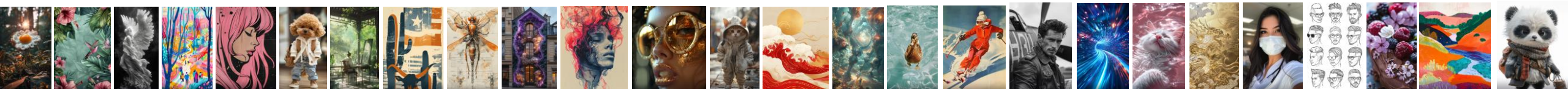
Diffusion Models
DDPM



Все ли картины настоящие?

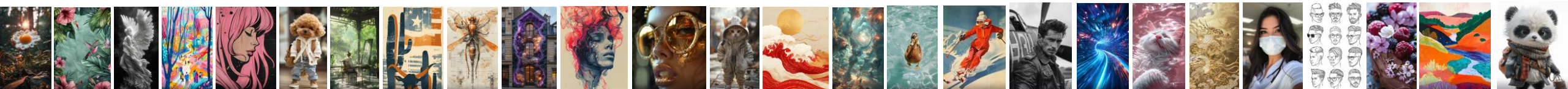
Хосин Зиани.
Венеция

MidJourney.
Ilya Repin painting depicting a foggy morning in Venice

Клод Моне.
Вестминстерский дворец

Сильные стороны диффузионных моделей

- Очень высокое качество генерации
- Способны к переносу стиля
- Генерация может быть как безусловная, так и условная
- Можно применять не только для генерации изображений/видео, но и:
  - к задачам генерации последовательностей
  - для повышения качества/детализированности изображения
  - для расшумления изображений

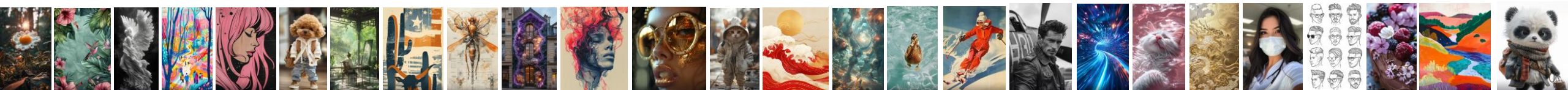


DDPM: основная идея

- Пусть есть пространство объектов $\{\mathbf{x}_0\}, \quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$

- **Предложение 1:** При долгом последовательном добавлении шума к объектам объекты становятся распределенными так же, как и шум

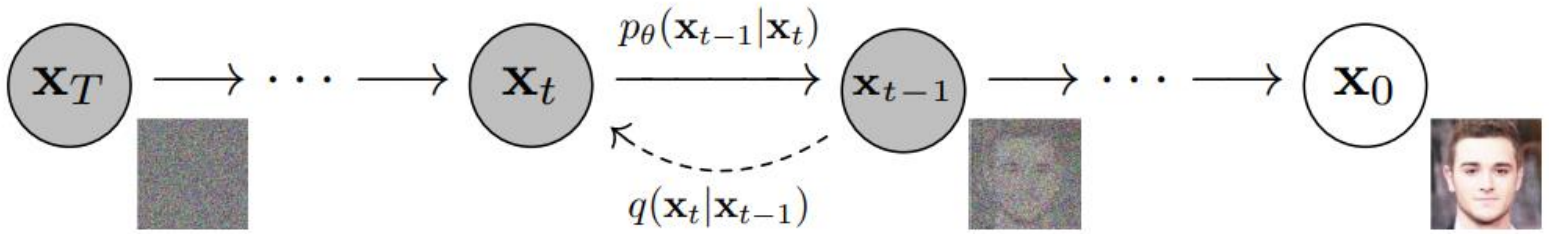
- **Предложение 2:** Убирая шум от зашумленных объектов так же, как и добавляли его, получим исходное распределение объектов

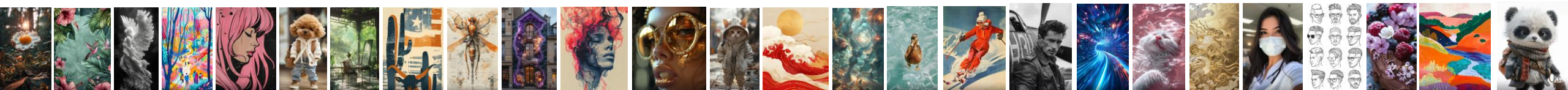


DDPM: основная идея

Идея: Используя предложения 1 и 2, можно, аппроксимируя обратные шаги, научиться восстанавливать из шума распределение объектов $q(\mathrm{x}_0)$.

Сэмплировать объекты можно так же: берём объект, заполненный шумом, и “восстанавливаем” его аппроксимациями обратных шагов.

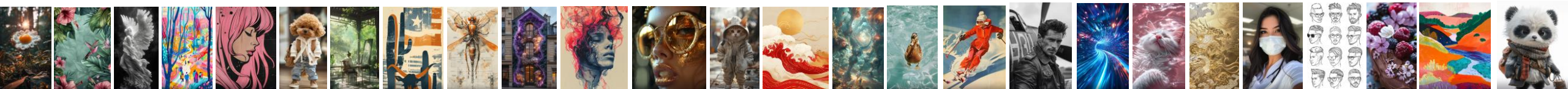




Немного определений и допущений

- Пусть есть пространство объектов $\{\mathbf{x}_0\}, \quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
- $\mathbf{x}_1, \ldots, \mathbf{x}_T$ - латентные переменные ($\mathbf{x}_t$ - объект после применения к нему t шагов forward process)

- Пусть для простоты весь рассматриваемый шум – нормальный (многомерное нормальное распределение).



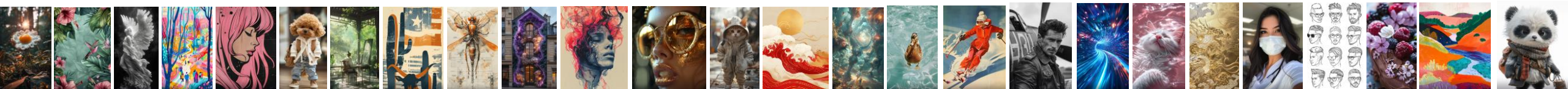
Немного определений и допущений

- Forward (diffusion) process:
  - Цепь Маркова
  - Стартовое состояние - $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  - Добавляет шум к данным согласно variance schedule $\beta_1, \ldots, \beta_T$, каждый переход между соседними двумя состояниями происходит со следующей вероятностью:

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

  - Воспользуемся марковским свойством. Тогда:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$



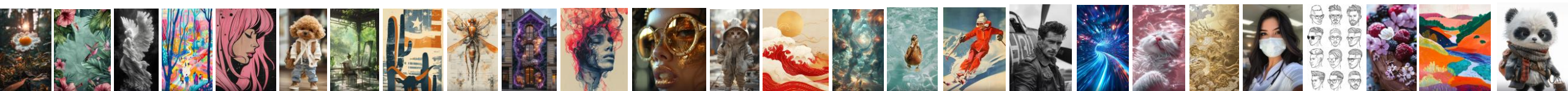
Немного определений и допущений

- Reverse process:
  - Тоже цепь Маркова
  - Стартовое состояние - $\mathbf{x}_T \sim p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$
  - Убирает шум гауссовскими обратными шагами (будем обучать модель их выполнять)
  - Каждый переход между соседними двумя состояниями происходит со следующей вероятностью:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

  - Воспользуемся марковским свойством. Тогда:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$



Немного определений и допущений

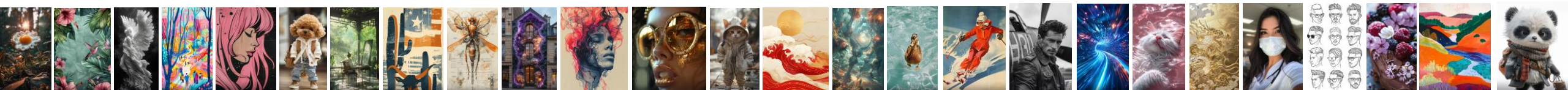
- Полезная репараметризация: $\alpha_t := 1 - \beta_t$, $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$
- С её помощью можно вывести, что:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}),$$

где $\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t$,

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

- Также верно, что: $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$



Причём тут нейросети?

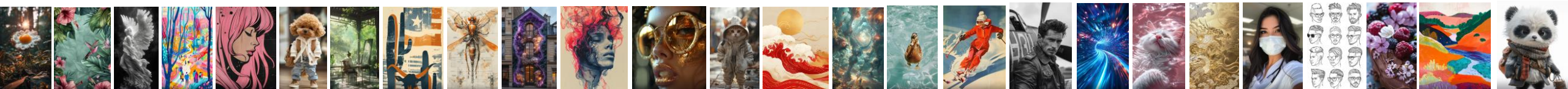
Задача:

- Хотим обучить модель предсказывать параметры распределений $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$.

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

- Хотим максимизировать $p_\theta(\mathbf{x}_0)$ $\Leftrightarrow$ хотим минимизировать $\underbrace{\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right]}_{\text{NLL Loss}}$

- Хотим обучаться с помощью градиентного спуска (так как это просто)



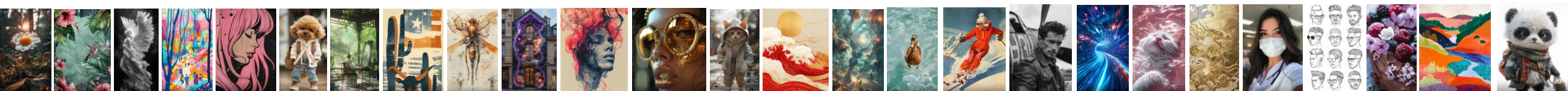
Проблема

Непонятно, как полученный лосс оптимизировать

В общем виде такое правдоподобие нельзя найти, так как не знаем $\mathbf{x}_{1:T}$

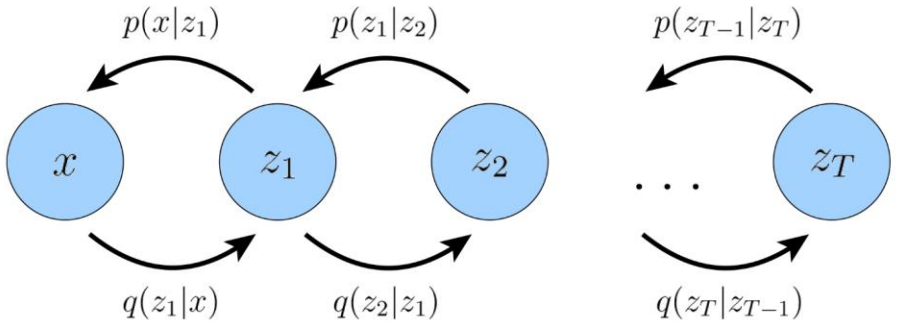
Решение

Надо перейти к какой-нибудь верхней оценке на лосс и минимизировать её



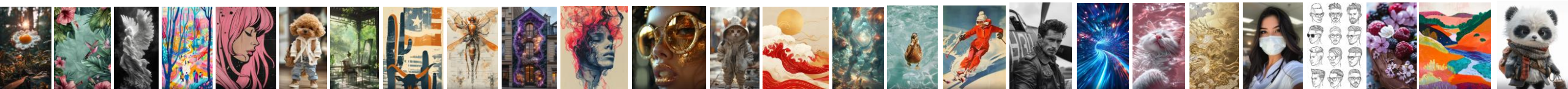
ELBO

- Заметим: построенные forward и reverse процессы – Markovian Hierarchical VAE.



- Для VAE верно неравенство ELBO:

$$\log p_\theta(\mathbf{x}_0) \geqslant \mathbb{E}_{\mathbf{x}_{1:T}\sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log\left(\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right)\right]$$

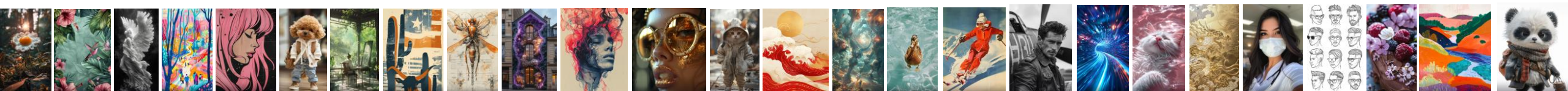


Применяем ELBO к лоссу

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L$$

ELBO

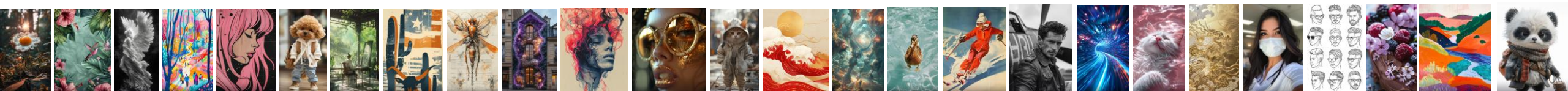
марковское свойство



Преобразуем лосс дальше

$$L = \mathbb{E}_q \Bigg[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \Bigg]$$

Хотим, чтобы соотв.
$q(\mathrm{x}_{t-1}|\mathrm{x}_t, \mathrm{x}_0)$ и $p_\theta(\mathrm{x}_{t-1}|\mathrm{x}_t)$
были максимально “близки”



Преобразуем лосс дальше

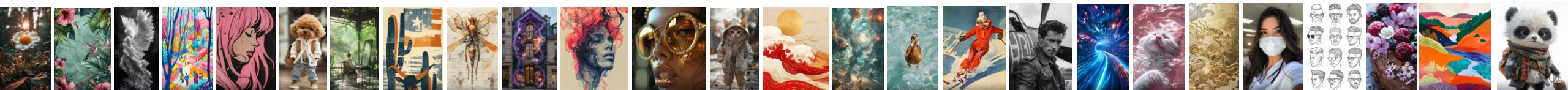
$$L = \mathbb{E}_q \left[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right]$$

$$\mathbf{x}_T \sim p(\mathbf{x}_T) = \underbrace{\mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})}_{\text{известно}}$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \underbrace{\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}}_{\text{Неизвестно, если не знаем variance schedule. Считаем, что знаем, так проще}})$$

Итог: не зависит от параметров

Экспериментально показано, что без оптимизации данного члена метрики лучше => не будем его рассматривать



Упрощаем KL-дивергенции

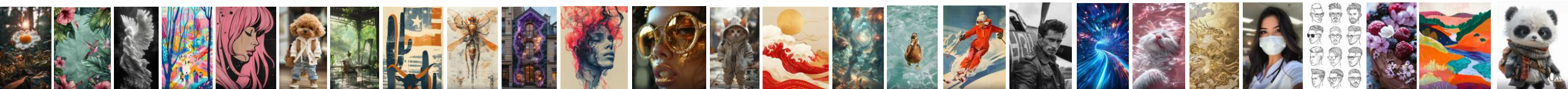
$$L = \mathbb{E}_q \Bigg[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \Bigg]$$

- **I результат:** $L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2 \right] + C$ **=>** $\|\underbrace{\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0)}_{\text{ground truth}} - \underbrace{\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)}_{\text{предсказание}}\|^2 \longrightarrow \min_\theta$

  ground truth $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$

  предсказание $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$

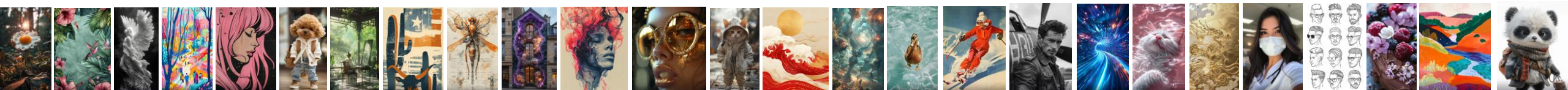
- **II результат:** $L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2 \right] \longrightarrow \min_\theta$



Какой лосс выбрать?

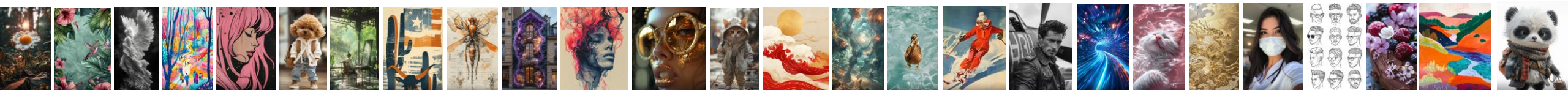
Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

Objective	IS	FID
$\tilde{\mu}$ prediction (baseline)		
L, learned diagonal Σ	7.28 ± 0.10	23.69
L, fixed isotropic Σ	8.06 ± 0.09	13.22
$\|\tilde{\mu} - \tilde{\mu}_\theta\|^2$	–	–
ϵ prediction (ours)		
L, learned diagonal Σ	–	–
L, fixed isotropic Σ	7.67 ± 0.13	13.51
$\|\tilde{\epsilon} - \epsilon_\theta\|^2$ (L_{simple})	$\mathbf{9.46 \pm 0.11}$	**3.17**



Оптимизация forward process

- В наивном варианте зашумляем $\mathbf{x}_0$ до $\mathbf{x}_t$ за $O(t)$ - долго
- Ранее вывели: $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$ => можно зашумлять за $O(1)$



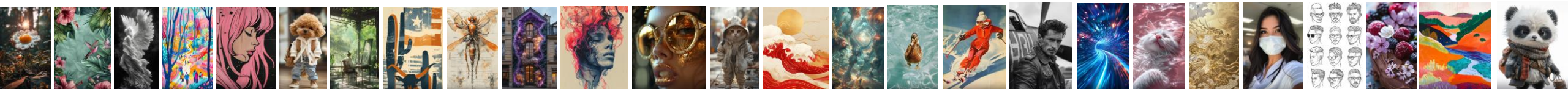
Алгоритмы обучения и сэмплирования

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
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

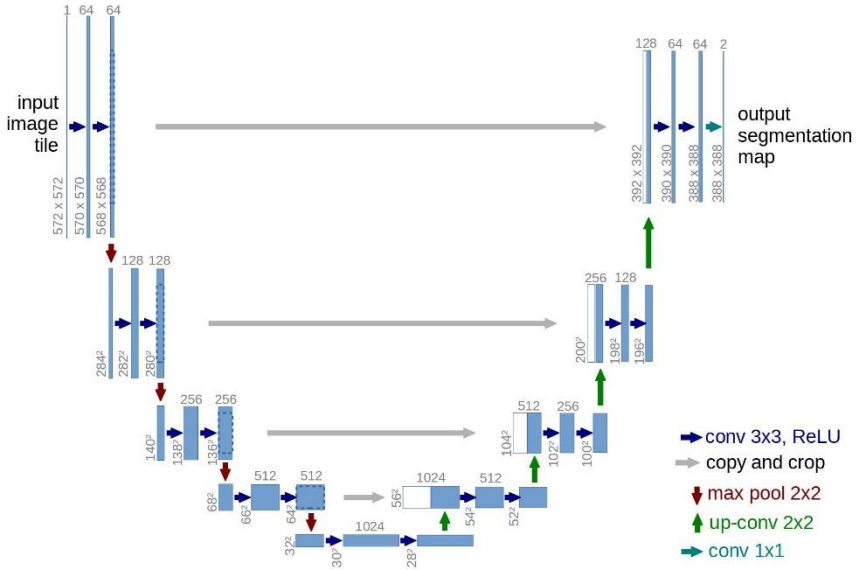
Algorithm 2 Sampling

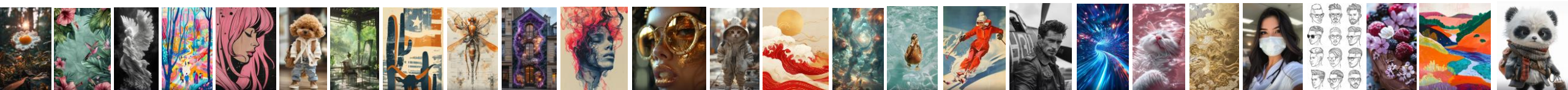
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}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



Архитектурные решения

- Backbone: U-Net со skip connections
- Параметры (предсказания) привязаны ко времени с помощью positional embedding’ов
- Между U-Net-блоками – Attention

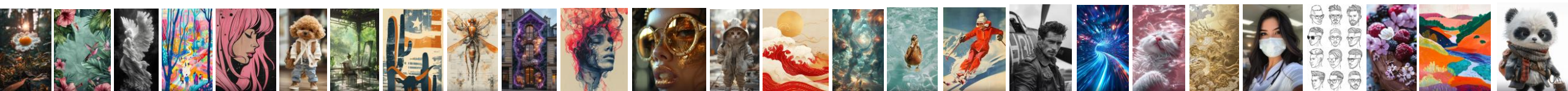




Результаты

CIFAR10 results.

Model	IS	FID
Conditional		
EBM [11]	8.30	37.9
JEM [17]	8.76	38.4
BigGAN [3]	9.22	14.73
StyleGAN2 + ADA (v1) [29]	**10.06**	**2.67**
Unconditional		
Diffusion (original) [53]		
Gated PixelCNN [59]	4.60	65.93
Sparse Transformer [7]		
PixelIQN [43]	5.29	49.46
EBM [11]	6.78	38.2
NCSNv2 [56]		31.75
NCSN [55]	8.87 ± 0.12	25.32
SNGAN [39]	8.22 ± 0.05	21.7
SNGAN-DDLS [4]	9.09 ± 0.10	15.42
StyleGAN2 + ADA (v1) [29]	$\mathbf{9.74} \pm 0.05$	3.26
Ours (L, fixed isotropic $\mathbf{\Sigma}$)	7.67 ± 0.13	13.51
Ours (L_{simple})	9.46 ± 0.11	**3.17**



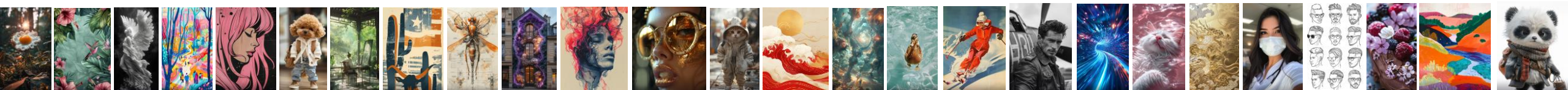
Сильные и слабые стороны DDPM

Плюсы:

- Высокое качество генерации (при достаточной настройке – лучше GANов)
- Flexibility

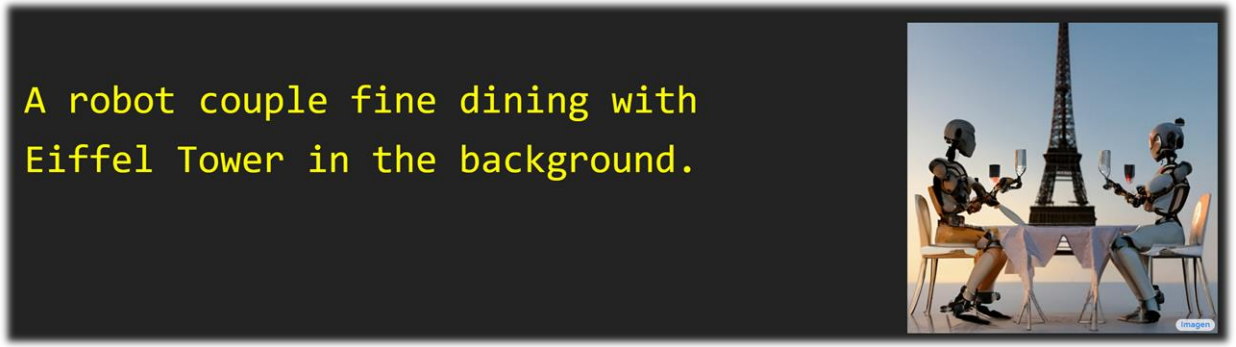
Минусы:

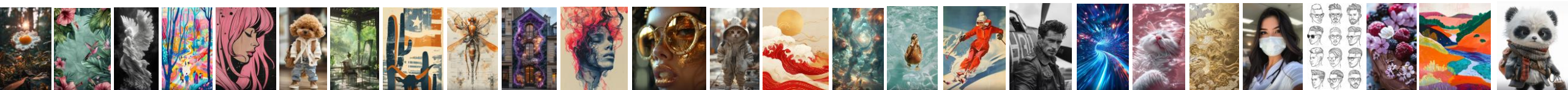
- Очень медленная генерация (за $O(T)$)
- По умолчанию нет условной генерации



Что дальше?

- Learnable variance => better results
- Conditional learning
- Cascaded diffusion (объединение нескольких диффузионок в пайплайн)
- CLIP + Diffusion model = DALL-E 2
- LLMs + Diffusion models (ImageGen)





Источники

- Denoising Diffusion Probabilistic Models (arxiv.org)
- Diffusion model - Wikipedia
- The DDPM Model | Daniel Gu (dg845.github.io)
- The Annotated Diffusion Model (huggingface.co)