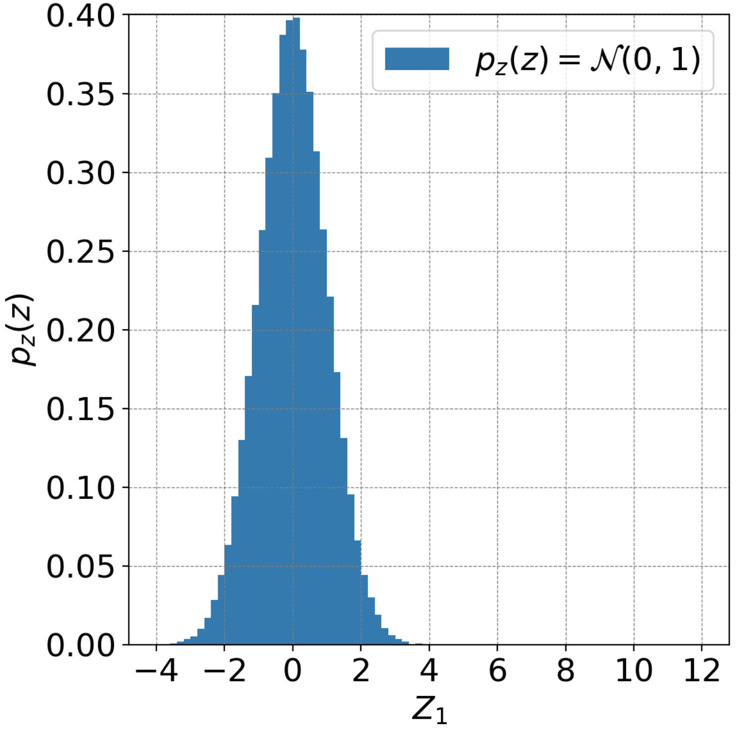
где матрица первых производных определяется так:

$$\frac{\partial \boldsymbol{f}(x_i)}{\partial x_i} = \begin{pmatrix} \frac{\partial \boldsymbol{f}(x_i)_1}{\partial x_{i1}} & \cdots & \frac{\partial \boldsymbol{f}(x_i)_1}{\partial x_{in}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{f}(x_i)_m}{\partial x_{i1}} & \cdots & \frac{\partial \boldsymbol{f}(x_i)_m}{\partial x_{in}} \end{pmatrix}.$$

# Пример 1

$$z = \boldsymbol{f}(x) = 0.5x - 2.5$$






$$x_i \sim p_x(x)$$

$$p_x(x) \text{ - ?}$$

$$z_i \sim p_z(z)$$

$$p_z(z) = \mathcal{N}(0, 1)$$

# Пример 1

Итак, дана функция $\boldsymbol{f}(x)$:

$$z = \boldsymbol{f}(x) = 0.5x - 2.5$$

Тогда, матрица первых производных:

$$\frac{\partial \boldsymbol{f}(x_i)}{\partial x_i} = (0.5)$$

И значение Якобиана:

$$\left| \det \frac{\partial \boldsymbol{f}(x_i)}{\partial x_i} \right| = 0.5$$

# Пример 1

Формула замены пременных:

$$p_x(x_i) = \boldsymbol{p_z}\big(\boldsymbol{f}(x_i)\big)\left|\det\frac{\partial \boldsymbol{f}(x_i)}{\partial x_i}\right|,$$

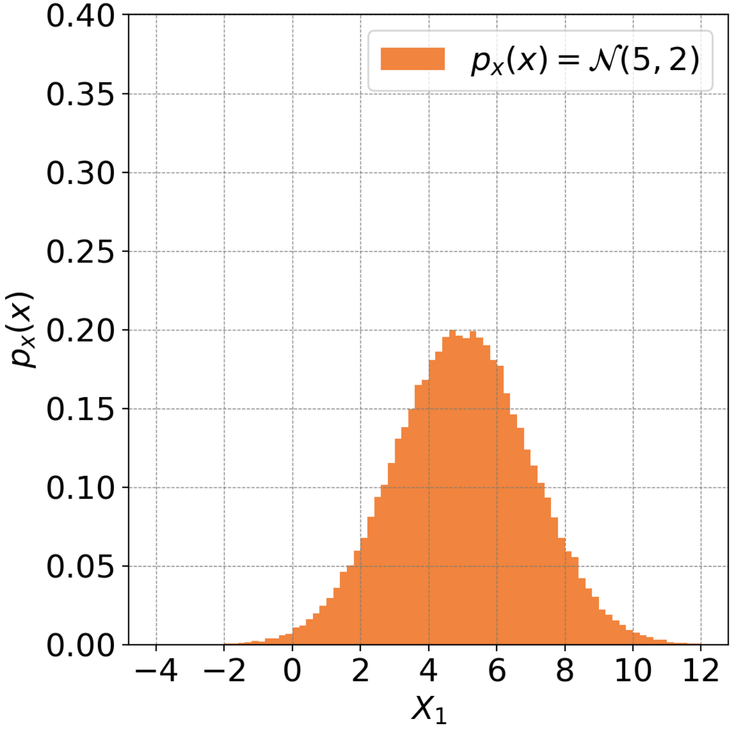
Подставим известные выражения:

$$\boldsymbol{p_z}(z_i) = \frac{1}{\sqrt{2\pi}}e^{-\frac{(z_i)^2}{2}}$$

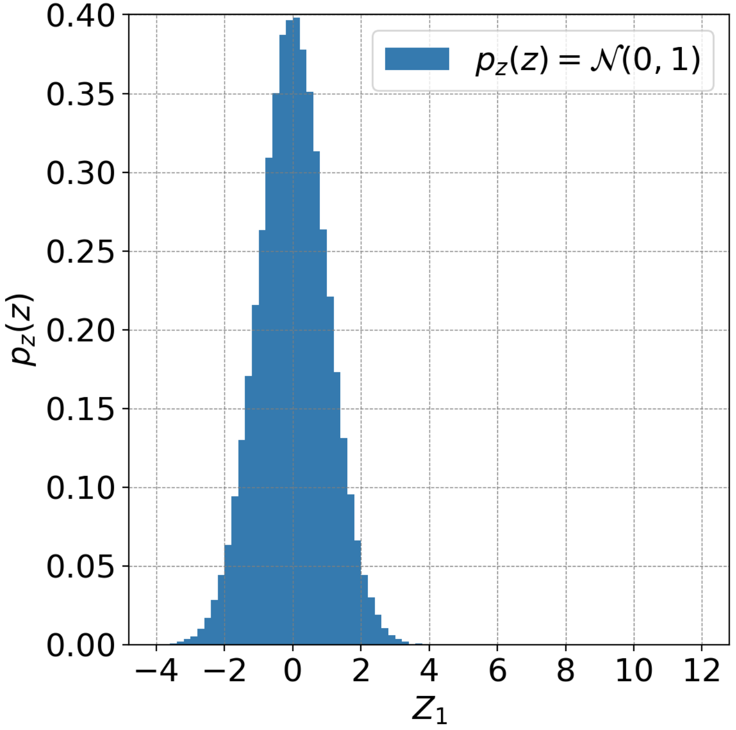
$$p_x(x_i) = \frac{1}{\sqrt{2\pi}}e^{-\frac{(0.5x_i-2.5)^2}{2}} * 0.5$$

$$= \frac{1}{2\sqrt{2\pi}}e^{-\frac{(x_i-5)^2}{2*2^2}} = \mathcal{N}(5,2)$$

# Пример 1




$$z = f(x) = 0.5x - 2.5$$




$$x_i \sim p_x(x)$$

$$p_x(x) \text{ - ?}$$

$$z_i \sim p_z(z)$$

$$p_z(z) = \mathcal{N}(0, 1)$$

# Теорема о замене переменных

Пусть даны $\boldsymbol{p_z}(z)$ и z = $\boldsymbol{f}(x)$, тогда $p_x(x)$ находим так:

Отношение объема $\partial z$ к новому объему$\partial x$

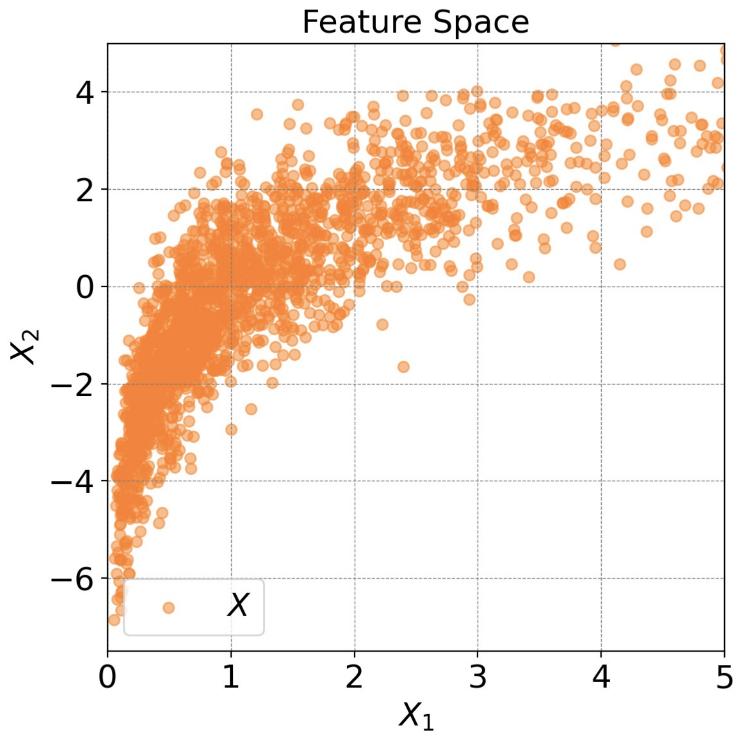
$$p_x(x_i) = \boldsymbol{p_z}(\boldsymbol{f}(x_i)) \left|\det \frac{\partial \boldsymbol{f}(x_i)}{\partial x_i}\right|,$$

Обратная замена переменных:

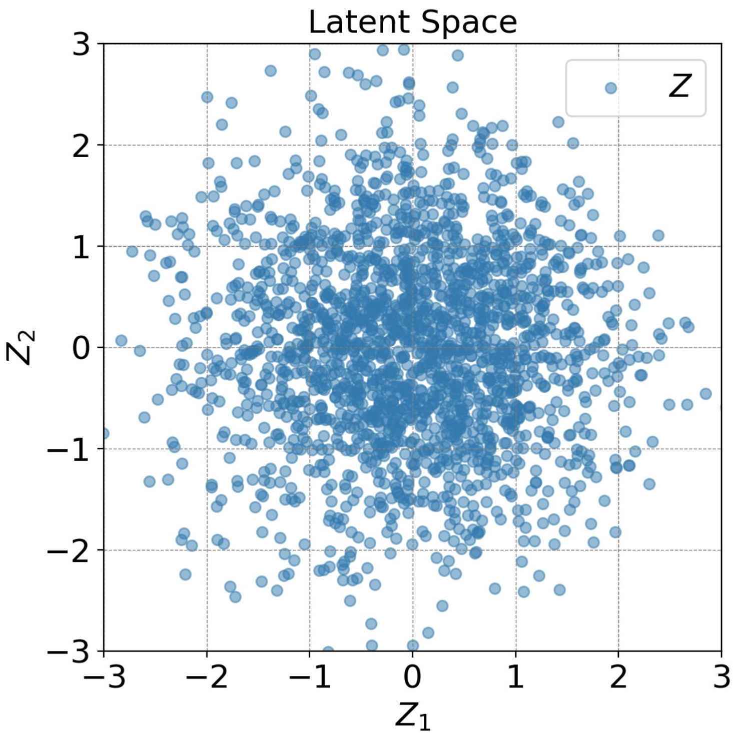
$$\boldsymbol{p_z}(z_i) = p_x\left(\boldsymbol{f}^{-1}(z_i)\right) \left|\det \frac{\partial \boldsymbol{f}^{-1}(z_i)}{\partial z_i}\right|$$

# Нормализационные потоки

# Постановка задачи




$$z = f(x) - ?$$




$$x_i \sim p_x(x)$$

$$p_x(x) - ?$$

$$z_i \sim p_z(z)$$

$p_z(z)$ - известно

# Постановка задачи

- **Дано:**
  - матрица реальных объектов $X$
- **Задача:**
  - найти такую $z_i = f(x_i)$, чтобы $z_i \sim p_z(z)$
  - при этом, $p_z(z)$ известно и задано

# Решение

- Как будем находить $z_i = \boldsymbol{f}(x_i)$?
- **Ответ:** методом градиентного спуска!

- Какую функцию потерь будем оптимизировать?
- **Ответ:** логарифм правдоподобия:

$$L = -\frac{1}{n}\sum_{i=1}^{n} \log p_x(x_i)$$

# Функция потерь

Функция потерь:

$$L = -\frac{1}{n}\sum_{i=1}^{n}\log p_x(x_i)$$

Замена переменных:

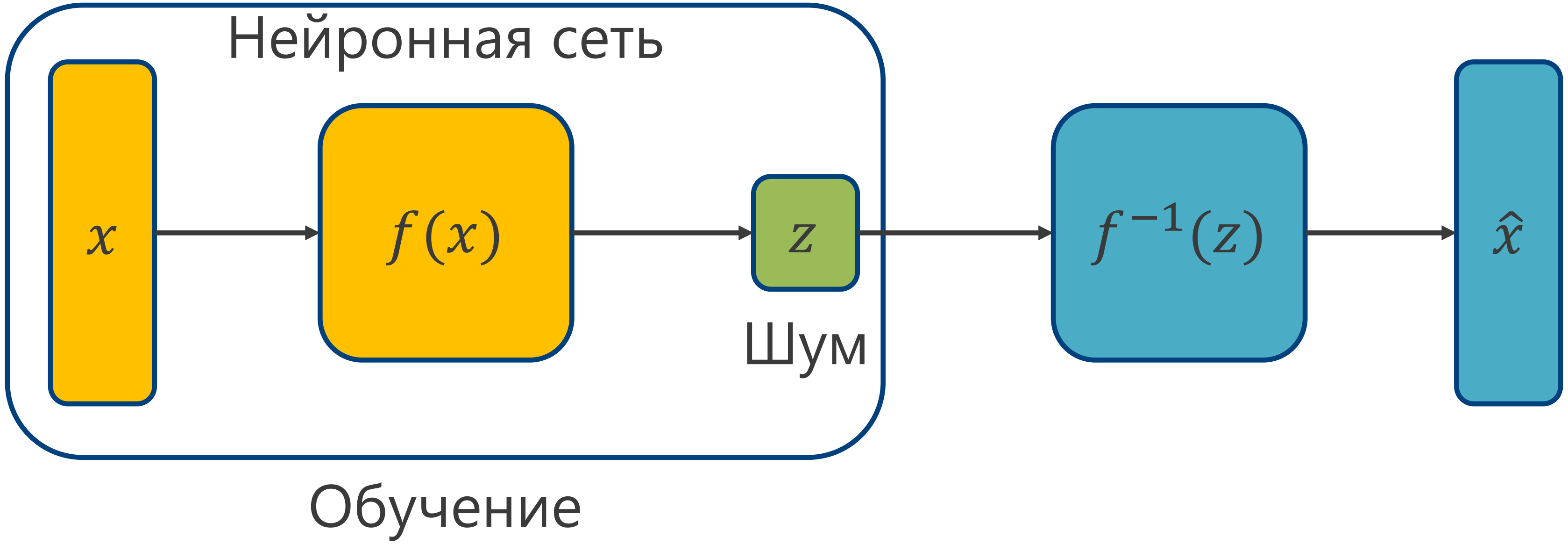
$$p_x(x_i) = p_z\big(f(x_i)\big)\left|det\frac{\partial f(x_i)}{\partial x_i}\right|$$

Подставим в функцию потерь:

$$L = -\frac{1}{n}\sum_{i=1}^{n}\left(log\, p_z\big(f(x_i)\big) + log\left|det\frac{\partial f(x_i)}{\partial x_i}\right|\right)$$

# Общий алгоритм

# Алгоритм обучения



# Алгоритм обучения

Цикл обучения:

- Берем $m$ реальных объектов $\{x_1, x_2, \dots, x_m\}$
- Считаем функцию потерь:

$$L = -\frac{1}{m}\sum_{i=1}^{m}\left(\log p_z\big(f(x_i)\big) + \log\left|\det\frac{\partial f(x_i)}{\partial x_i}\right|\right)$$

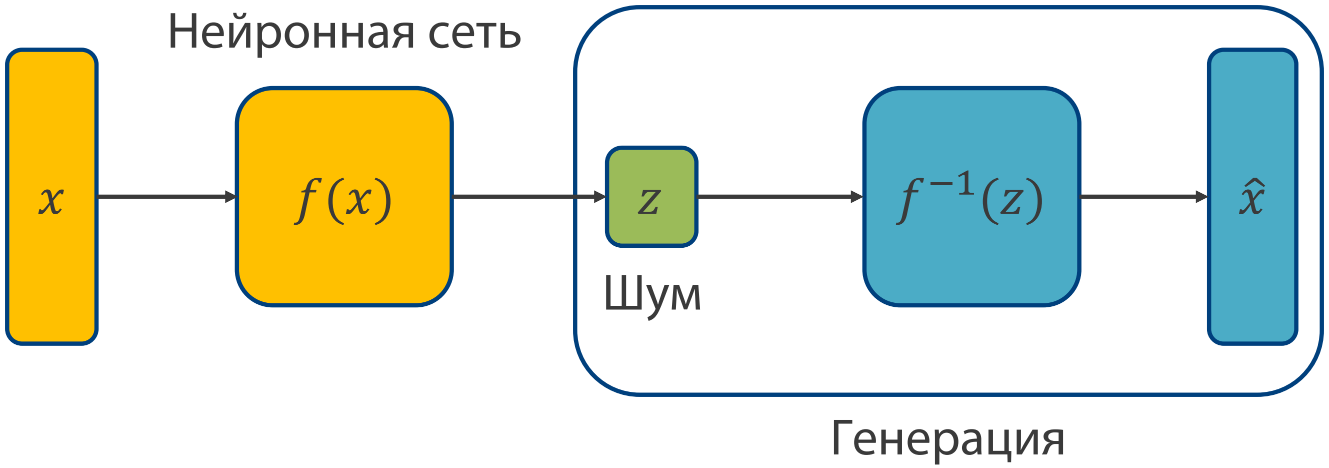
- Обновляем параметры $\theta_f$ функции $z_i = f(x_i)$:

$$\theta_f = \theta_f - \nabla_{\theta_f} L$$

# Алгоритм генерации

- Генерируем случайный шум $\{z_1, z_2, \dots, z_m\}$
- Генерируем новые объекты $\{x_1, x_2, \dots, x_m\}$ по формуле:

$$x_i = f^{-1}(z_i)$$



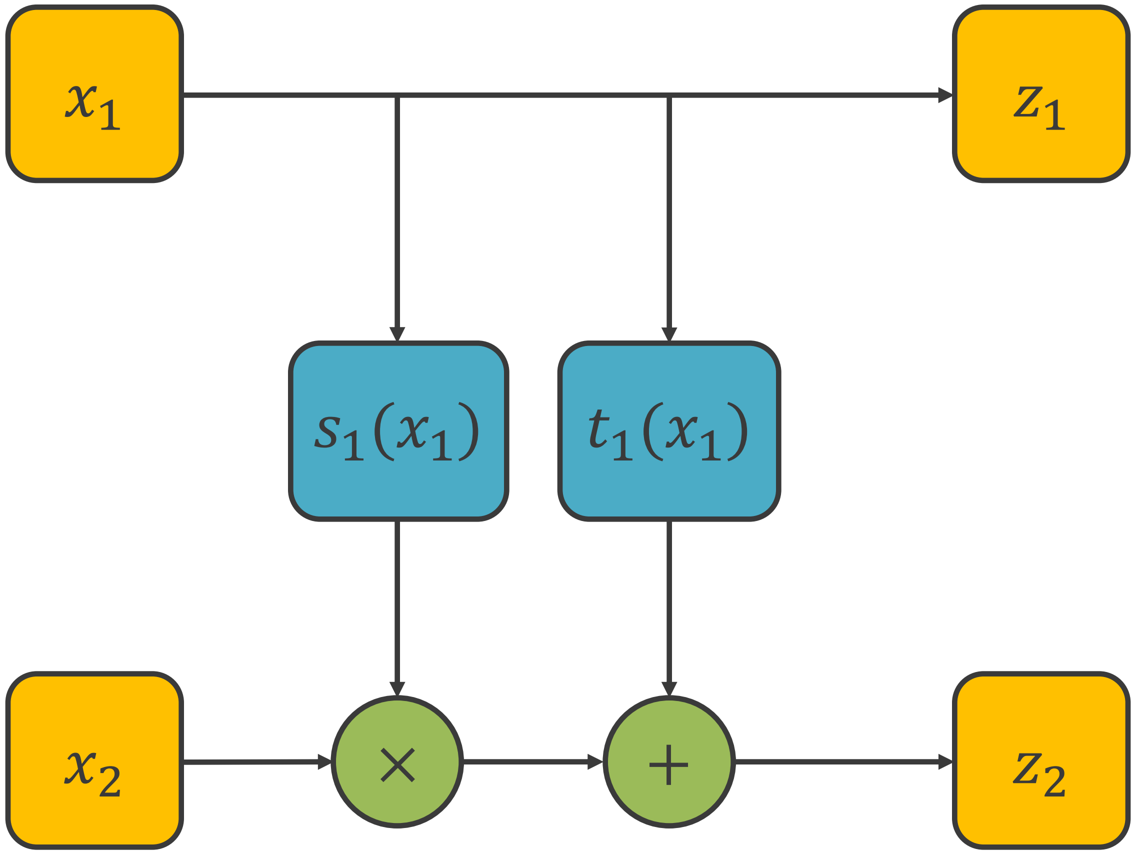
# Выбор функции

Как выбрать такую функцию $z_i = \boldsymbol{f}(x_i)$, чтобы она:

- была дифференцируемой,
- была обратимой?

# Real-NVP

# Real-NVP



# Функция

$$z = \boldsymbol{f}(x) = \begin{cases} z_{1:d} = x_{1:d} \\ z_{d+1:D} = x_{d+1:D} \odot \exp\big(\boldsymbol{s}(x_{1:d})\big) + \boldsymbol{t}(x_{1:d}) \end{cases}$$

где:

- $z_{1:d}$ - первые $d$ компонент вектора $z$;
- $\boldsymbol{s}(x_{1:d})$ и $\boldsymbol{t}(x_{1:d})$ – **нейронные сети** с $d$ входами и $D - d$ выходами;
- $\odot$ - поэлементное умножение.

https://arxiv.org/abs/1605.08803

# Якобиан

- Матрица первых производных:

$$\frac{\partial \boldsymbol{f}(x)}{\partial x} = \begin{pmatrix} \mathbb{I}_d & 0 \\ \frac{\partial z_{1:d}}{\partial x_{1:d}} & diag\left(\exp(\boldsymbol{s}(x_{1:d}))\right) \end{pmatrix}$$

- Значение Якобиана:

$$\left|\det \frac{\partial \boldsymbol{f}(x)}{\partial x}\right| = \exp\left(\sum_{j=d+1}^{D} \boldsymbol{s}(x_{1:d})_j\right)$$

https://arxiv.org/abs/1605.08803

# Обратная функция

$$x = \boldsymbol{f}^{-1}(z) = \begin{cases} x_{1:d} = z_{1:d} \\ x_{d+1:D} = (z_{d+1:D} - \boldsymbol{t}(x_{1:d})) \odot \exp\left(-\boldsymbol{s}(x_{1:d})\right) \end{cases}$$

https://arxiv.org/abs/1605.08803

# Пример

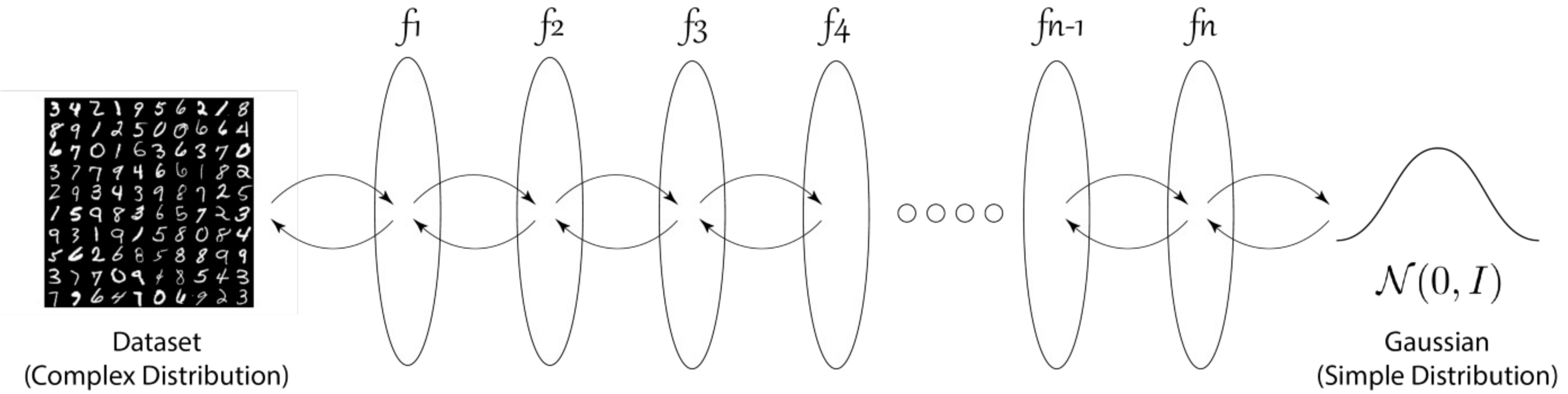
Рис.: https://github.com/laurent-dinh/laurent-dinh.github.io/blob/master/img/real_nvp_fig/celeba_samples.png

# Пример Glow

https://arxiv.org/pdf/1807.03039.pdf

# Больше слоев

# Больше слоев!




https://towardsdatascience.com/introduction-to-normalizing-flows-d002af262a4b

# Больше слоев!

Пусть $z_i = \boldsymbol{f_2}(y_i),\ \ y_i = \boldsymbol{f_1}(x_i)$.

Тогда:

$$p_x(x_i) = p_y\big(f_1(x_i)\big)\left|det\frac{\partial f_1(x_i)}{\partial x_i}\right|$$

$$p_y(y_i) = p_z\big(f_2(y_i)\big)\left|det\frac{\partial f_2(y_i)}{\partial y_i}\right|$$

В итоге получим:

$$p_x(x_i) = p_z\Big(f_2\big(f_1(x_i)\big)\Big)\left|det\frac{\partial f_2(y_i)}{\partial y_i}\right|\left|det\frac{\partial f_1(x_i)}{\partial x_i}\right|$$

# Другие модели

| Architecture | Coupling function | Flow name |
|---|---|---|
| Coupling, 3.4.1 | Affine, 3.4.4.1 | RealNVP<br>Glow |
| | Mixture CDF, 3.4.4.3 | Flow++ |
| | Splines, 3.4.4.4 | quadratic (C)<br>cubic<br>RQ-NSF(C) |
| | Piecewise Bijective, 3.4.4.7 | RAD |
| Autoregressive, 3.4.2 | Affine | MAF |
| | Polynomial, 3.4.4.6 | SOS |
| | Neural Network, 3.4.4.5 | NAF<br>UMNN |
| | Splines | quadratic (AR)<br>RQ-NSF(AR) |
| Residual, 3.5 | | iResNet |
| | | Residual flow |
| ODE, 3.6.1 | | FFJORD |

# Генеративные модели

**GAN:** minimax the classification error loss.

**VAE:** maximize ELBO.

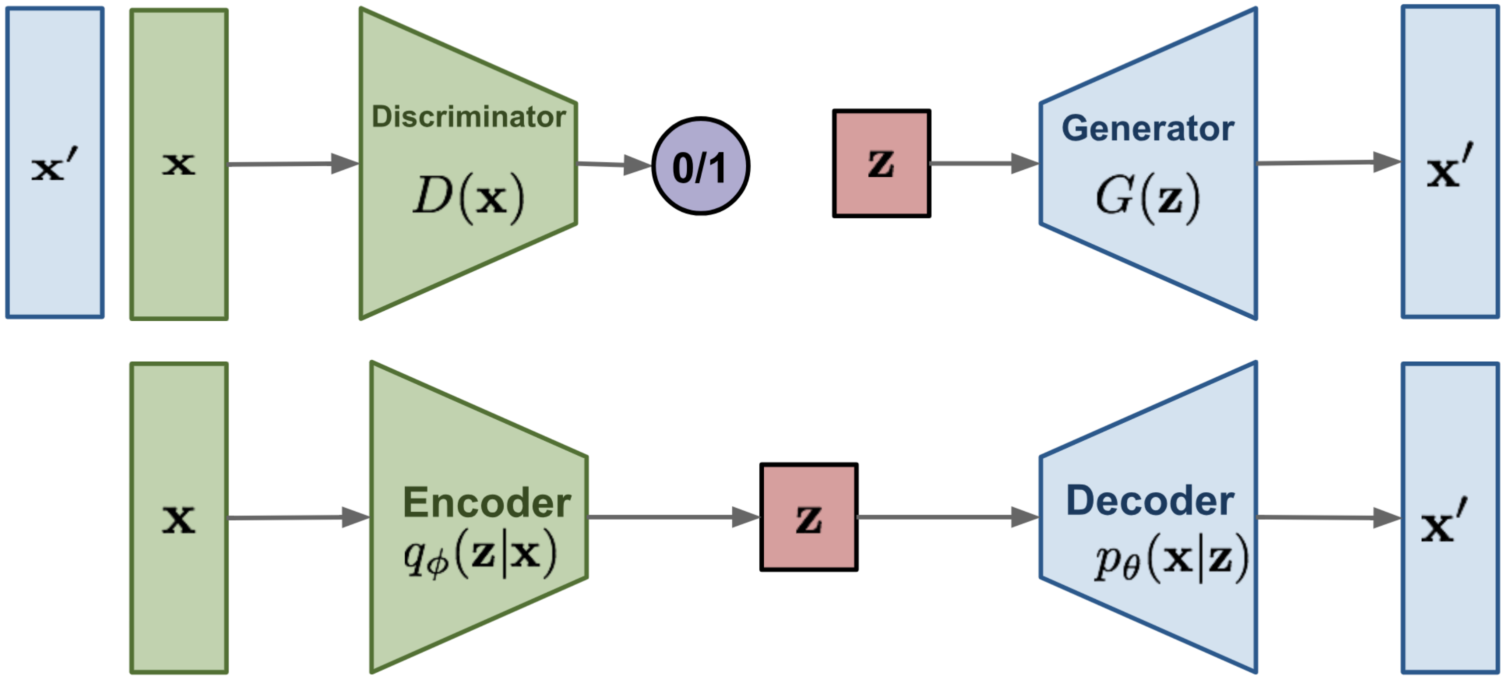



Рис.: https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html

# Генеративные модели

**GAN:** minimax the classification error loss.

**VAE:** maximize ELBO.

**Flow-based generative models:** minimize the negative log-likelihood

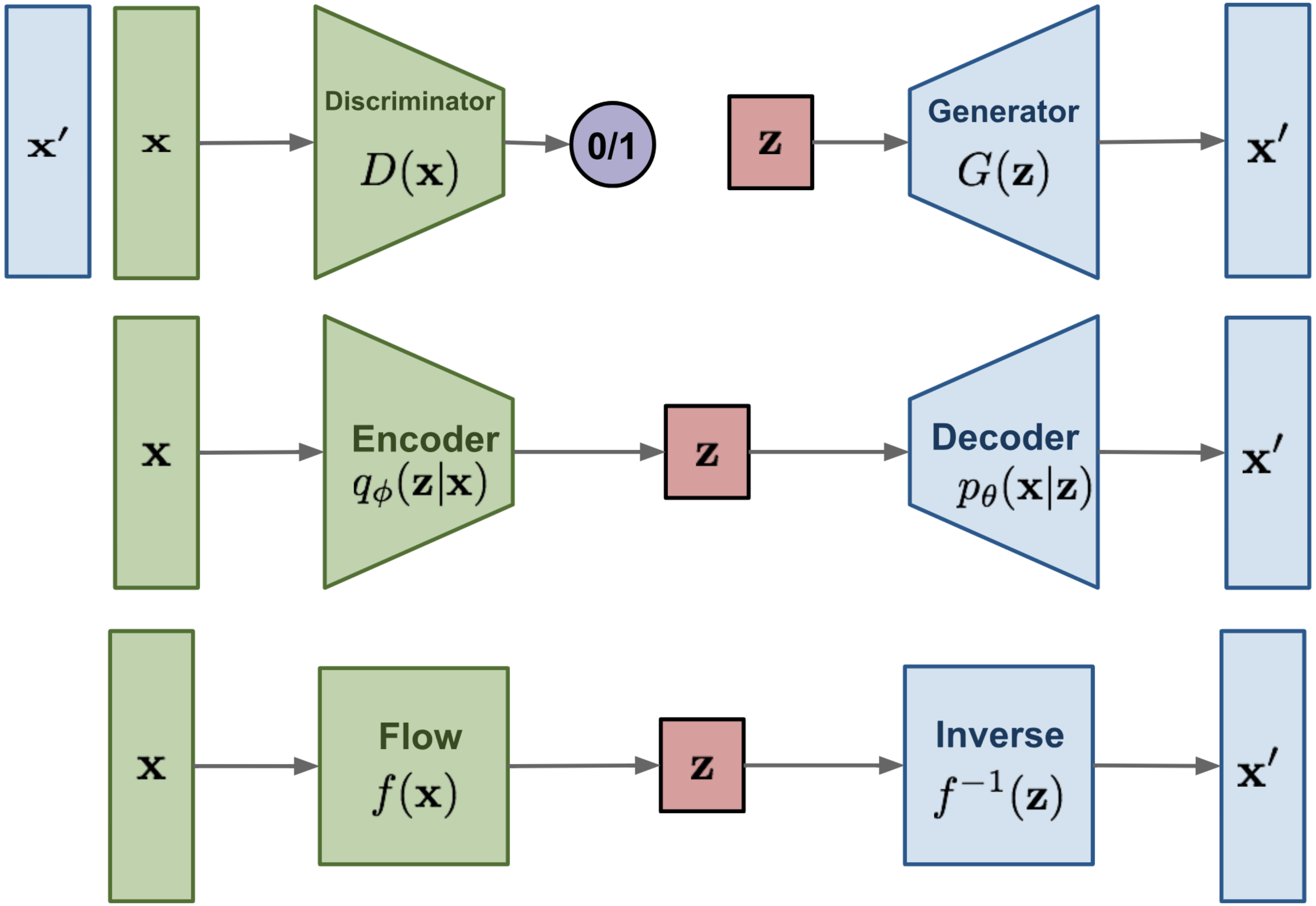



Рис.: https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html