Генеративные модели в машинном обучении

Лекция 5

Диффузионные генеративные модели

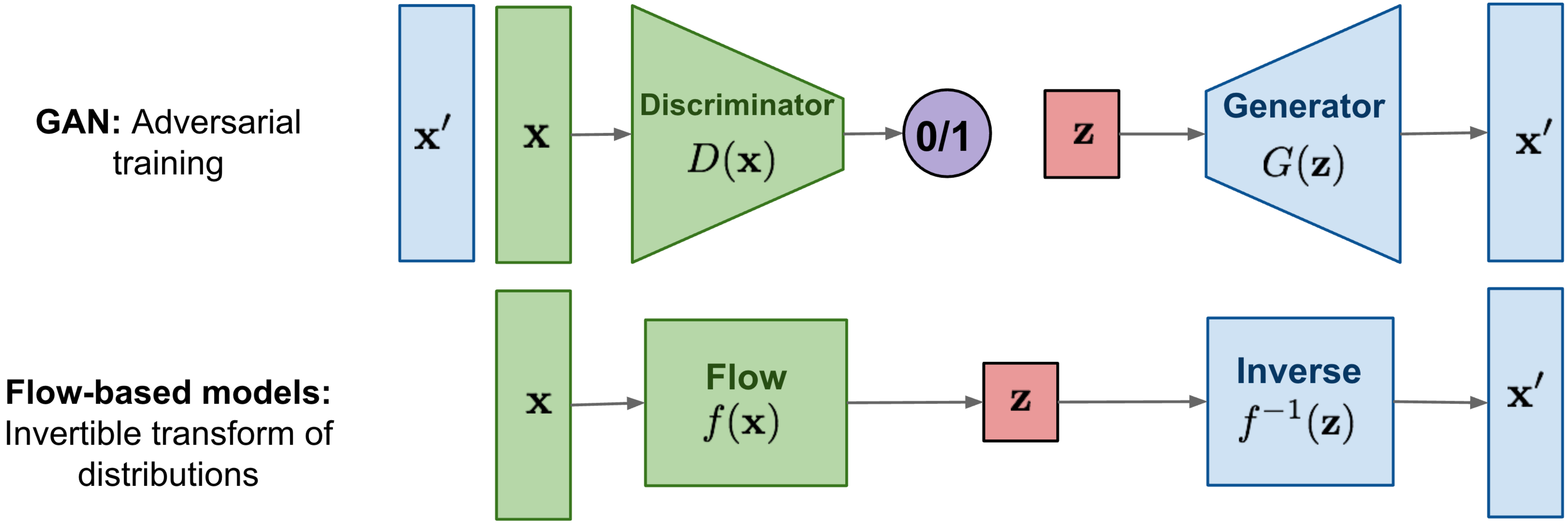
Михаил Гущин

mhushchyn@hse.ru

НИУ ВШЭ, 2024

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

В предыдущих лекциях



Source: https://lilianweng.github.io/posts/2021-07-11-diffusion-models

Приложения

Dall-E 2 (OpenAI)

“People walking on a beach during sunrise, a reflection of the sun on the water, realistic“

“Flamingos standing on water, red sunset, pink-red water reflection, photo-realistic, 4k“

Source: https://learnopencv.com/image-generation-using-diffusion-models/

Imagen (Google)

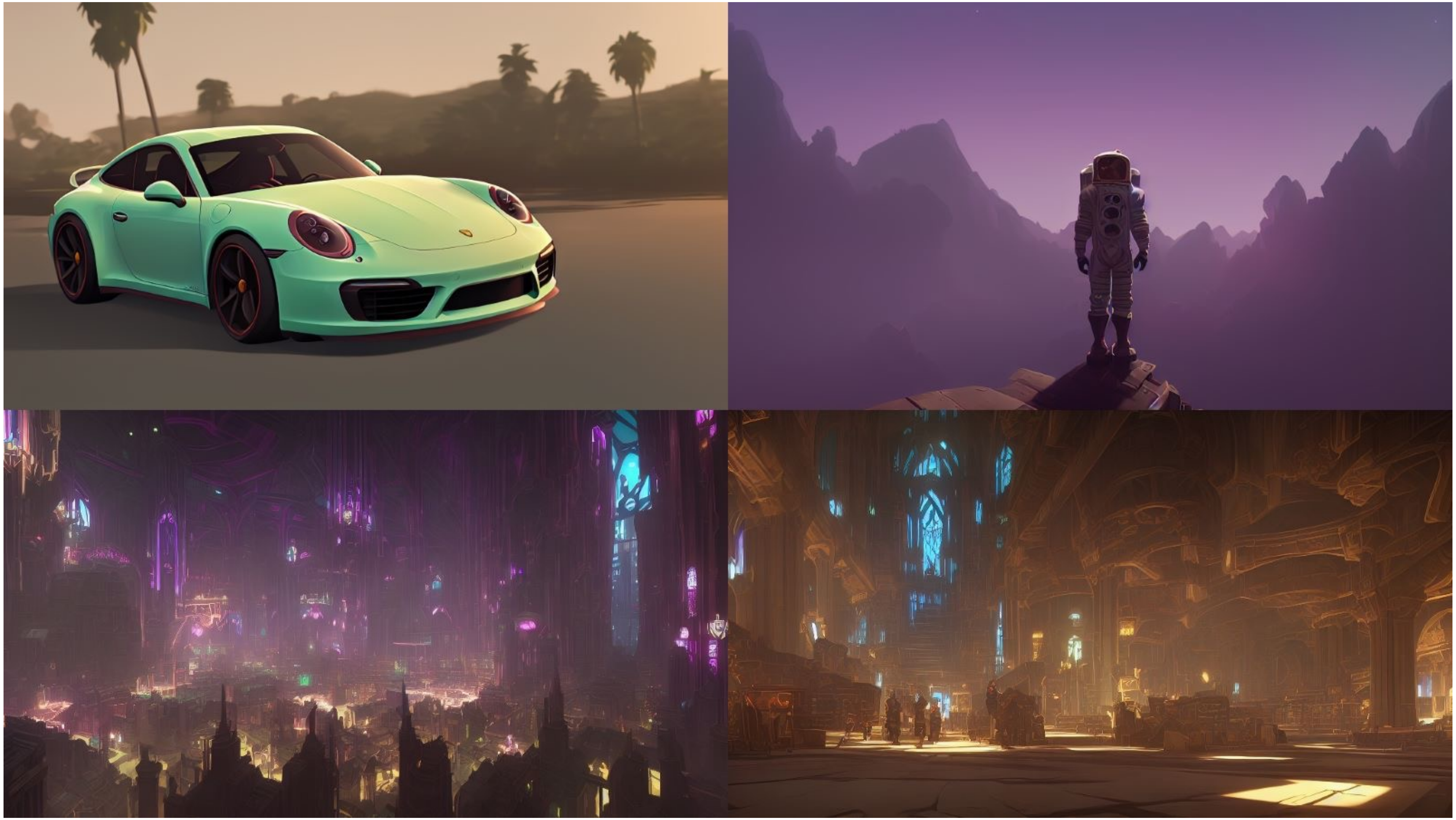
“A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.“

“A blue jay standing on a large basket of rainbow macarons“

“A brain riding a rocketship heading towards the moon“

Source: https://learnopencv.com/image-generation-using-diffusion-models/

Stable Diffusion (StabilityAI)



Source: https://learnopencv.com/image-generation-using-diffusion-models/

Text-to-Video

Make-a-Video

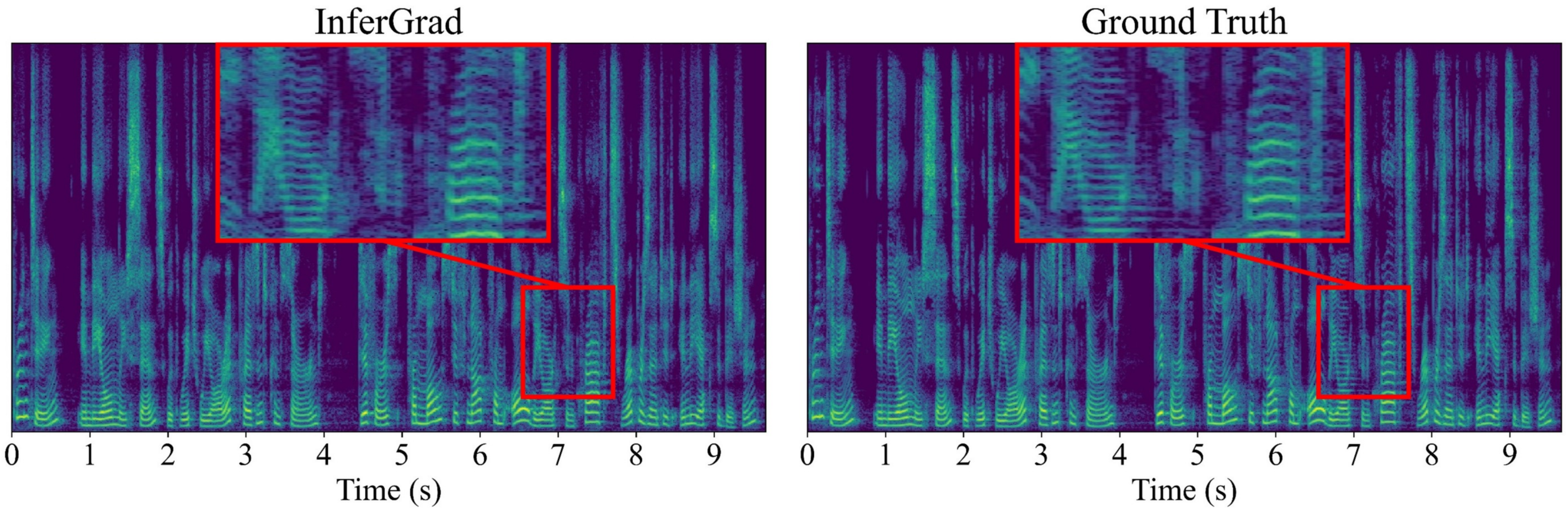
(a) A dog wearing a superhero outfit with red cape flying through the sky.

(b) There is a table by a window with sunlight streaming through illuminating a pile of books.

Source: stateof.ai 2022

Как отличить картинки

Text-to-speech generation



Source: https://github.com/heejkoo/Awesome-Diffusion-Models#text-to-speech

Prediction uncertainty estimation

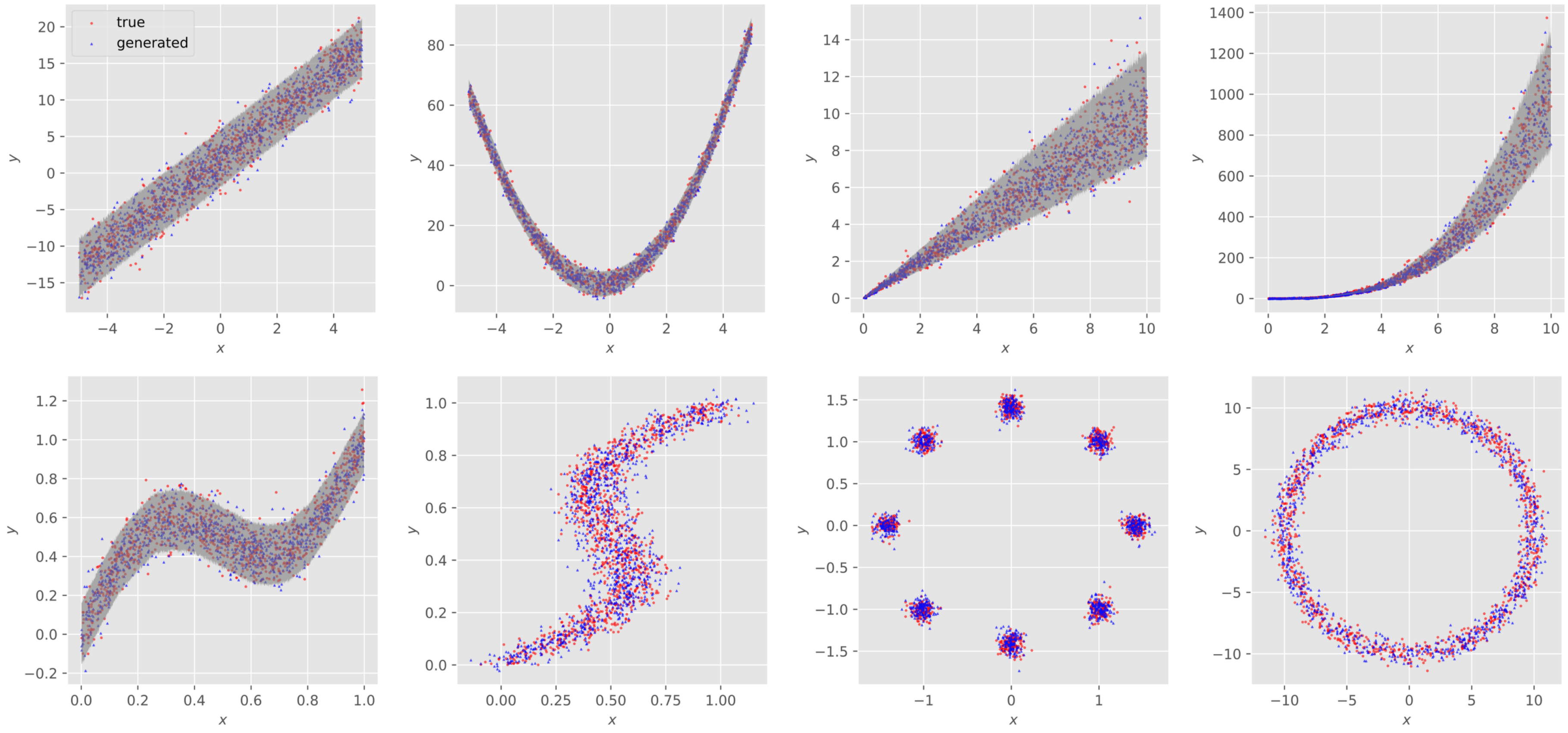
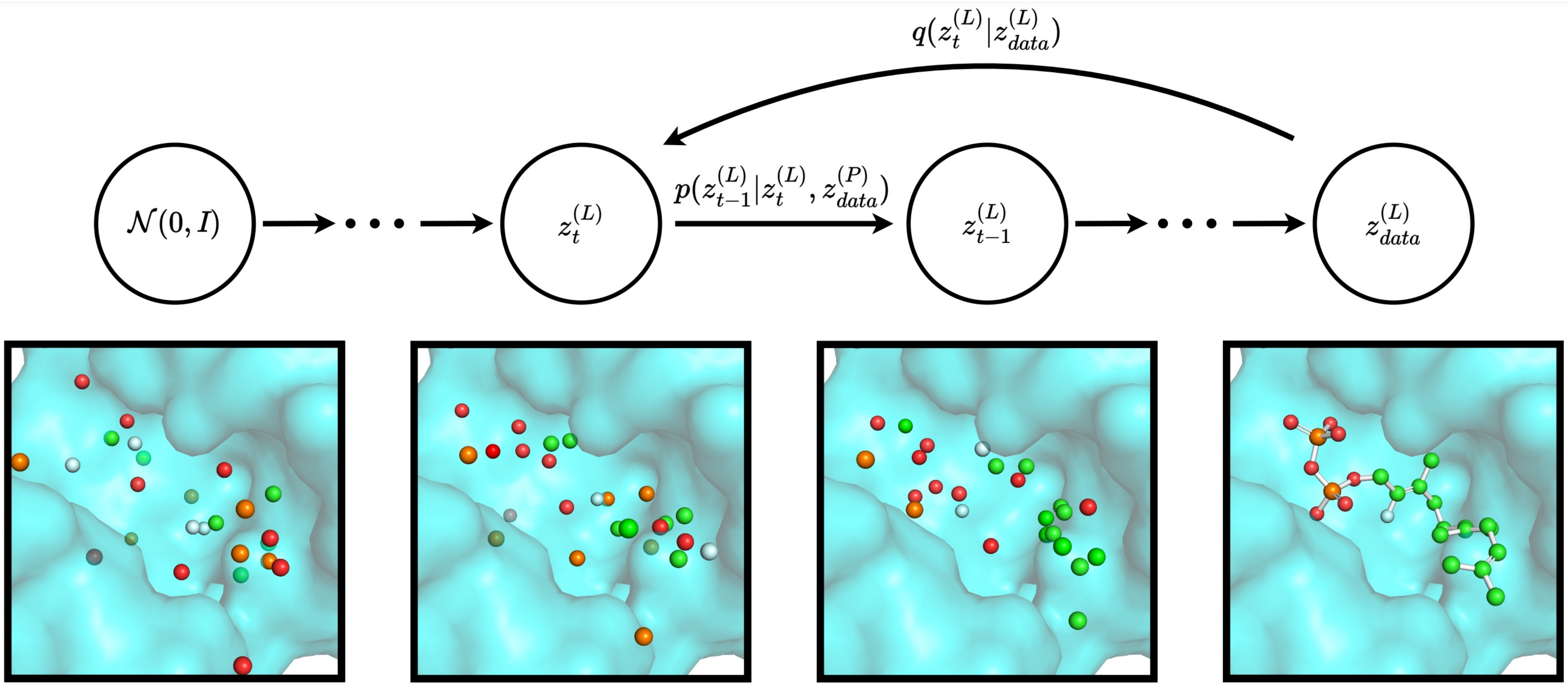


Figure 1: Regression toy example scatter plots. (**Top**) left to right: linear regression, quadratic regression, log-log linear regression, log-log cubic regression; (**Bottom**) left to right: sinusoidal regression, inverse sinusoidal regression, 8 Gaussians, full circle.

NeurIPS 2022 https://arxiv.org/abs/2301.03028

Molecular and material generation



Source: https://github.com/heejkoo/Awesome-Diffusion-Models#molecular-and-material-generation

Astronomical spectra generation

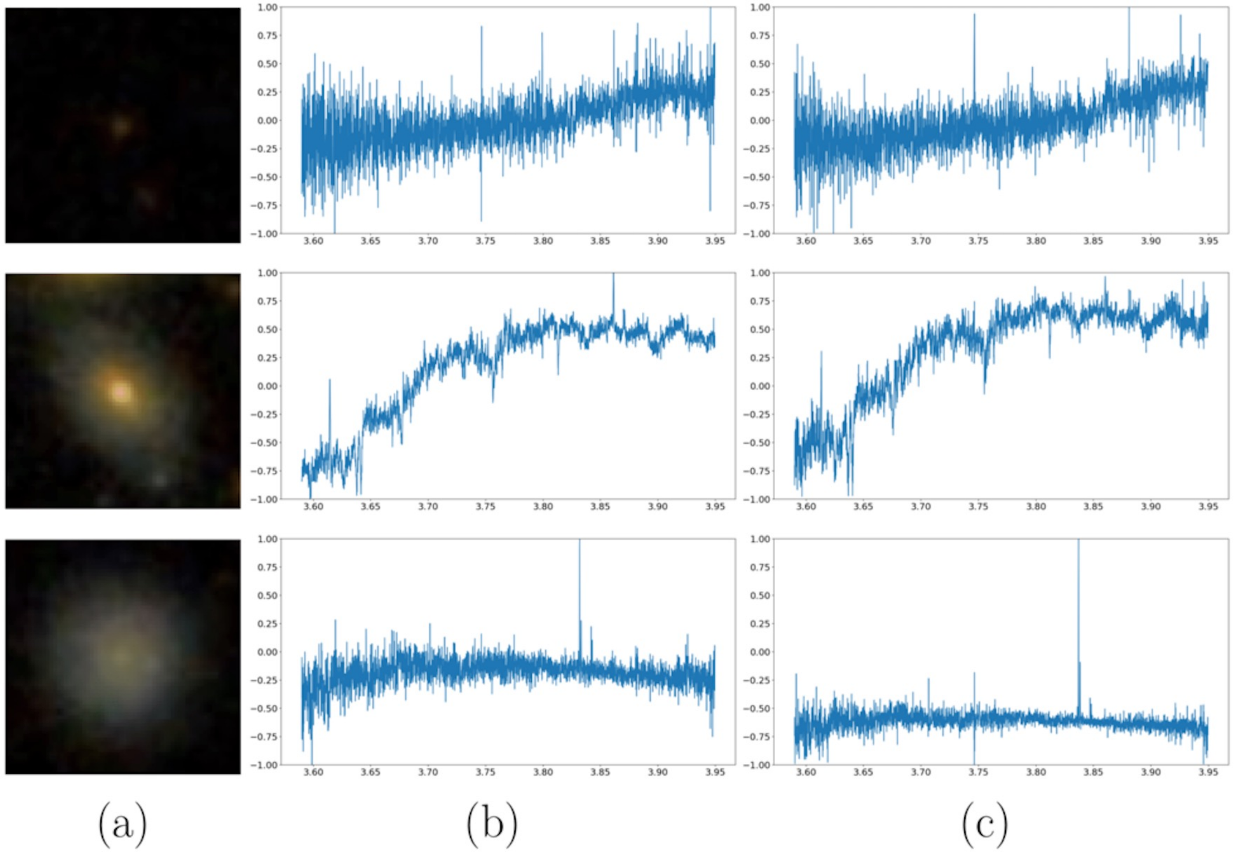


Figure 3: Generated spectra for the images in (a). In (b) we show the real spectra, in (c) the best match according to our contrastive model, out of 25 samples.

Source: https://ml4physicalsciences.github.io/2022/files/NeurIPS_ML4PS_2022_78.pdf

Dark matter density modelling

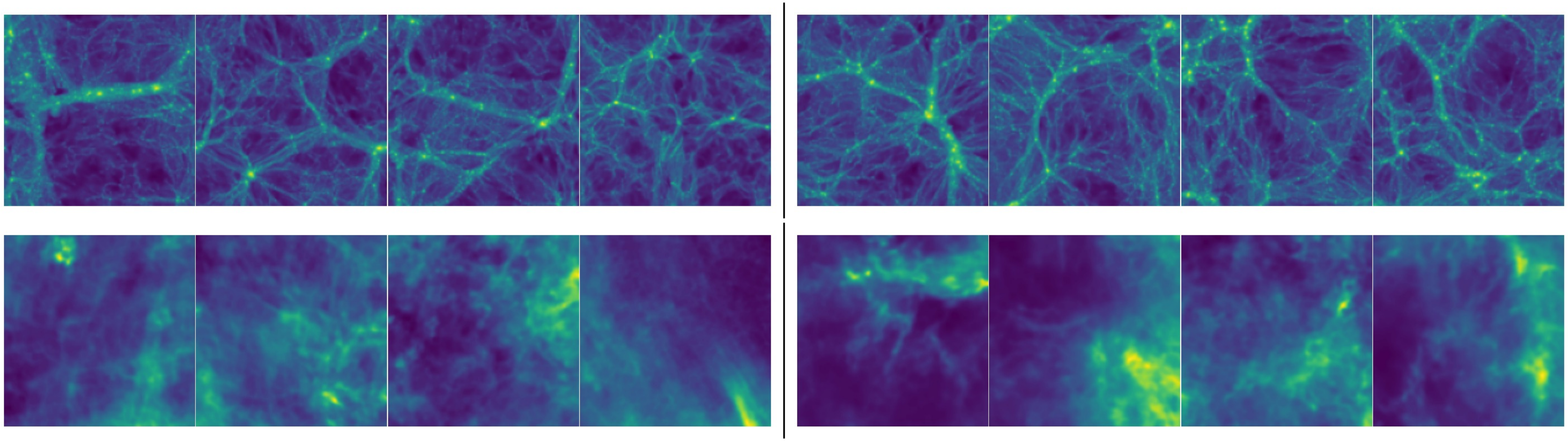
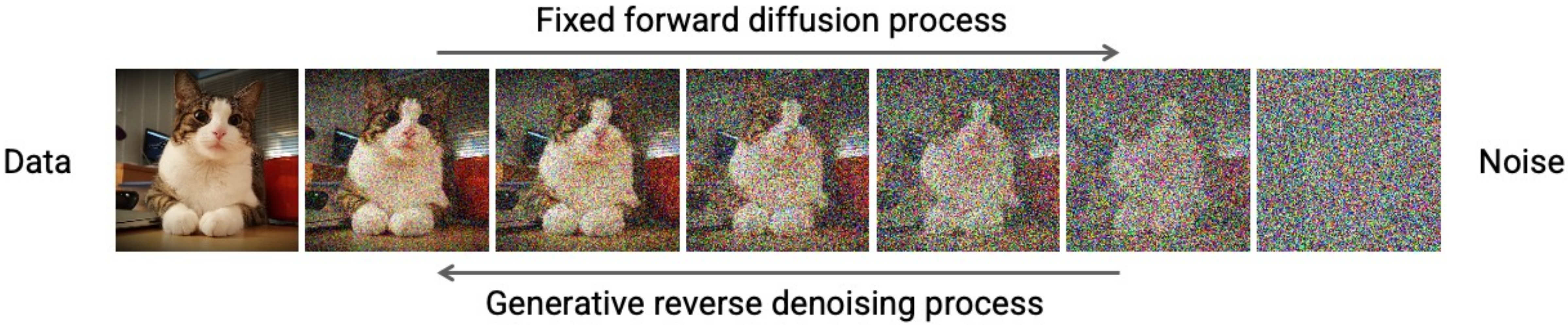


Figure 1: Four log cold dark matter mass density fields from the training data (top left) and from the sampled model (top right) at 128x128. Four samples of dust from the training data (bottom left) and from the trained model (bottom right).

Source: https://ml4physicalsciences.github.io/2022/files/NeurIPS_ML4PS_2022_25.pdf

Интуиция

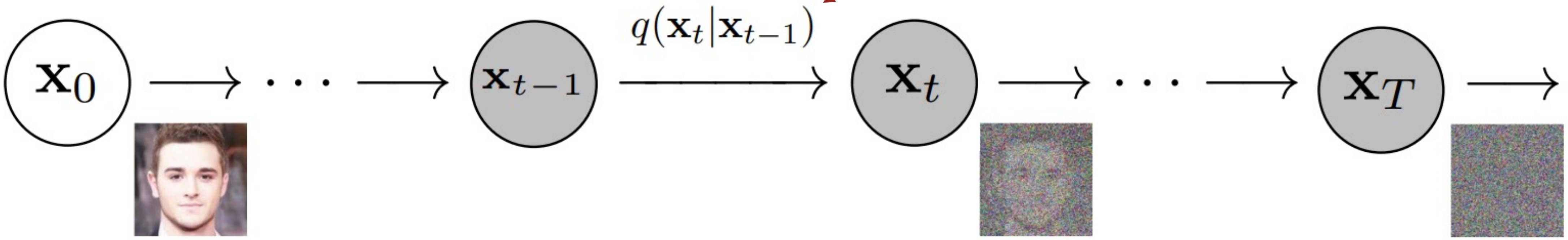
Общая идея диффузионных моделей



Источник: https://cvpr2022-tutorial-diffusion-models.github.io/

Процесс диффузии (зашумления)

Мы сами задаем как будем добавлять шум



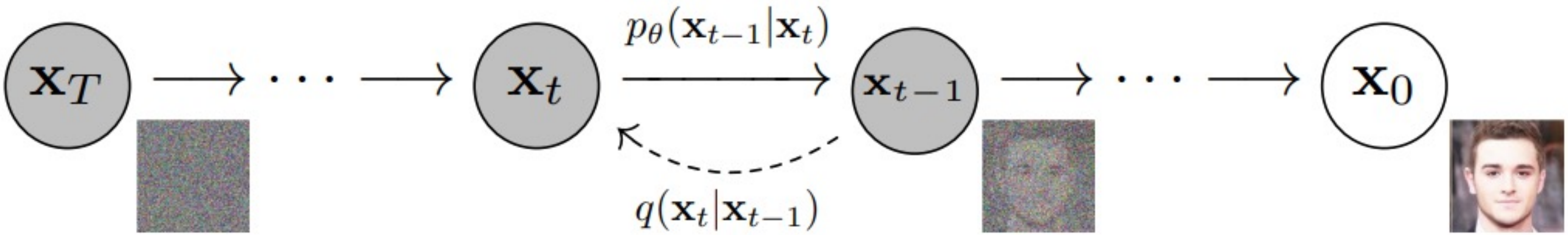
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \boldsymbol{\Sigma}_t = \beta_t \mathbf{I})$$

Константа

Источник: https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/

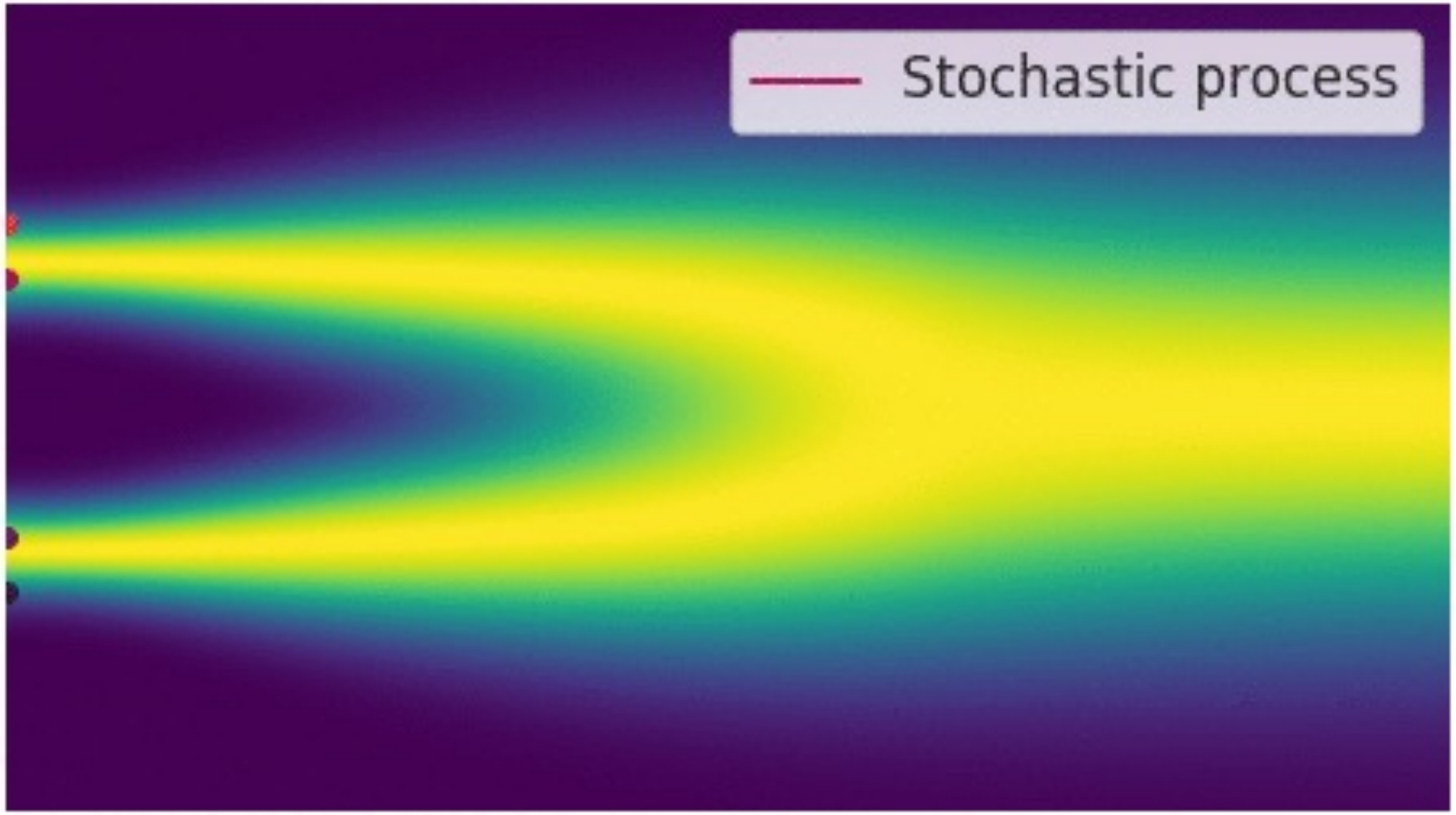
Обратный процесс

Учим с помощью нейронной сети



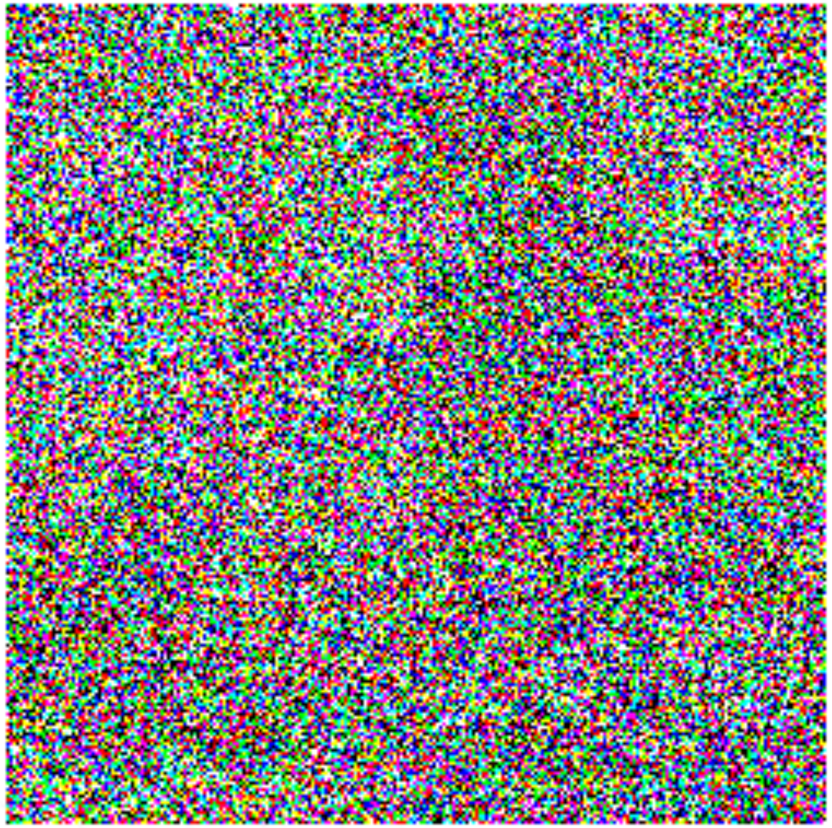
Источник: https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/

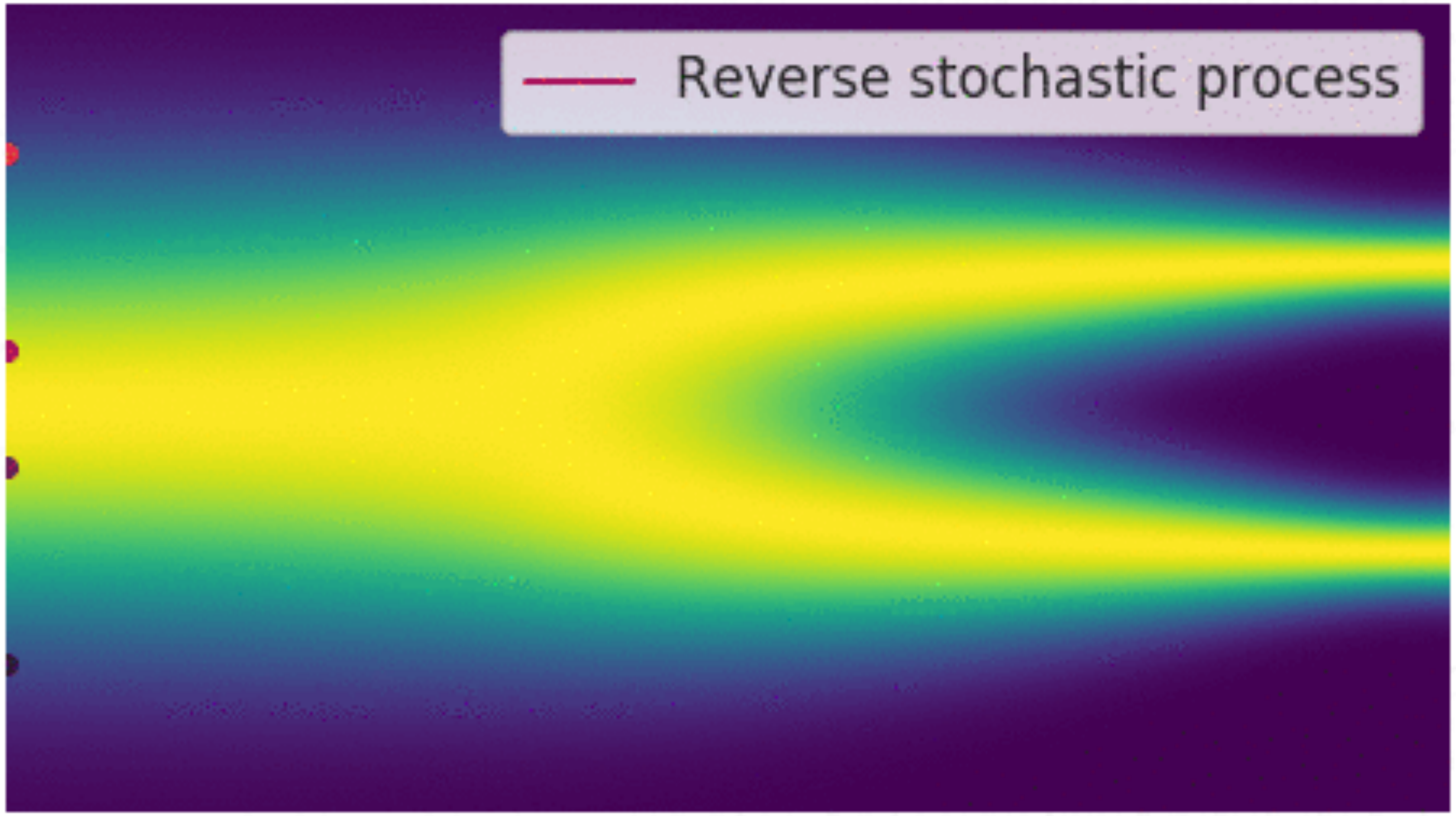
Демо: диффузия



Источник: https://yang-song.net/blog/2021/score/

Демо: обратный процесс

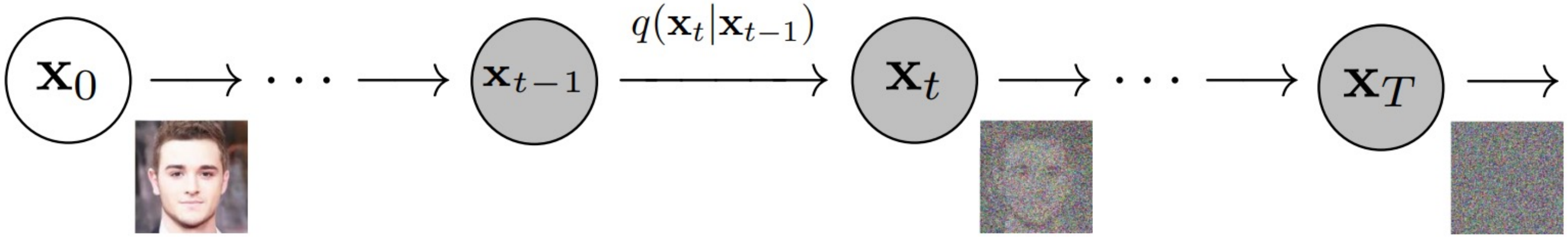




Источник: https://yang-song.net/blog/2021/score/

Диффузионные модели

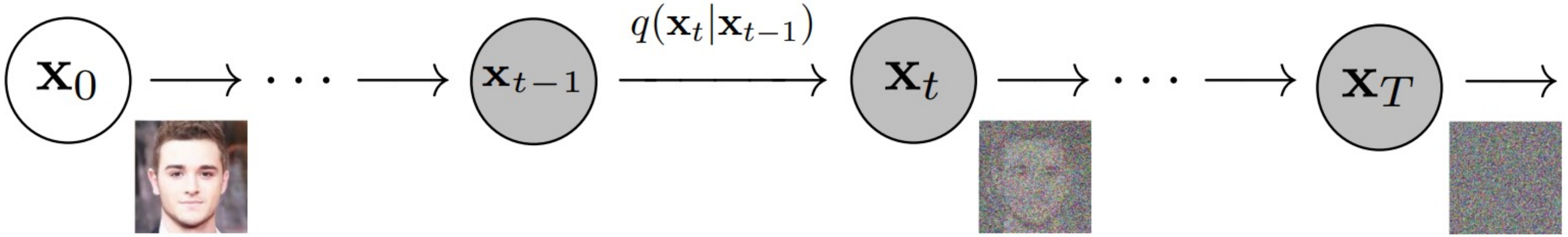
Процесс диффузии (зашумления)



- $x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1 - \alpha_t} \epsilon_t$ $\qquad \epsilon_t \sim N(0, I), \;\; \alpha_t = 1 - \beta_t$

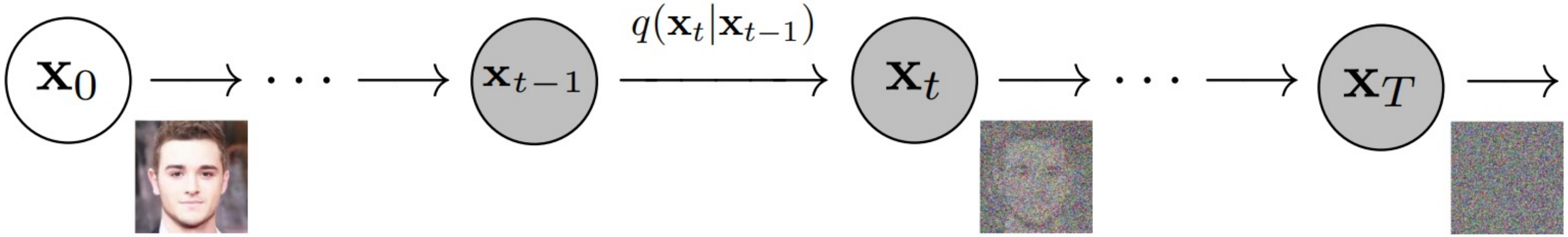
Мы сами так задали процесс

Процесс диффузии (зашумления)



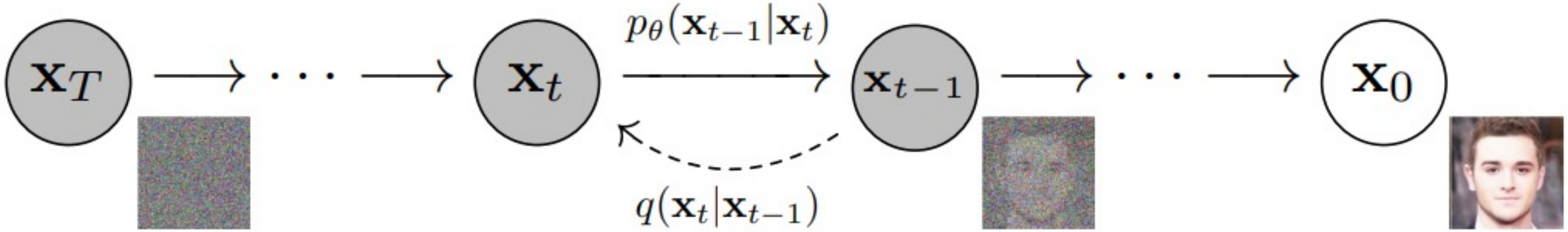
- $x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t$ $\qquad \epsilon_t \sim N(0, I), \quad \alpha_t = 1 - \beta_t$
- $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ $\qquad \epsilon \sim N(0, I), \quad \bar{\alpha}_t = \prod_{i=1}^{t}\alpha_t$

Процесс диффузии (зашумления)



- $x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t$ $\qquad \epsilon_t \sim N(0, I), \quad \alpha_t = 1 - \beta_t$
- $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ $\qquad \epsilon \sim N(0, I), \quad \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_t$
- $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right) + \tilde{\beta}_t z$ $\qquad z \sim N(0, I), \quad \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$

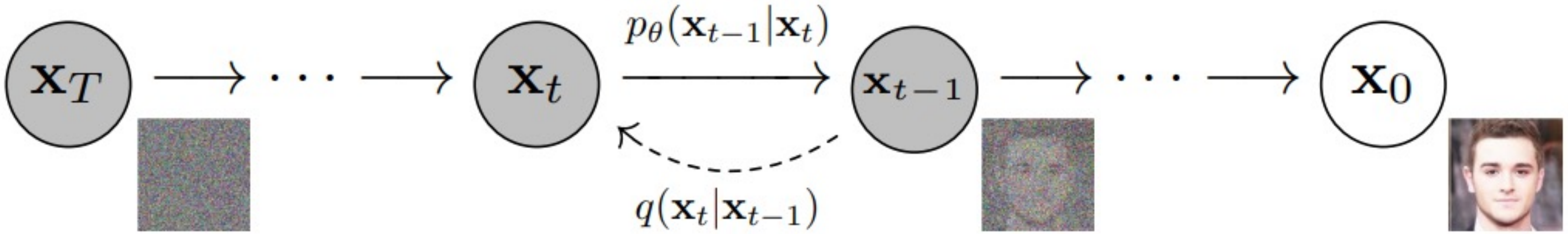
Обратный процесс



- $\hat{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t z$ $\qquad z \sim N(0, I),\quad \sigma_t = const$

Предсказываем нейронной сетью

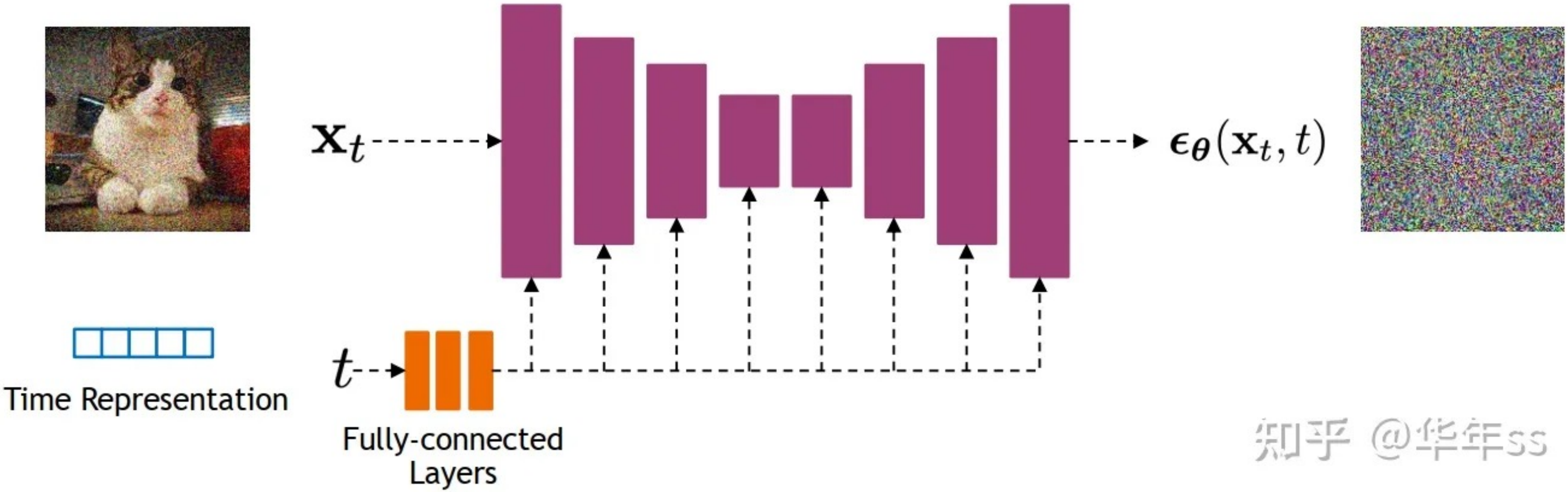
Обратный процесс



- $\hat{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t z \qquad z \sim N(0, I), \quad \sigma_t = const$
- Функция потерь для обучения:

$$L_t = \|x_{t-1} - \hat{x}_{t-1}\|_2^2 \propto \|\epsilon - \epsilon_\theta(x_t, t)\|_2^2 \to \min_\theta$$

Архитектура нейронной сети



Источник: https://www.zhihu.com/question/536012286/answer/2683123893

Алгоритм обучения

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Источник: https://arxiv.org/pdf/2006.11239.pdf

Алгоритм генерации

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

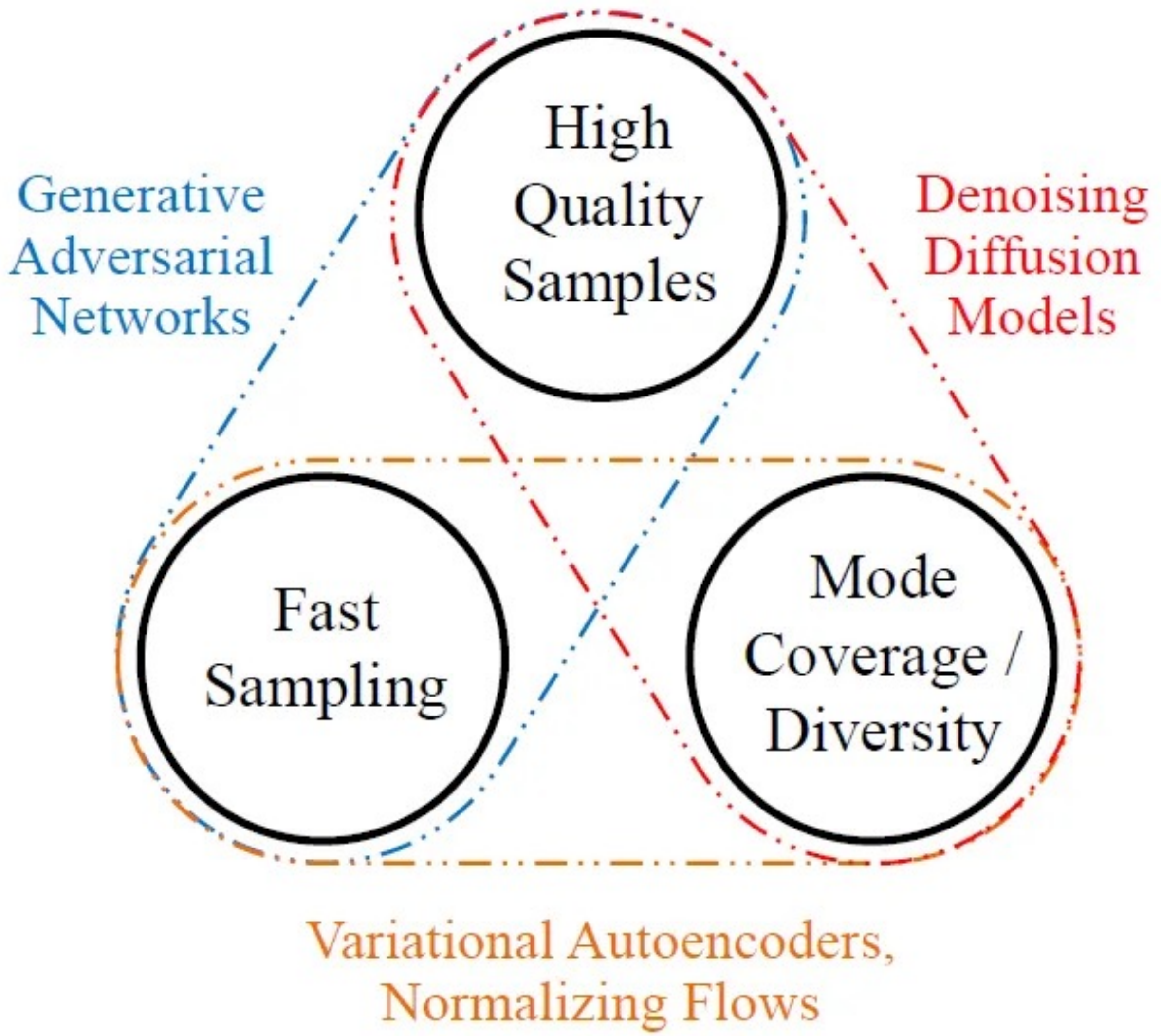
Источник: https://arxiv.org/pdf/2006.11239.pdf

Значения гиперпараметров

- $T = 1000$
- $\beta_1 < \beta_2 < \cdots < \beta_t < \cdots < \beta_T$
- $\beta_1 = 0.0001, \beta_T = 0.02$
- $\sigma_t^2 = \beta_t$ or $\sigma_t^2 = \tilde{\beta}_t$

Заключение

Generative learning trilemma



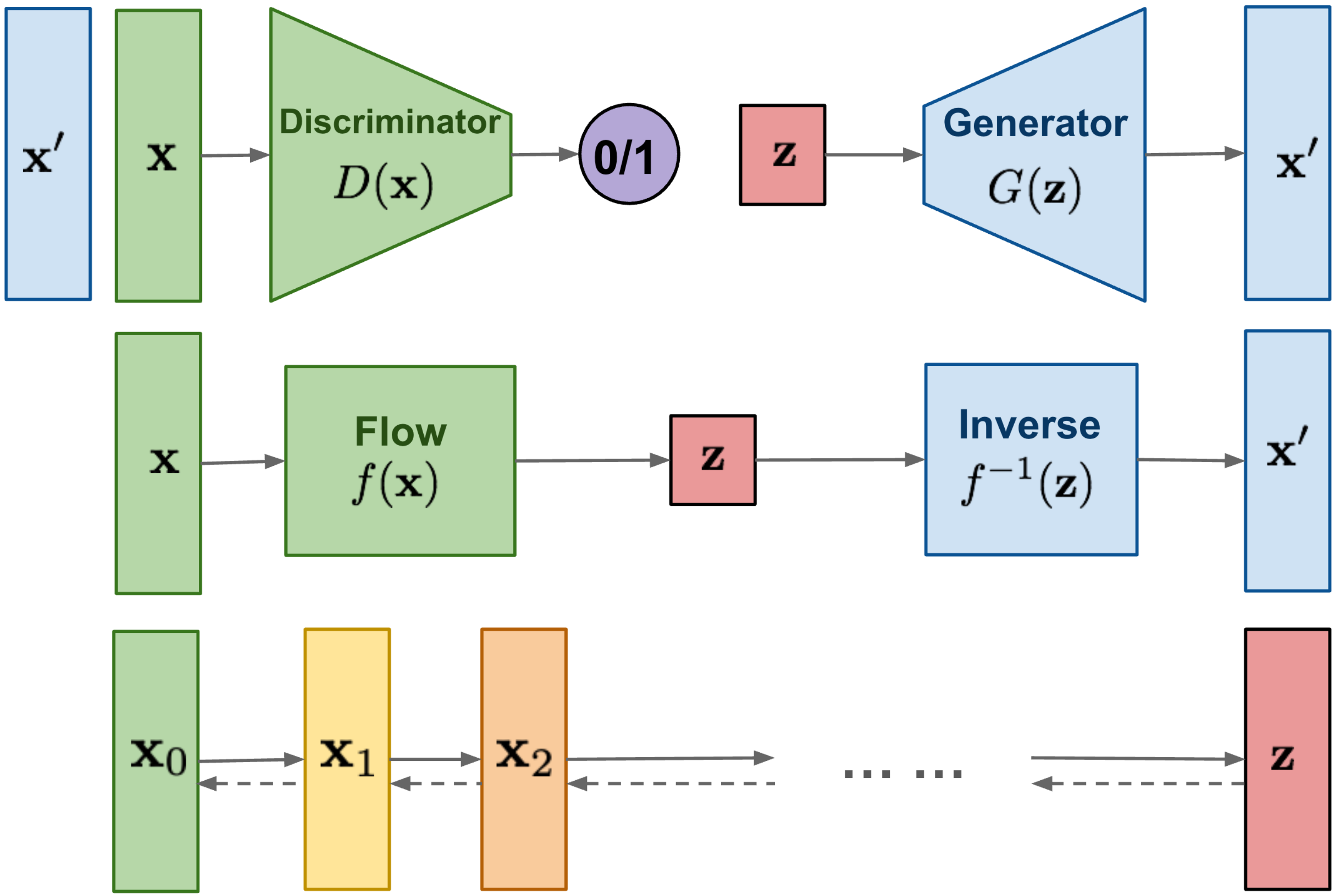
Source: https://zhuanlan.zhihu.com/p/503932823

Заключение

GAN: Adversarial training

Flow-based models: Invertible transform of distributions

Diffusion models: Gradually add Gaussian noise and then reverse



Source: https://lilianweng.github.io/posts/2021-07-11-diffusion-models