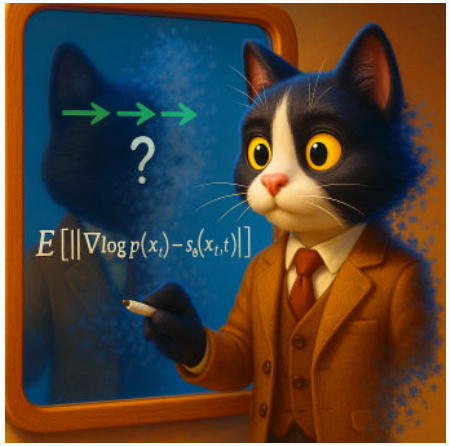
КАК РАБОТАЮТ ДИФФУЗИОННЫЕ МОДЕЛИ

Сергей Николенко

СПбГУ — Санкт-Петербург

20 ноября 2025 г.



Random facts:

- 20 ноября в ООН — Всемирный день ребёнка; в этот день в 1959 г. была принята «Декларация прав ребёнка», а в 1989 — «Конвенция прав ребёнка»
- 20 ноября 1805 г. венский мясник Иоганн Ланер изобрёл сосиски
- 20 ноября 1910 г. Франсиско Мадеро опубликовал «план Сан-Луис-Потоси», в котором объявлял результаты президентских выборов недействительными и призывал к борьбе с режимом Порфирио Диаса, правившего к тому времени 34 года, чем запустил Мексиканскую революцию; а в доме Ивана Озолина, начальника станции Астапово, в тот же день умер Лев Толстой
- 20 ноября 1947 г. в Вестминстерском аббатстве принцесса Елизавета вышла замуж за лейтенанта Филипа Маунтбэттена, который при этом стал герцогом Эдинбургским
- 20 ноября 1979 г. у стен Каабы появился человек и объявил, что настало время для исполнения древнего пророчества, во имя которого в пределах Запретной Мечети была пролита кровь; захват заложников длился до 4 декабря, погибло 255 человек
- 20 ноября 1985 г. вышла первая графическая операционная система от Microsoft, Windows 1.0

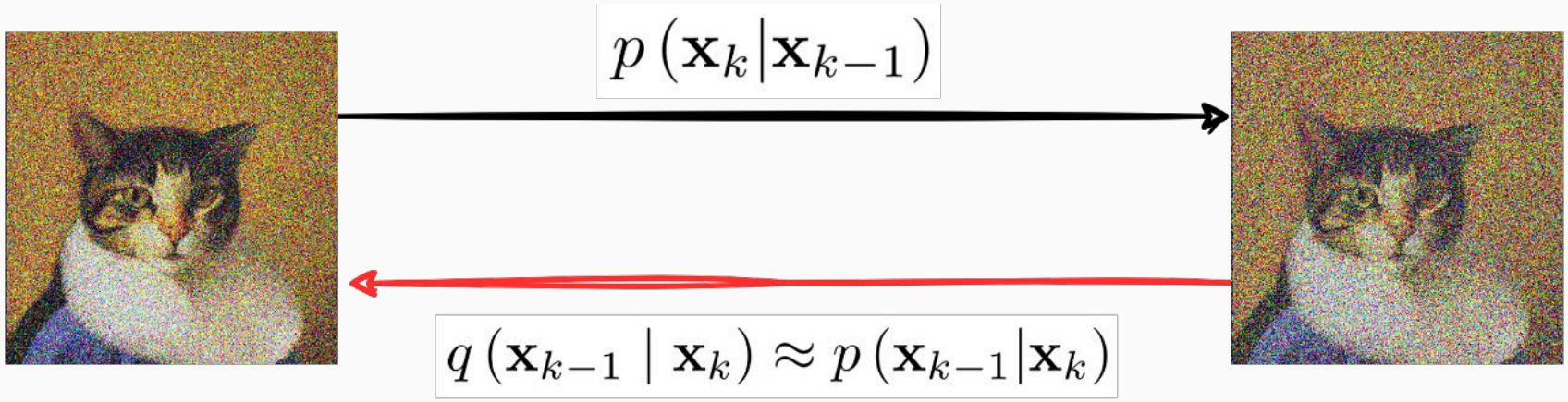
Диффузионные модели

Диффузионные модели

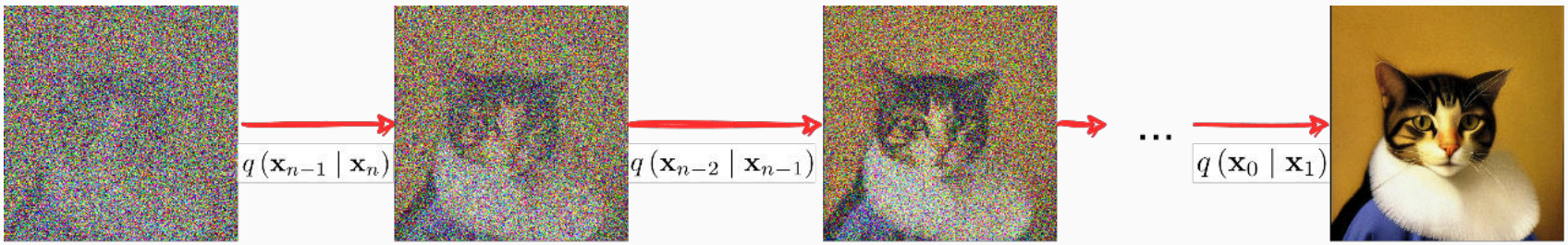
- (Sohl-Dickstein et al., 2015): Deep Unsupervised Learning using Nonequilibrium Thermodynamics
- Идея из статистической физики: давайте обучать марковскую цепь, которая постепенно сделает из случайного шума нужное распределение
- Для этого начнём с цепи, которая сделает из нужного распределения случайный шум:

Диффузионные модели

- А потом попробуем обратить каждый шаг:



- Если получится, то потом мы можем начать со случайного шума и двигаться обратно:



Диффузионные модели

- Forward diffusion: начнём с $p(\mathbf{x}_0)$ и будем добавлять гауссовский шум:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = N\left(\mathbf{x}_t \middle| \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right).$$

- Тогда постепенно $\mathbf{x}_0$ превращается в шум, скажем, за T шагов:

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

Диффузионные модели

- И для каждого t мы можем просэмплировать $\mathbf{x}_t$ в замкнутой форме через reparametrization trick: пусть $\epsilon \sim N(0, \mathbf{I})$ и обозначим $\alpha_t = 1 - \beta_t$, $A_t = \prod_{i=1}^{T} \alpha_i$; тогда

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon = \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon = \ldots$$

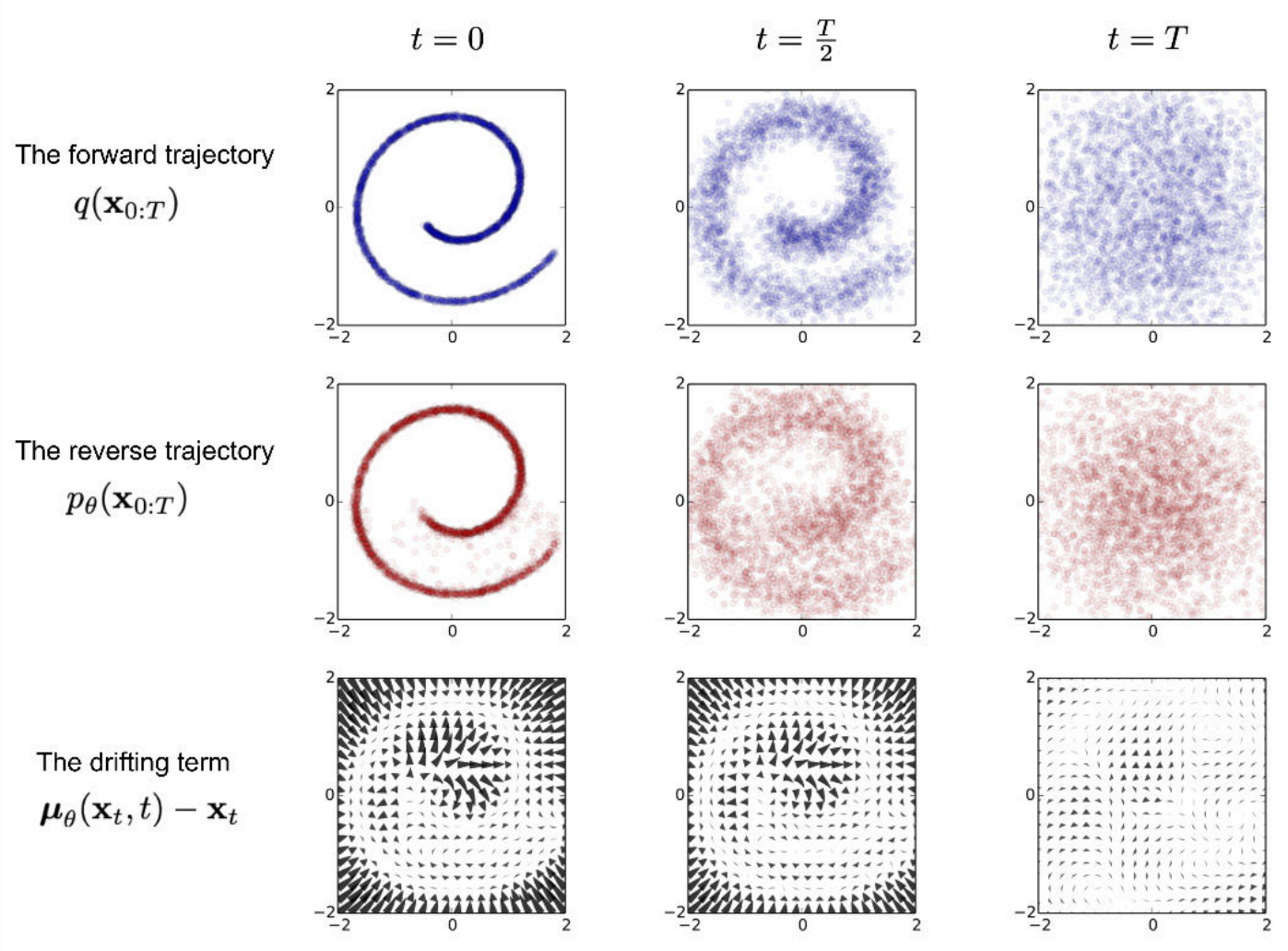
$$\ldots = \sqrt{A_t}\mathbf{x}_0 + \sqrt{1-A_t}\epsilon, \text{ то есть } q(\mathbf{x}_t \mid \mathbf{x}_0) = N\left(\mathbf{x}_t \middle| \sqrt{A_t}\mathbf{x}_0, (1-A_t)\mathbf{I}\right)$$

потому что когда мы складываем два гауссиана $N(0, \sigma_1^2\mathbf{I})$ и $N(0, \sigma_1^2\mathbf{I})$, получается гауссиан $N(0, (\sigma_1^2 + \sigma_2^2)\mathbf{I})$, и у нас это

$$(1-\alpha_t) + \alpha_t(1-\alpha_t) = 1 - \alpha_t\alpha_{t-1}.$$

Диффузионные модели

- А теперь задача, казалось бы, в том, чтобы обратить этот процесс, научиться сэмплировать из $q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$



Диффузионные модели

- Для маленьких β_t обратные распределения тоже будут гауссовскими, но параметры их теперь уже зависят от точек данных.
- Reverse diffusion: теперь уже надо обучать распределения

$$p_\theta\left(\mathbf{x}_{0:T}\right) = p_\theta\left(\mathbf{x}_T\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right), \quad \text{где}$$

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) = N\left(\mathbf{x}_{t-1} \middle| \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)\right).$$

Диффузионные модели

- Если бы мы знали $\mathbf{x}_0$, то обратить было бы можно аналитически, кстати:

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) = q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0\right) \frac{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_0\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)} \propto$$

$$\propto e^{-\frac{1}{2}\left(\frac{\left(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1}\right)^2}{\beta_t} + \frac{\left(\mathbf{x}_{t-1} - \sqrt{A_{t-1}}\mathbf{x}_0\right)^2}{1-A_{t-1}} + \frac{\left(\mathbf{x}_t - \sqrt{A_t}\mathbf{x}_0\right)^2}{1-A_t}\right)},$$

и в этой формуле теперь надо выделить полный квадрат по $\mathbf{x}_{t-1}$...

Диффузионные модели

- Получится

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) = N\left(\mathbf{x}_{t-1} \middle| \tilde{\mu}\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t \mathbf{I}\right), \quad \text{где}$$

$$\tilde{\beta}_t = \frac{1 - A_{t-1}}{1 - A_t} \cdot \beta_t,$$

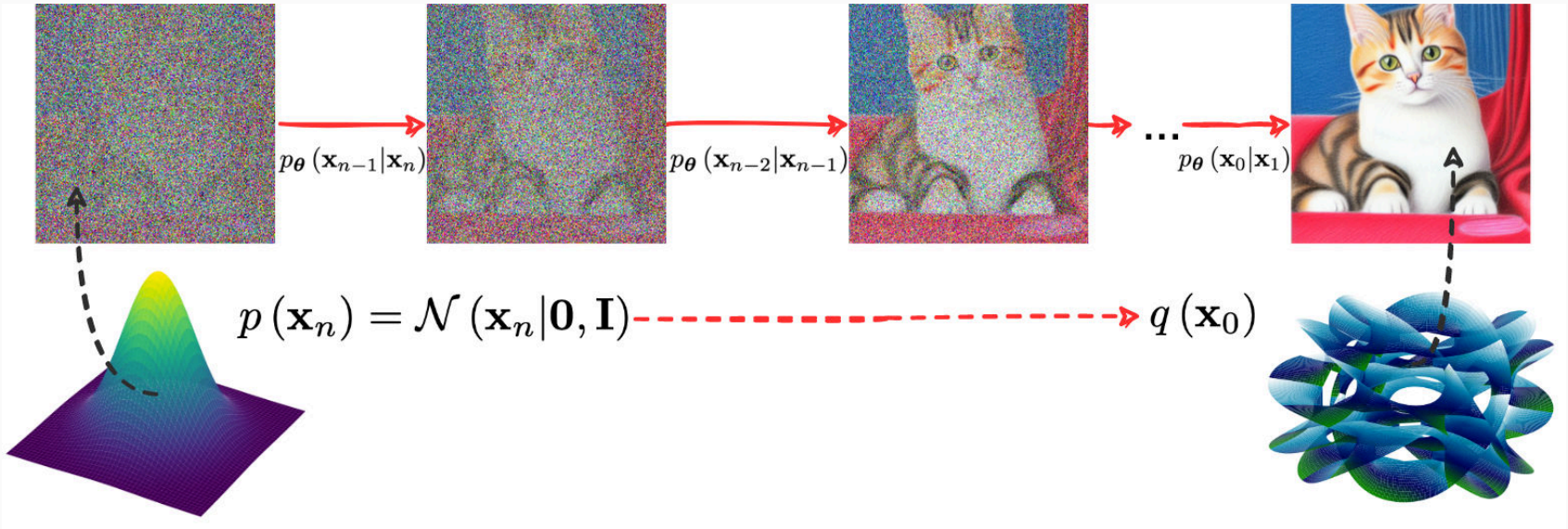
$$\tilde{\mu}\left(\mathbf{x}_t, \mathbf{x}_0\right) = \frac{\sqrt{\alpha_t}\left(1 - A_{t-1}\right)}{1 - A_t}\mathbf{x}_t + \frac{\sqrt{A_{t-1}}\beta_t}{1 - A_t}\mathbf{x}_0,$$

и поскольку мы знаем, что $\mathbf{x}_t = \sqrt{A_t}\mathbf{x}_0 + \sqrt{1 - A_t}\epsilon$,

$$\tilde{\mu}\left(\mathbf{x}_t, \mathbf{x}_0\right) = \frac{1}{\sqrt{A_t}}\left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - A_t}}\epsilon_t\right).$$

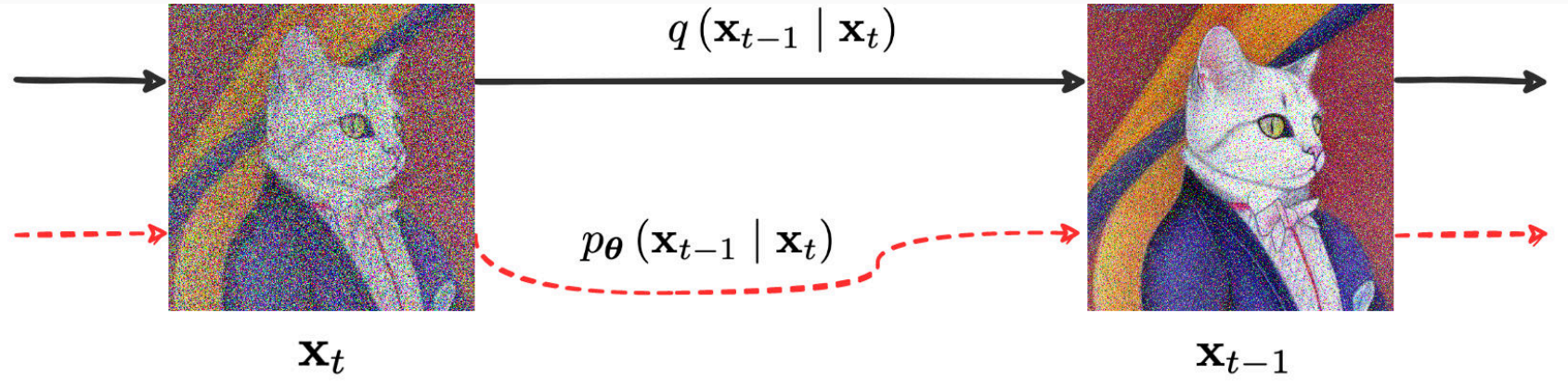
Диффузионные модели

- В конечном счёте хочется, чтобы обратная диффузия реконструировала $q(\mathbf{x}_0)$ из стандартного входа в $\mathbf{x}_n$; примерно так:



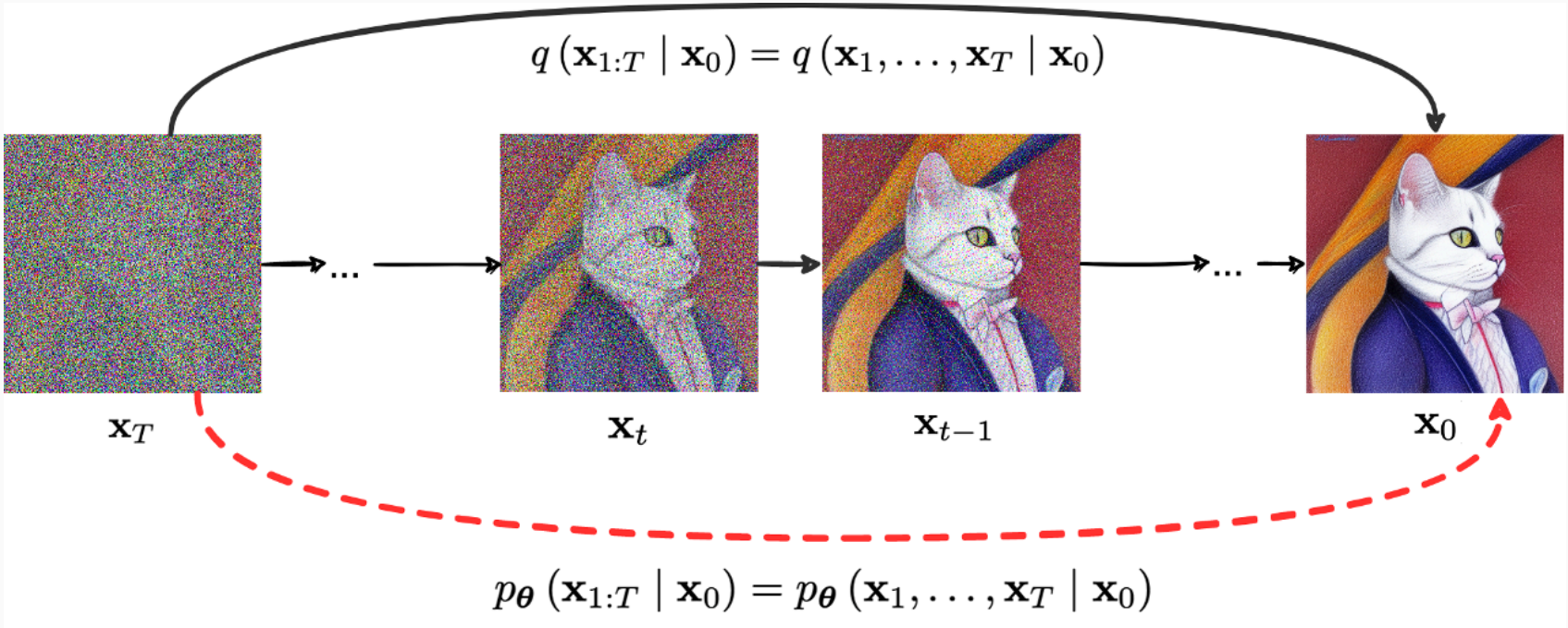
Диффузионные модели

- Что делать? Как всегда в байесовском выводе, приближать!
- На каждом шаге модель должна стать хорошим приближением к $q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$, без условия на неизвестное $\mathbf{x}_0$:



Диффузионные модели

- Чтобы получить эту аппроксимацию, нужна вариационная нижняя оценка, похожая на ту, которая была в VAE и DALL-E
- Начинаем с оценки на глобальное распределение $q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right) = q\left(\mathbf{x}_1, \ldots, \mathbf{x}_T \mid \mathbf{x}_0\right)$:



- А потом она разложится на оценки для отдельных шагов диффузионного процесса

Диффузионные модели

- Вспомним идею вариационных приближений:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x})\, p(\mathbf{z}|\mathbf{x}),$$

$$\log p(\mathbf{x}) = \log p(\mathbf{x}, \mathbf{z}) - \log p(\mathbf{z}|\mathbf{x}),$$

возьмём ожидание по $q(\mathbf{z})$:

$$\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x})] = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}, \mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z}|\mathbf{x})],$$

а затем сделаем стандартные преобразования:

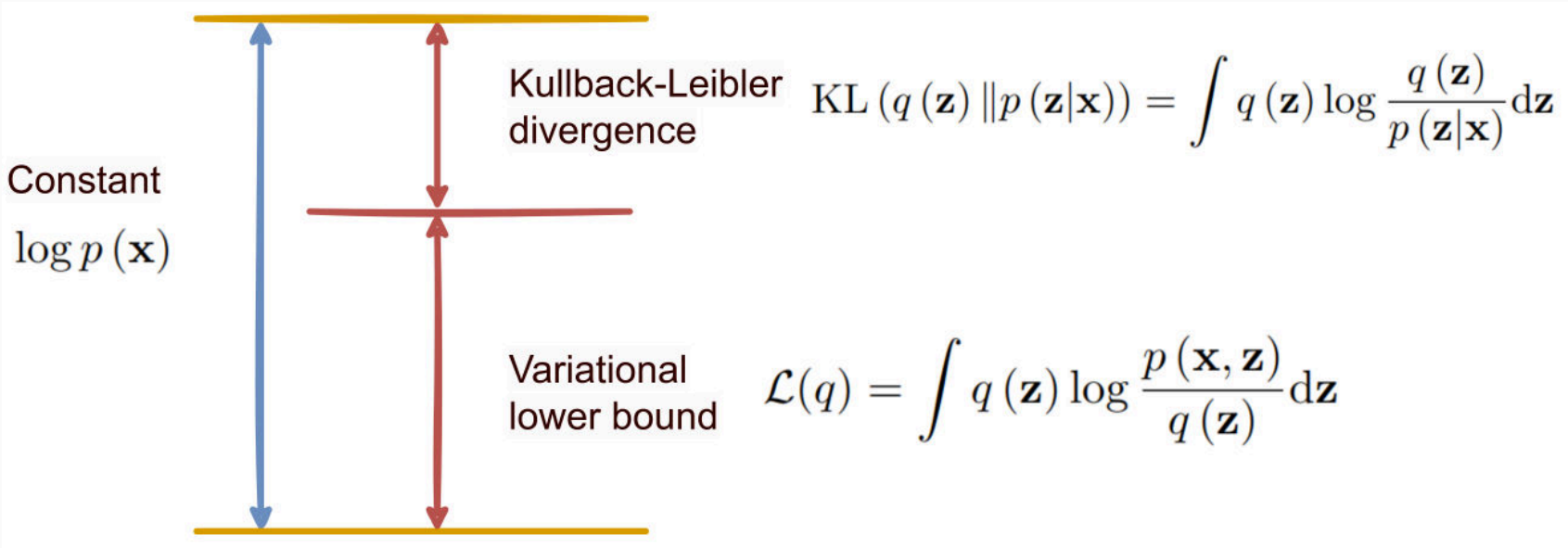
$$\begin{aligned}\log p(\mathbf{x}) =& \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}, \mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})] + \\ &+ \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z}|\mathbf{x})], \\ \log p(\mathbf{x}) =& \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \mathrm{d}\mathbf{z} + \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \mathrm{d}\mathbf{z}.\end{aligned}$$

Диффузионные модели

- Итого получается, что

$$\log p\left(\mathbf{x}\right) = L(q) + \mathrm{KL}\left(q\left(\mathbf{z}\right) \| p\left(\mathbf{z}|\mathbf{x}\right)\right), \quad \text{где}$$

$$L(q) = \int q\left(\mathbf{z}\right) \log \frac{p\left(\mathbf{x}, \mathbf{z}\right)}{q\left(\mathbf{z}\right)} \mathrm{d}\mathbf{z}.$$



Диффузионные модели

- Давайте для диффузионной модели выведем вариационную оценку снова из первых принципов:

$$\log p_\theta(\mathbf{x}_0) = \log p_\theta(\mathbf{x}_0, \dots, \mathbf{x}_T) - \log p_\theta(\mathbf{x}_1, \dots, \mathbf{x}_T \mid \mathbf{x}_0),$$

$$\mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)] = \mathbb{E}_{q(\mathbf{x}_{0:T})}[\log p_\theta(\mathbf{x}_{0:T})] - \mathbb{E}_{q(\mathbf{x}_{0:T})}[\log p_\theta(\mathbf{x}_{1:T} \mid \mathbf{x}_0)] =$$
$$= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right] + \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}{p_\theta(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right].$$

- Здесь второе слагаемое — это KL-дивергенция, и мы получаем функцию ошибки как минус вариационную нижнюю оценку:

$$L = \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \geq -\mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)].$$

Диффузионные модели

- Эту оценку можно подсчитать как сумму:

$$
\begin{aligned}
L =& \mathbb{E}_q \left[\log \frac{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{0:T}\right)} \right] = \mathbb{E}_q \left[\log \frac{\prod_{t=1}^{T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}{p_\theta\left(\mathbf{x}_T\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)} \right] \\
=& \mathbb{E}_q \left[-\log p_\theta\left(\mathbf{x}_T\right) + \sum_{t=1}^{T} \log \frac{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)} \right] \\
=& \mathbb{E}_q \left[-\log p_\theta\left(\mathbf{x}_T\right) + \sum_{t=2}^{T} \log \frac{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)} + \log \frac{q\left(\mathbf{x}_1 \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)} \right] \\
=& \mathbb{E}_q \left[-\log p_\theta\left(\mathbf{x}_T\right) + \sum_{t=2}^{T} \log \left(\frac{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)} \frac{q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)}{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_0\right)} \right) + \log \frac{q\left(\mathbf{x}_1 \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)} \right] \\
=& \mathbb{E}_q \left[-\log p_\theta\left(\mathbf{x}_T\right) + \sum_{t=2}^{T} \log \frac{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)} + \log \frac{q\left(\mathbf{x}_T \mid \mathbf{x}_0\right)}{q\left(\mathbf{x}_1 \mid \mathbf{x}_0\right)} + \log \frac{q\left(\mathbf{x}_1 \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)} \right] \\
=& \mathbb{E}_q \left[\log \frac{q\left(\mathbf{x}_T \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_T\right)} + \sum_{t=2}^{T} \log \frac{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)} - \log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right) \right].
\end{aligned}
$$

Диффузионные модели

- Итого получается

$$
\begin{aligned}
L =& L_T + L_{T-1} + ... + L_0, \quad \text{where} \\
L_T =& \mathrm{KL}\left(q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_T\right)\right), \\
L_t =& \mathrm{KL}\left(q\left(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_t \mid \mathbf{x}_{t+1}\right)\right), \quad t = 1, ..., T-1, \\
L_0 =& -\log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right).
\end{aligned}
$$

Диффузионные модели

- Всё это KL-дивергенции между гауссианами, их можно подсчитать и подставить в функцию ошибки
- Например, в L_t мы используем гауссовскую параметризацию

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)=N\left(\mathbf{x}_{t-1} \middle| \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)\right)$$

и пытаемся её параметры совместить с $q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0\right)$. Для среднего, например,

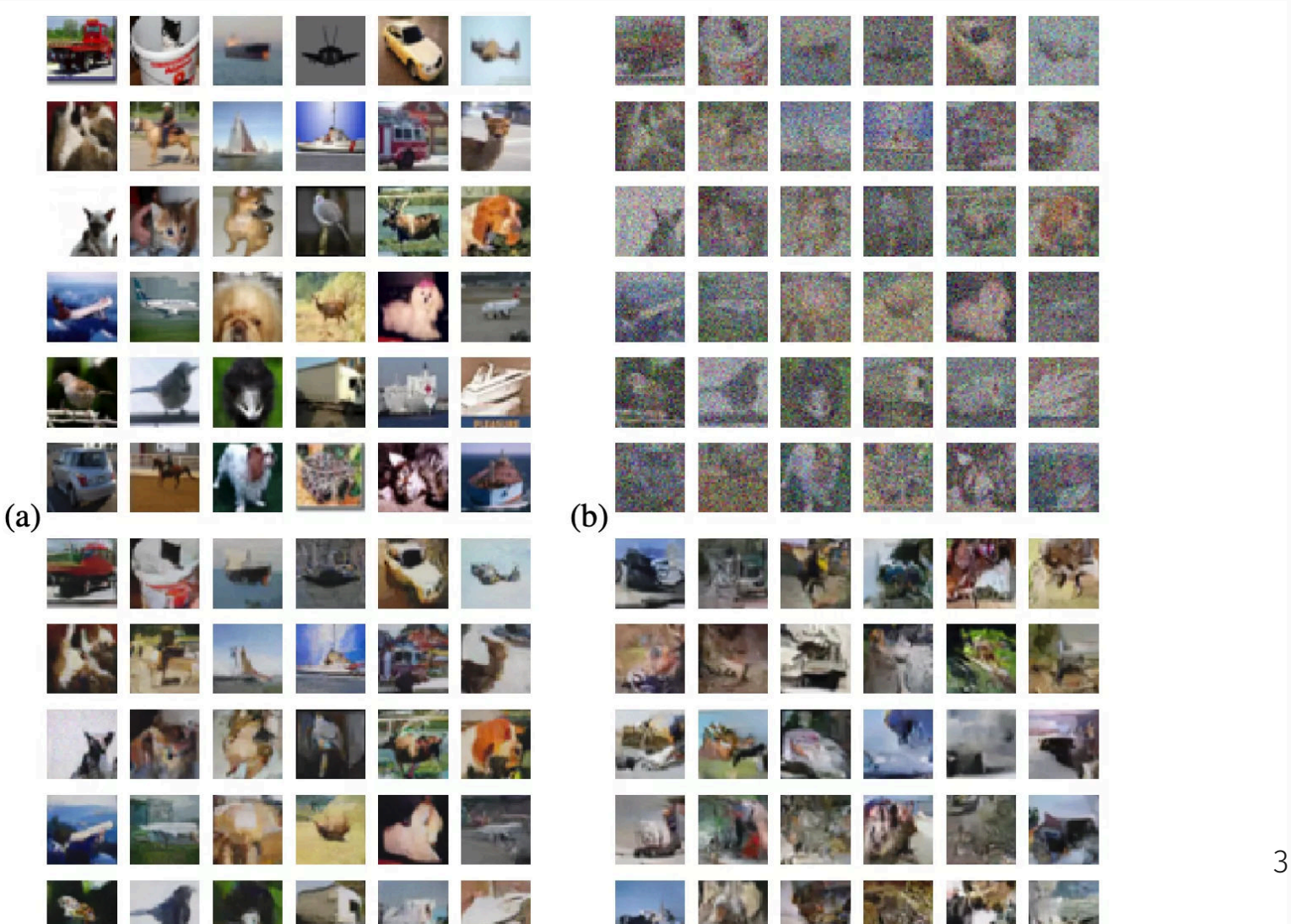
$$\mu_\theta(\mathbf{x}_t, t) \approx \tilde{\mu}_t\left(\mathbf{x}_t, \mathbf{x}_0\right)=\frac{1}{\sqrt{A_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-A_t}} \epsilon_t\right),$$

и поскольку мы знаем $\mathbf{x}_t$ при обучении, мы можем параметризовать шум напрямую:

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)=N\left(\mathbf{x}_{t-1} \middle| \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-A_t}} \epsilon_\theta(\mathbf{x}_t, t)\right), \Sigma_\theta(\mathbf{x}_t, t)\right).$$

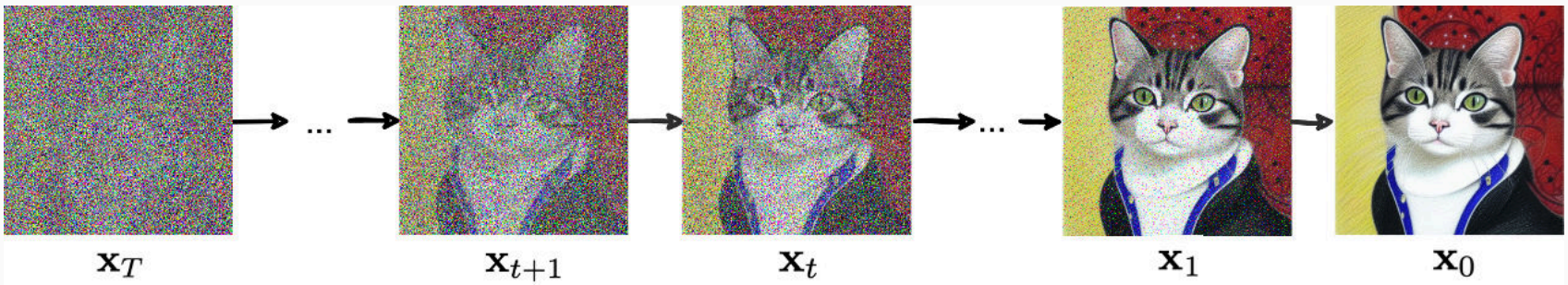
Диффузионные модели

- (Sohl-Dickstein et al., 2015) само по себе не слишком хорошо работало, разве что на CIFAR:



DDPM

- Следующий шаг: “Denoising Diffusion Probabilistic Models” (DDPM; Ho et al., 2020)
- Та же основная идея, та же структура функции ошибки:



DDPM

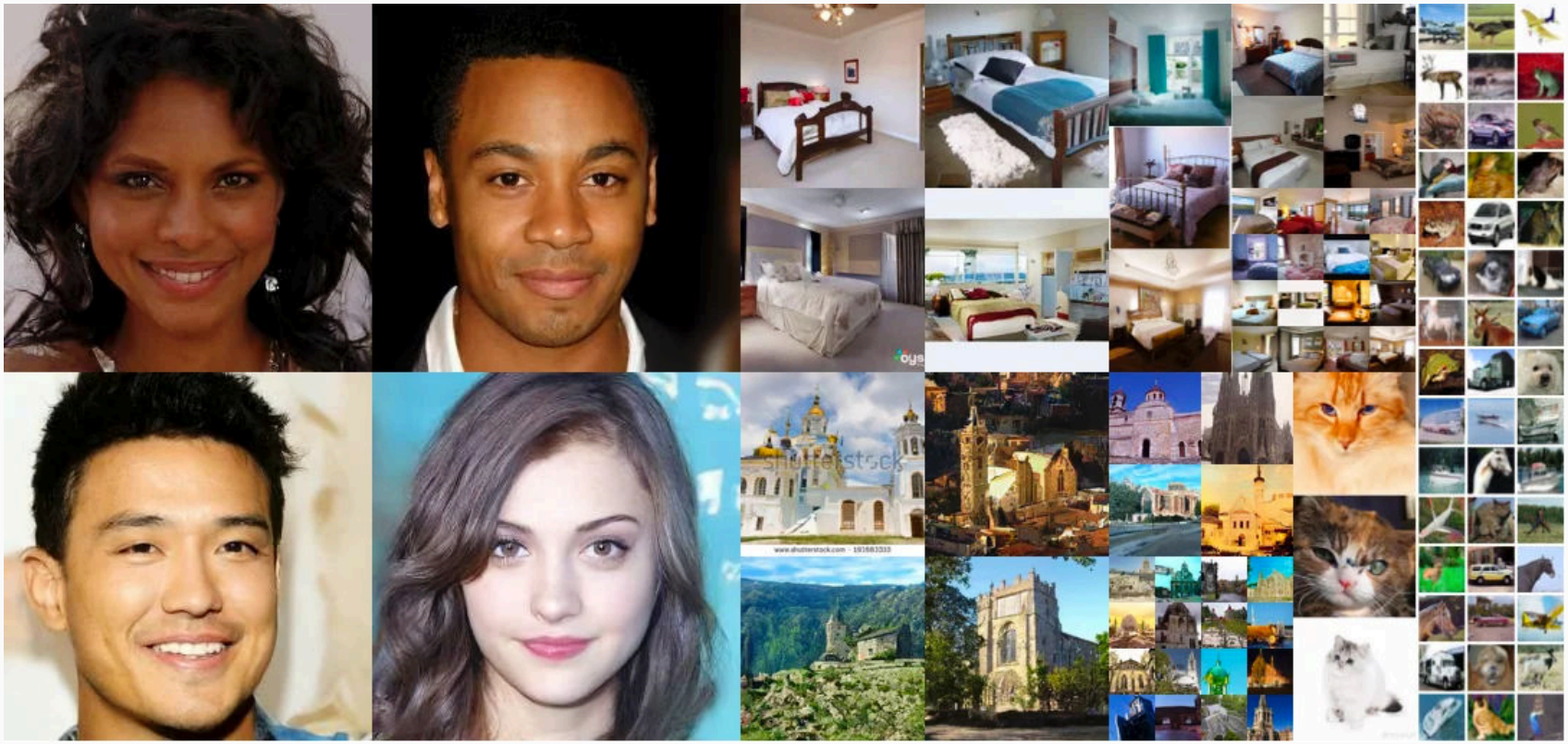
- Но дальше DDPM делает следующие замечания:
  - дисперсии прямой диффузии β_t — постоянные гиперпараметры, их не обучают, так что все q не обучаются; т.к. $p_\theta(\mathbf{x}_T)$ — фиксированное распределение, из которого мы будем сэмплировать, L_T является константой;
  - для промежуточных шагов дисперсии в $p_\theta(\mathbf{x}_t \mid \mathbf{x}_{t+1})$ тоже не обучаются, и можно найти простую замкнутую форму для L_t;
  - главное — давайте введём отдельный дискретный декодер для L_0; если данные состоят из целых чисел от 0 до 255, отмасштабированных в $[-1, 1]$ (это естественное представление для картинок), DDPM моделирует

$$p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1) = \prod_{i=1}^{D} \int_{\delta_-(x_0,i)}^{\delta_+(x_0,i)} N\left(x \middle| \mu_{\theta,i}(\mathbf{x}_1), \sigma_1^2\right) \mathrm{d}x,$$

    где i идёт по пикселям, $\mu_\theta(\mathbf{x}_1)$ — декодер, пределы интегрирования $\frac{1}{255}$ в каждую сторону от $x_{0,i}$
- Т.е. можно подставить другую модель на последнем шаге и использовать $\mu_\theta(\mathbf{x}_1)$ без шума во время сэмплирования

DDPM

- DDPM уже давала отличное порождение, сравнимое с лучшими GAN'ами того времени:

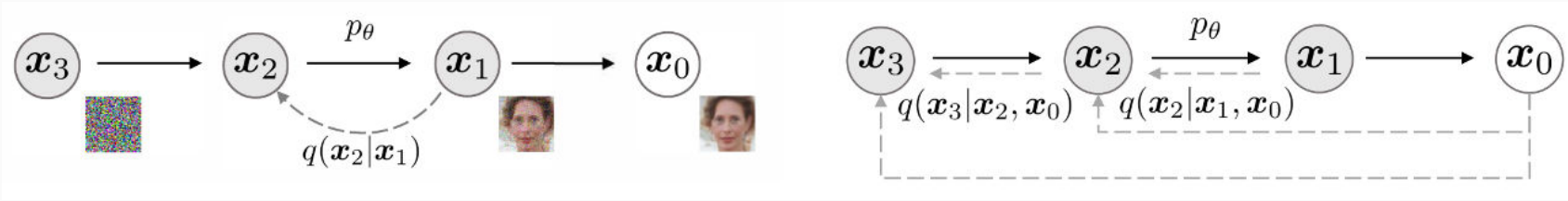


DDIM

- Следующий шаг пришёл очень скоро: “Denoising Diffusion Implicit Models” (DDIM; Song et al., 2020)
- Важный недостаток всех диффузионных моделей пока что в том, что они *очень медленные*
- Порождение должно пройти по всем шагам диффузии, а их буквально тысячи, и они идут последовательно, не параллелизуются
- Song et al. упоминают, что сэмплирование из обученного GAN примерно в 1000 раз быстрее, чем сэмплирование из DDPM для того же размера картинок

DDIM

- Как можно ускорить диффузионные модели?
- Song et al. обобщают диффузионные модели и в частности DDPM
- Функция ошибки L не зависит напрямую от совместного распределения $q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)$, а только от маргиналов $q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)$
- Это значит, что мы можем переиспользовать ту же функцию ошибки для другого совместного распределения, если у него те же маргиналы
- В частности, обобщить можно даже на немарковские диффузионные процессы, если для них получится найти обратную марковскую цепь:



Вывод DDIM

- Зададим вероятностную модель для данных $\mathbf{x}_0$ (из распределения $q(\mathbf{x}_0)$) через латентные переменные $\mathbf{x}_1, \dots, \mathbf{x}_T$
- Сначала прямой проход (noising) $q_\sigma(\mathbf{x}_{1:T} \mid \mathbf{x}_0)$, в котором к данным постепенно добавляется шум:

$$q_\sigma(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = q_\sigma(\mathbf{x}_T \mid \mathbf{x}_0) \prod_{t=2}^{T} q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0).$$

- Здесь мы по определению задаём, как связаны $\mathbf{x}_1, \dots, \mathbf{x}_T$ с $\mathbf{x}_0$
- Важный момент: переходы $q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$ зависят и от $\mathbf{x}_t$, и от $\mathbf{x}_0$, т.е. процесс $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_T$ не марковский! это осознанное решение

Вывод DDIM

- Мы выбираем следующие распределения (с тем же маргиналом, что и в DDPM):

$$q_\sigma(\mathbf{x}_T \mid \mathbf{x}_0) = \mathcal{N}(\sqrt{A_T}\,\mathbf{x}_0,\ (1 - A_T)\,\mathbf{I}),$$

$$q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\sqrt{A_{t-1}}\mathbf{x}_0 + \sqrt{1 - A_{t-1} - \sigma_t^2}\,\frac{\mathbf{x}_t - \sqrt{A_t}\mathbf{x}_0}{\sqrt{1 - A_t}},\ \sigma_t^2\,\mathbf{I}\right).$$

- σ_t^2 – дисперсия добавочного шума в обратном переходе; форма среднего такая для того, чтобы маргиналы $q_\sigma(\mathbf{x}_t \mid \mathbf{x}_0)$ оставались гауссианами и совпадали с DDPM

Вывод DDIM

- Ключевое утверждение: для любого t

$$q_\sigma(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\sqrt{A_t}\,\mathbf{x}_0,\ (1 - A_t)\,\mathbf{I}\big).$$

- Эквивалентно можно записать в виде «одноступенчатой» формулы:

$$\mathbf{x}_t = \sqrt{A_t}\,\mathbf{x}_0 + \sqrt{1 - A_t}\,\epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

- Это ключевая формула DDPM/DDIM: любое зашумлённое состояние $\mathbf{x}_t$ можно получить напрямую из $\mathbf{x}_0$, не прогоняя цепочку шаг за шагом

Вывод DDIM

- Доказательство по индукции. Предположим, что

$$q_\sigma(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\sqrt{A_t}\,\mathbf{x}_0,\ (1 - A_t)\,\mathbf{I}), \quad \text{то есть}$$

$$\mathbf{x}_t = \sqrt{A_t}\,\mathbf{x}_0 + \sqrt{1 - A_t}\,\epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

- Подставим это в $q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$, получим

$$q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\sqrt{A_{t-1}}\mathbf{x}_0 + \sqrt{1 - A_{t-1} - \sigma_t^2}\,\epsilon_t,\ \sigma_t^2\,\mathbf{I}\right).$$

- Мы просто заменили $\dfrac{\mathbf{x}_t - \sqrt{A_t}\mathbf{x}_0}{\sqrt{1 - A_t}}$ на ϵ_t, используя выражение для $\mathbf{x}_t$ через $\mathbf{x}_0$ и ϵ_t; среднее стало линейной функцией $\mathbf{x}_0$ и ϵ_t

Вывод DDIM

- Используем факт про гауссианы: если

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \mu,\ \Lambda^{-1}),$$

$$p(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}(\mathbf{y} \mid A\mathbf{x} + \mathbf{b},\ L^{-1}), \quad \text{то}$$

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} \mid A\mu + \mathbf{b},\ L^{-1} + A\Lambda^{-1}A^{\top}).$$

- В нашем случае $\mathbf{x}$ – это ϵ_t, распределённый как $\mathcal{N}(\mathbf{0}, \mathbf{I})$; $\mathbf{y}$ – это $\mathbf{x}_{t-1}$; матрица A и сдвиг $\mathbf{b}$ задаются коэффициентами при ϵ_t и $\mathbf{x}_0$

Вывод DDIM

- Сопоставим параметры:

$$\mu \equiv \mathbf{0}, \quad \Lambda^{-1} \equiv \mathbf{I},$$

$$\mathbf{A} \equiv \sqrt{1 - A_{t-1} - \sigma_t^2}\,\mathbf{I}, \quad \mathbf{b} \equiv \sqrt{A_{t-1}}\,\mathbf{x}_0, \quad L^{-1} \equiv \sigma_t^2\,\mathbf{I}.$$

- Подставив в формулу, получим

$$q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_0) = \mathcal{N}\big(A\mu + \mathbf{b},\, L^{-1} + A\Lambda^{-1}A^\top\big) = \mathcal{N}\big(\sqrt{A_{t-1}}\mathbf{x}_0,\, (1 - A_{t-1})\,\mathbf{I}\big),$$

потому что

$$A\mu + \mathbf{b} = \sqrt{A_{t-1}}\,\mathbf{x}_0,$$

$$L^{-1} + A\Lambda^{-1}A^\top = \sigma_t^2\,\mathbf{I} + (1 - A_{t-1} - \sigma_t^2)\,\mathbf{I} = (1 - A_{t-1})\,\mathbf{I}.$$

Вывод DDIM

- Из полученного выражения

$$\mathbf{x}_t = \sqrt{A_t}\,\mathbf{x}_0 + \sqrt{1 - A_t}\,\epsilon_t$$

можно выразить $\mathbf{x}_0$ через $\mathbf{x}_t$ и ϵ_t:

$$\mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1 - A_t}\,\epsilon_t}{\sqrt{A_t}}.$$

- Эта формула важна для порождения, т.к. в обратном процессе мы будем оценивать ϵ_t нейросетью и тем самым получать оценку $\mathbf{x}_0$ из $\mathbf{x}_t$

Вывод DDIM

- Поскольку истинный шум ϵ_t нам неизвестен, мы вводим аппроксимацию

$$\epsilon_\theta^{(t)}(\mathbf{x}_t) \approx \epsilon_t,$$

  где $\epsilon_\theta^{(t)}$ – нейросеть, зависящая от $\mathbf{x}_t$ и (неявно) от шага t

- Тогда оценка $\mathbf{x}_0$ на шаге t задаётся как

$$f_\theta^{(t)}(\mathbf{x}_t) \equiv \frac{\mathbf{x}_t - \sqrt{1 - A_t}\,\epsilon_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{A_t}}.$$

- Обратное выражение:

$$\epsilon_\theta^{(t)}(\mathbf{x}_t) = \frac{\mathbf{x}_t - \sqrt{A_t}\,f_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{1 - A_t}}.$$

Вывод DDIM

- Теперь переходим к обратному (порождающему) процессу
- Совместное распределение модели задаётся как

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

  где обычно берут

$$p_\theta(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

  а переходы $p_\theta^{(t)}$ параметризуются через нейросеть
- Правдоподобие данных:

$$\log p(\mathbf{x}_0) = \log \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$$

Вывод DDIM

- Введём вспомогательное распределение – наш прямой процесс $q_\sigma(\mathbf{x}_{1:T} \mid \mathbf{x}_0)$, умножим и поделим на него внутри интеграла и применим неравенство Йенсена:

$$\begin{aligned}\log p(\mathbf{x}_0) &= \log \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T} = \log \int \frac{p_\theta(\mathbf{x}_{0:T})}{q_\sigma(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\, q_\sigma(\mathbf{x}_{1:T} \mid \mathbf{x}_0)\, d\mathbf{x}_{1:T} \\ &= \log \mathbb{E}_{q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\frac{p_\theta(\mathbf{x}_{0:T})}{q_\sigma(\mathbf{x}_{1:T} \mid \mathbf{x}_0)} \right] \\ &\geq \mathbb{E}_{q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log p_\theta(\mathbf{x}_{0:T}) - \log q_\sigma(\mathbf{x}_{1:T} \mid \mathbf{x}_0) \right].\end{aligned}$$

- Обозначим $J_\sigma = -\mathbb{E}_{q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log p_\theta(\mathbf{x}_{0:T}) - \log q_\sigma(\mathbf{x}_{1:T} \mid \mathbf{x}_0) \right]$, подставим факторизации p_θ и q_σ:

$$q_\sigma(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = q_\sigma(\mathbf{x}_T \mid \mathbf{x}_0) \prod_{t=2}^{T} q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0),$$

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

Вывод DDIM

- Тогда после раскрытия логарифмов получаем

$$J_\sigma = \mathbb{E}_{q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0)} \Bigg[\log q_\sigma(\mathbf{x}_T \mid \mathbf{x}_0) + \sum_{t=2}^{T} \log q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) - \sum_{t=1}^{T} \log p_\theta^{(t)}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) - \log p_\theta(\mathbf{x}_T) \Bigg].$$

- Следующий шаг – перегруппировать всё это так, чтобы получилось что-то похожее на сумму KL-дивергенций, плюс отдельное слагаемое для реконструкции $\mathbf{x}_0$

Вывод DDIM

- После некоторых мучений получим:

$$J_\sigma = \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \Big[D_{\mathrm{KL}}\big(q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta^{(t)}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\big) \Big]$$
$$- \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \big[\log p_\theta^{(0)}(\mathbf{x}_0 \mid \mathbf{x}_1) \big].$$

- Здесь $p_\theta^{(0)}(\mathbf{x}_0 \mid \mathbf{x}_1)$ – отдельный шаг, как в DDPM; в реальных реализациях его часто опускают или сливают с общей ошибкой

Вывод DDIM

- Параметризация обратного шага и KL-дивергенция: для $t > 1$ возьмём

$$p_\theta^{(t)}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\Big(\sqrt{A_{t-1}}\, f_\theta^{(t)}(\mathbf{x}_t) + \sqrt{1 - A_{t-1} - \sigma_t^2}\, \epsilon_\theta^{(t)}(\mathbf{x}_t),\, \sigma_t^2\, \mathbf{I}\Big).$$

- Это та же структура, что и у $q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$, только вместо истинного $\mathbf{x}_0$ и истинного шума ϵ_t мы подставляем оценки $f_\theta^{(t)}(\mathbf{x}_t)$ и $\epsilon_\theta^{(t)}(\mathbf{x}_t)$
- KL между двумя гауссианами с одинаковой ковариацией имеет особенно простую форму – это просто квадратичная функция разности средних

Вывод DDIM

- У нас:

$$q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\Big(\sqrt{A_{t-1}}\mathbf{x}_0 + \sqrt{1 - A_{t-1} - \sigma_t^2}\,\epsilon_t,\ \sigma_t^2\,\mathbf{I}\Big),$$

$$p_\theta^{(t)}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\Big(\sqrt{A_{t-1}}\,f_\theta^{(t)}(\mathbf{x}_t) + \sqrt{1 - A_{t-1} - \sigma_t^2}\,\epsilon_\theta^{(t)}(\mathbf{x}_t),\ \sigma_t^2\,\mathbf{I}\Big).$$

- Для двух гауссианов с одинаковой ковариацией $\Sigma = \sigma_t^2\mathbf{I}$ KL-дивергенция равна

$$\mathrm{KL}\big(\mathcal{N}(\mu_q, \Sigma) \,\|\, \mathcal{N}(\mu_p, \Sigma)\big) = \frac{1}{2}\,(\mu_q - \mu_p)^\top \Sigma^{-1} (\mu_q - \mu_p).$$

- Подставляя наши средние и $\Sigma^{-1} = \sigma_t^{-2}\mathbf{I}$, получаем

$$D_{\mathrm{KL}}\big(q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta^{(t)}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\big) =$$
$$= \frac{1}{2\sigma_t^2}\Big\|\sqrt{A_{t-1}}(\mathbf{x}_0 - f_\theta^{(t)}(\mathbf{x}_t)) + \sqrt{1 - A_{t-1} - \sigma_t^2}(\epsilon_t - \epsilon_\theta^{(t)}(\mathbf{x}_t))\Big\|^2.$$

Вывод DDIM

- Используя связь

$$\mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1 - A_t}\,\epsilon_t}{\sqrt{A_t}}, \qquad f_\theta^{(t)}(\mathbf{x}_t) = \frac{\mathbf{x}_t - \sqrt{1 - A_t}\,\epsilon_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{A_t}},$$

можно показать (после упрощений), что KL-член пропорционален

$$D_{\mathrm{KL}}(\cdot) = \frac{\gamma_t}{2\sigma_t^2}\left\|\epsilon_t - \epsilon_\theta^{(t)}(\mathbf{x}_t)\right\|^2 + \text{const},$$

где γ_t – некоторая известная функция A_t и σ_t^2, которую можно заранее посчитать по расписанию шумов, как вес для шага t

- Главный вывод: оптимизация KL между истинным и параметризованным обратным переходом эквивалентна MSE между истинным шумом ϵ_t и предсказанным шумом $\epsilon_\theta^{(t)}(\mathbf{x}_t)$, с некоторым весом, зависящим от t

Вывод DDIM

- Если собрать все шаги t и учесть ожидание по $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ и $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$, то окончательная функция ошибки (с весами) имеет вид

$$\mathbb{E}_{t,\mathbf{x}_0,\epsilon_t}\left[\frac{\gamma_t}{2\sigma_t^2}\left\|\epsilon_t - \epsilon_\theta^{(t)}(\mathbf{x}_t)\right\|^2\right].$$

- На практике, как и в DDPM, обычно игнорируют веса $\frac{\gamma_t}{2\sigma_t^2}$, беря более простой loss

$$L_{\text{simple}} = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta^{(t)}(\mathbf{x}_t)\|^2\right],$$

где t выборочно сэмплируется из $\{1, \dots, T\}$

Вывод DDIM

- Итого обучение идёт точно как в DDPM: берём $\mathbf{x}_0$, сэмплируем $\mathbf{x}_t$, обновляем параметры сети по ошибке $L = \left\| \epsilon - \epsilon_\theta^{(t)}(\mathbf{x}_t) \right\|$
- А вот inference после обучения имеет вид

$$p_\theta^{(t)}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mu_\theta(\mathbf{x}_t, t),\ \sigma_t^2\, \mathbf{I}\right),$$

где среднее

$$\mu_\theta(\mathbf{x}_t, t) = \sqrt{A_{t-1}}\, f_\theta^{(t)}(\mathbf{x}_t) + \sqrt{1 - A_{t-1} - \sigma_t^2}\, \epsilon_\theta^{(t)}(\mathbf{x}_t)$$

- Если мы явно подставим $f_\theta^{(t)}(\mathbf{x}_t)$ через $\epsilon_\theta^{(t)}(\mathbf{x}_t)$, то получится более удобная форма:

$$\mathbf{x}_{t-1} = \sqrt{A_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{1 - A_t}\, \epsilon_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{A_t}} \right) + \sqrt{1 - A_{t-1} - \sigma_t^2}\, \epsilon_\theta^{(t)}(\mathbf{x}_t) + \sigma_t \epsilon$$

где $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ – новый случайный шум

Вывод DDIM

- Часто это записывают как

$$\mathbf{x}_{t-1} = \underbrace{\sqrt{A_{t-1}}\hat{\mathbf{x}}_0^{(t)}}_{\text{проекция предсказанного } \mathbf{x}_0} + \underbrace{\sqrt{1 - A_{t-1} - \sigma_t^2}\,\epsilon_\theta^{(t)}(\mathbf{x}_t)}_{\text{направление к текущему } \mathbf{x}_t} + \underbrace{\sigma_t\,\epsilon_t}_{\text{случайный шум}},$$

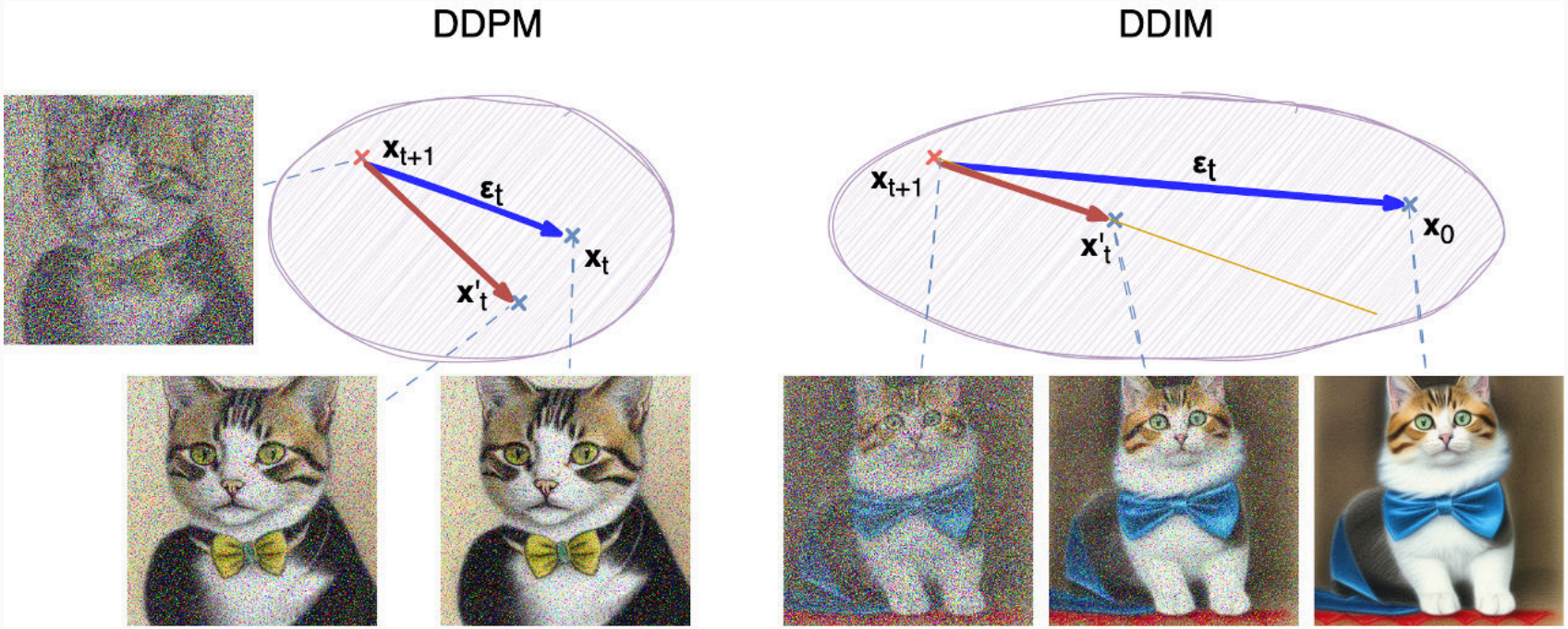
  где $\hat{\mathbf{x}}_0^{(t)} = f_\theta^{(t)}(\mathbf{x}_t)$.
- Получается, что первый член возвращает нас к предсказанной версии $\mathbf{x}_0$, второй корректирует направление в пространстве к текущему $\mathbf{x}_t$, третий добавляет случайность
- В DDIM можно взять $\sigma_t = 0$ и получить детерминированный обратный процесс; это тоже очень важное преимущество

Вывод DDIM

- Но главное, конечно, в том, что теперь можно перепрыгивать через некоторые шаги!
- В DDIM можно рассмотреть подмножество шагов $\tau = \{\tau_1, \dots, \tau_S\} \subset \{1, \dots, T\}$, по которым мы действительно делаем обратные переходы, и делать всё то же, но по τ
- Интуитивно говоря, мы сохраняем маргинал $q(\mathbf{x}_t \mid \mathbf{x}_0)$ для всех t, но обратный процесс будем реализовывать только на подмножестве τ, перепрыгивая через остальные шаги
- Таким образом можно тысячу шагов превратить в примерно пятьдесят

DDIM

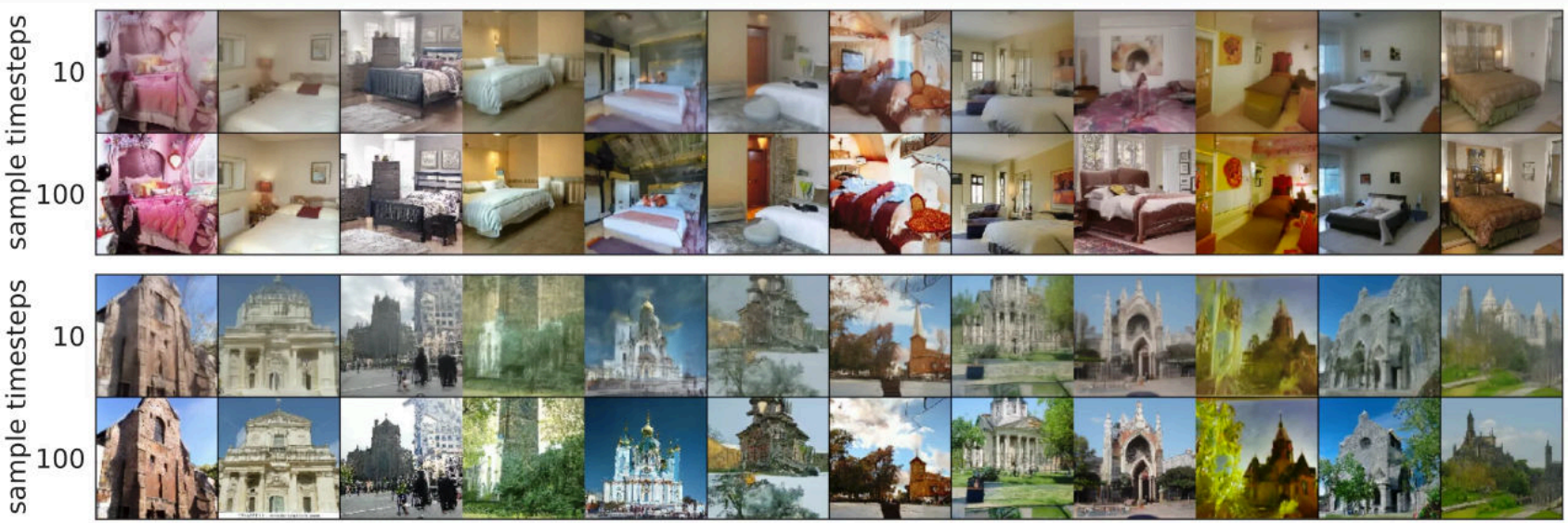
- Иллюстрация:



- Сложно сказать, какое приближение лучше, но теперь мы разделили зависимости ϵ_t от $\mathbf{x}_0$, и мы можем перепрыгнуть на несколько шагов сразу, двигаясь от $\mathbf{x}_t$ к $\mathbf{x}_{t+k}$ за дин шаг с увеличенным ϵ!

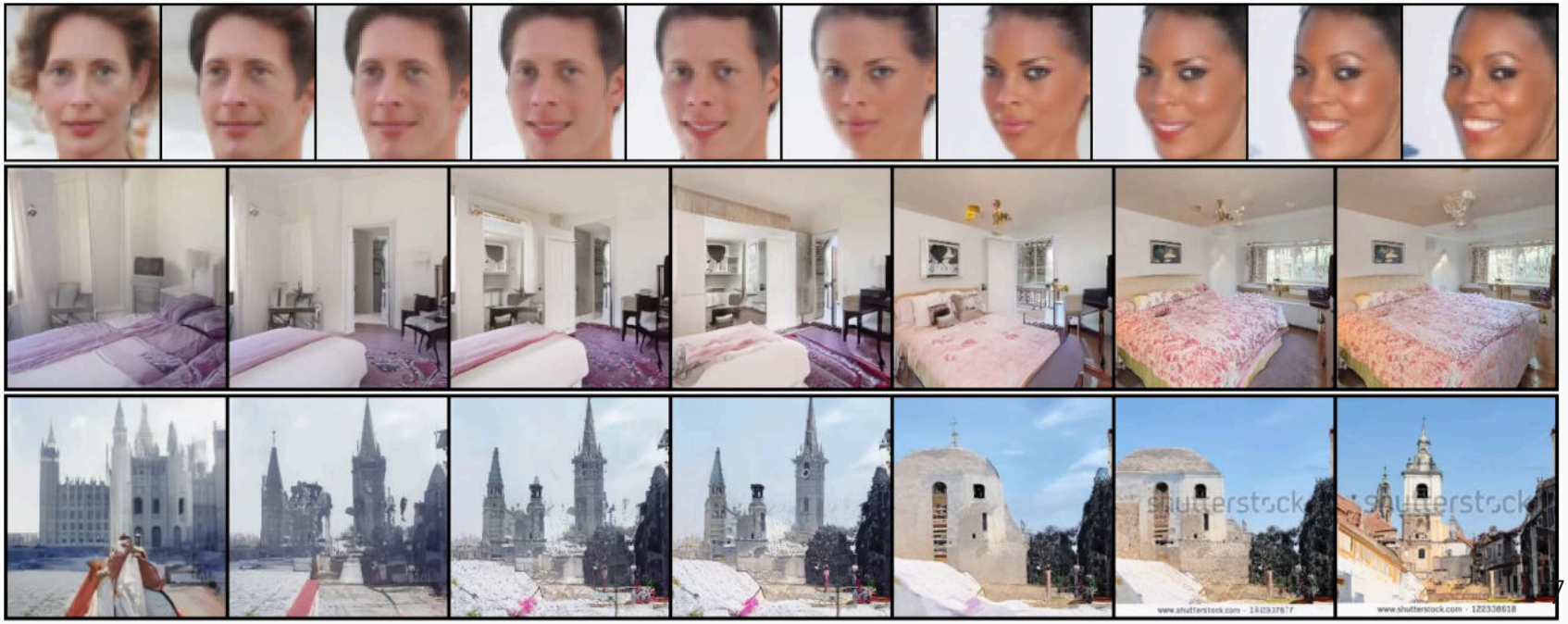
DDIM

- Это привело к 10x–100x ускорению по сравнению с DDPM без потери качества:



DDIM

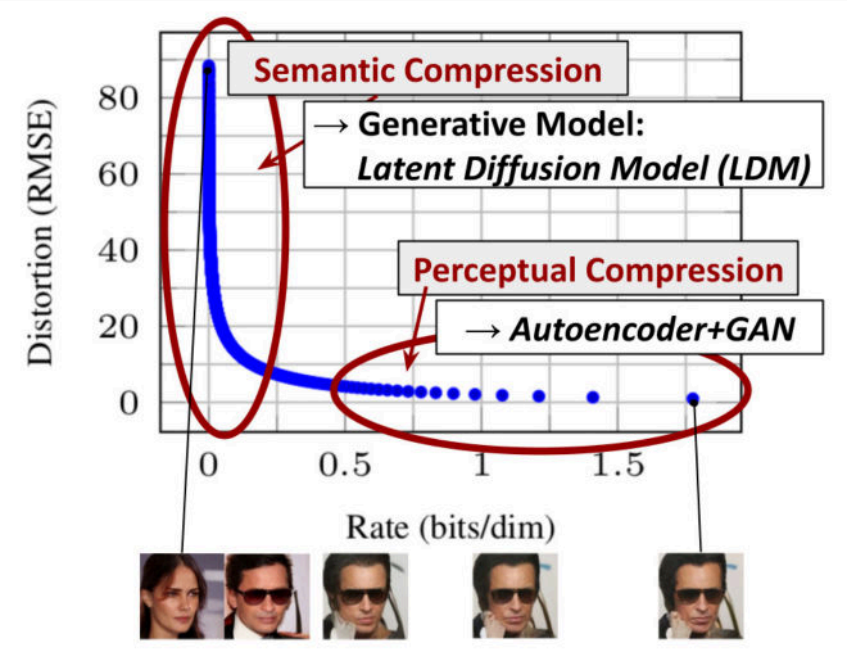
- Порождение в DDIMs тоже не обязано быть стохастическим; можно дисперсию обратной диффузии установить в ноль при порождении, $\sigma \to 0$
- И теперь латентный код в пространстве $\mathbf{x}_T$ соответствует ровно одной картинке, а DDIM — хорошая модель для латентных представлений; вот, например, интерполяции в латентном пространстве:



Stable Diffusion

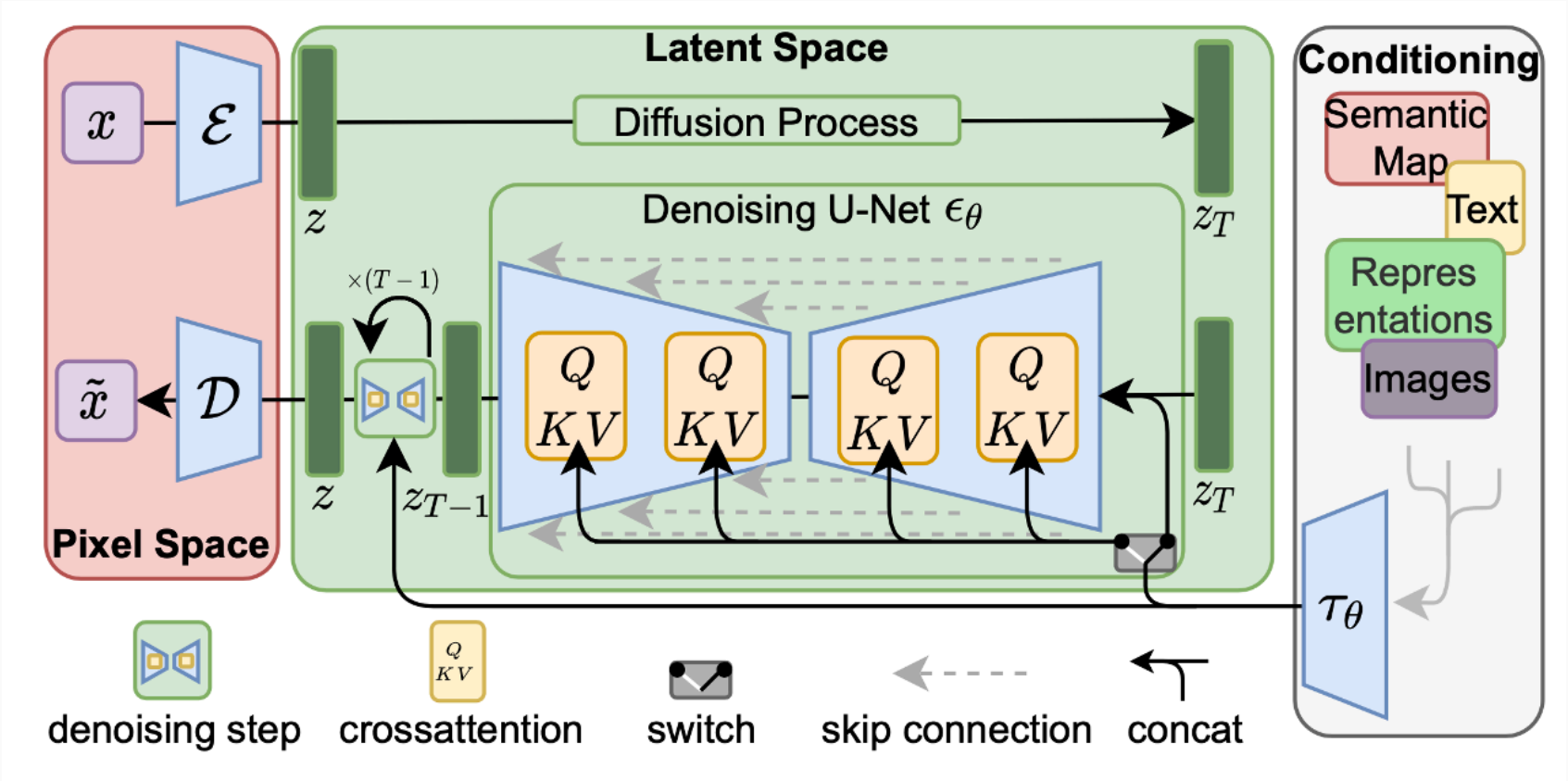
Stable Diffusion

- Ещё один прорыв пришёл тогда, когда стали использовать этот диффузионный процесс *в латентном пространстве*
- Stable Diffusion (Rombach, Blattmann et al., 2022):



Stable Diffusion

- Stable Diffusion (Rombach, Blattmann et al., 2022):

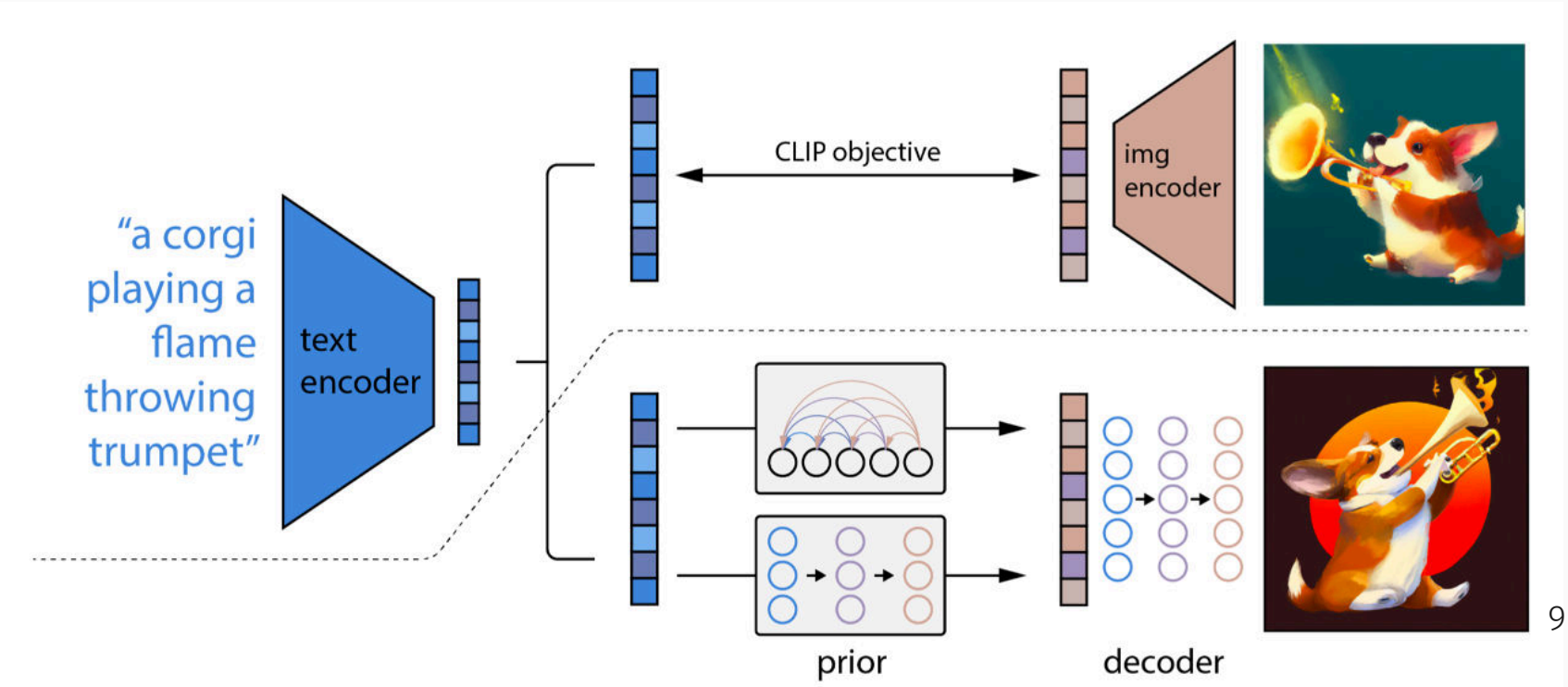


Stable Diffusion

- unCLIP, он же DALL-E 2 (Ramesh et al., 2022), обучает

$$p\left(\mathbf{x}|y\right) = p\left(\mathbf{x}|\mathbf{c}, y\right) p\left(\mathbf{c}|y\right),$$

где prior model $p\left(\mathbf{c}|y\right)$ строит CLIP embedding по тексту, а в $p\left(\mathbf{x}|\mathbf{c}, y\right)$ сначала порождается «шум» (prior) для картинки, а потом по ней диффузионная модель рисует саму картинку:



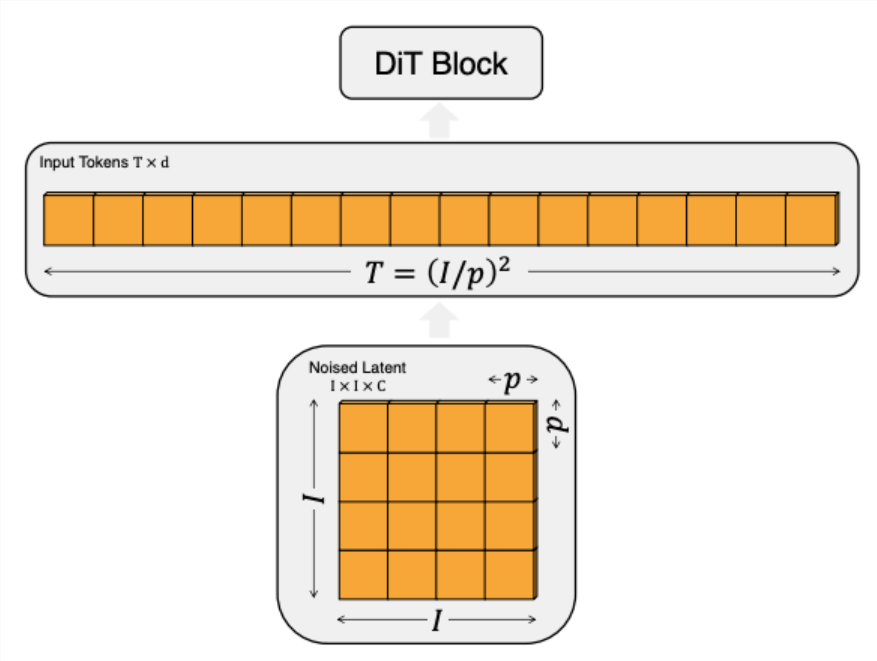
DALL-E 2

Midjourney

Diffusion Transformers

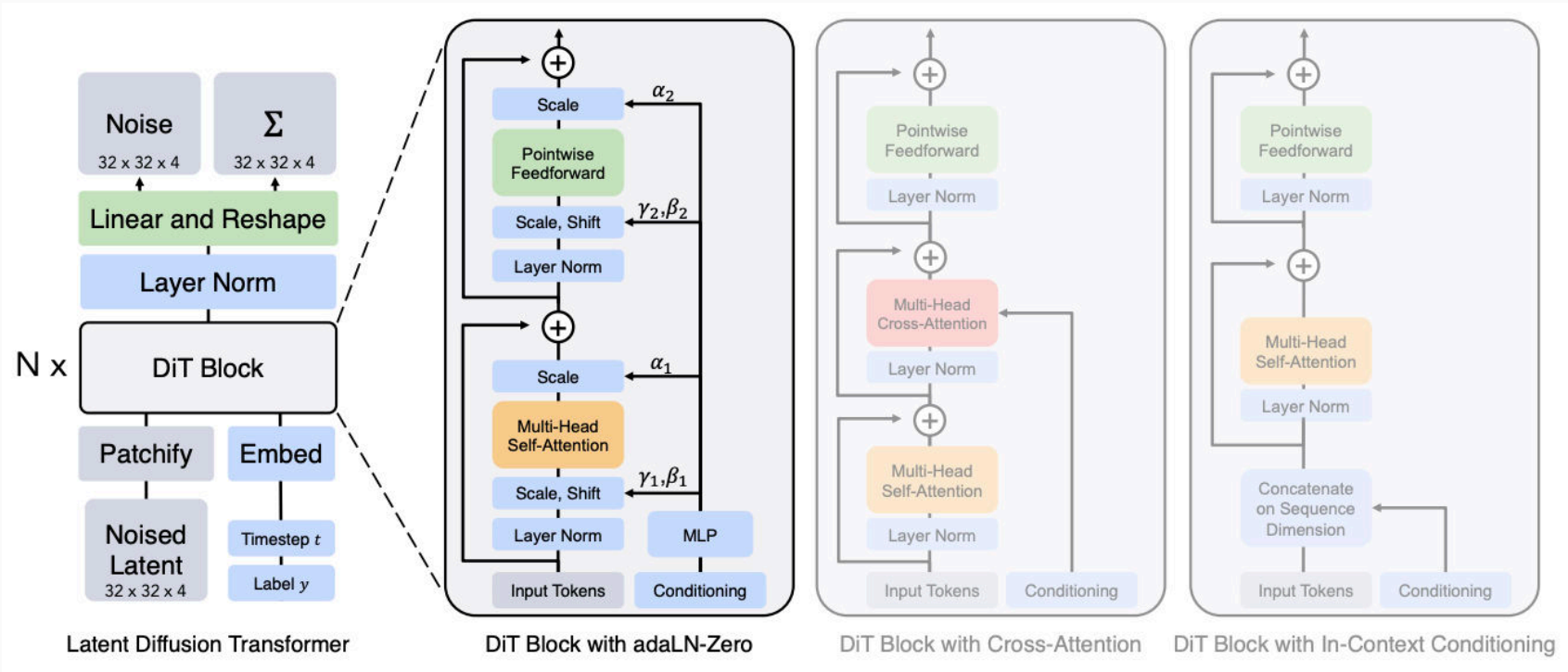
Diffusion Transformers

- Уже вполне современная архитектура – DiT (Peebles, Xie, 2022)
- Это модель латентной диффузии, но вместо U-Net-подобной архитектуры для латентных кодов используют трансформер
- Вход трансформера – тензор $I \times I \times C$, "patchified" в последовательность $p \times p$ патчей длины $T = (I/p)^2$; p – гиперпараметр, квадратично влияющий на сложность



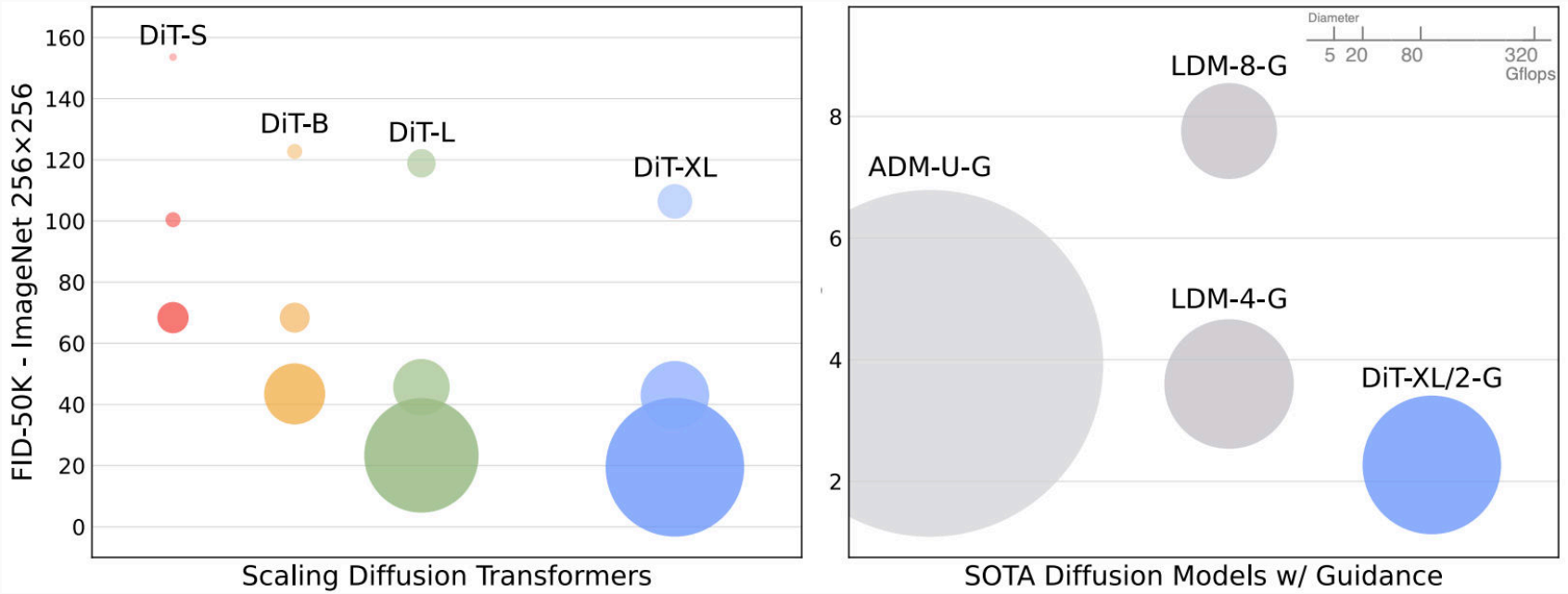
Diffusion Transformers

- Затем его обрабатывают блоками DiT; интересная часть здесь – adaptive layer norm с параметрами, взятыми из условия



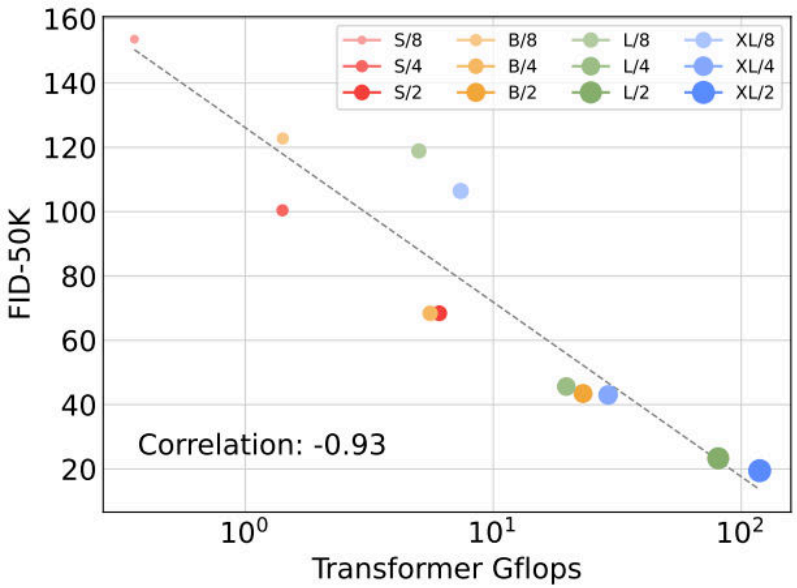
Diffusion Transformers

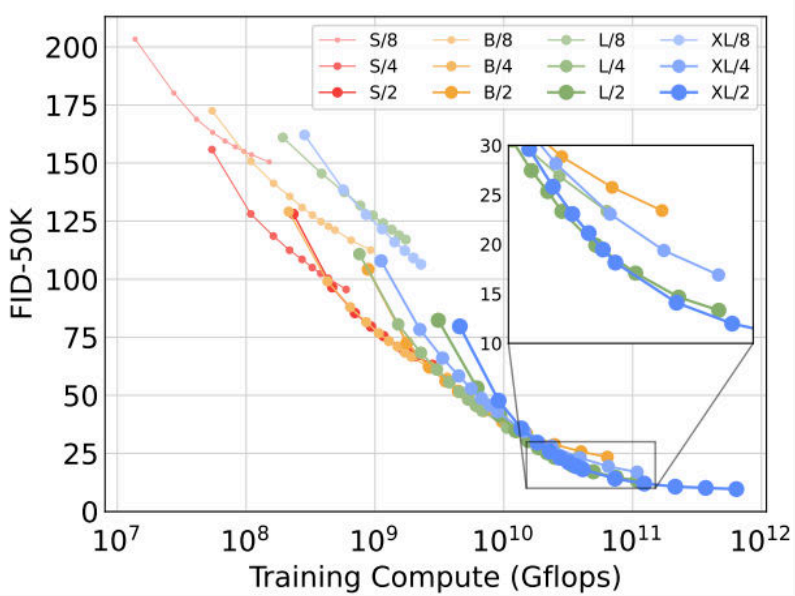
- Получается, что DiT куда более вычислительно эффективны:



Diffusion Transformers

- Transformer GFlops сильно скоррелированы с качеством, а большие модели DiT используют вычисления более эффективно:





Diffusion Transformers

- Хорошие сэмплы:

Flux

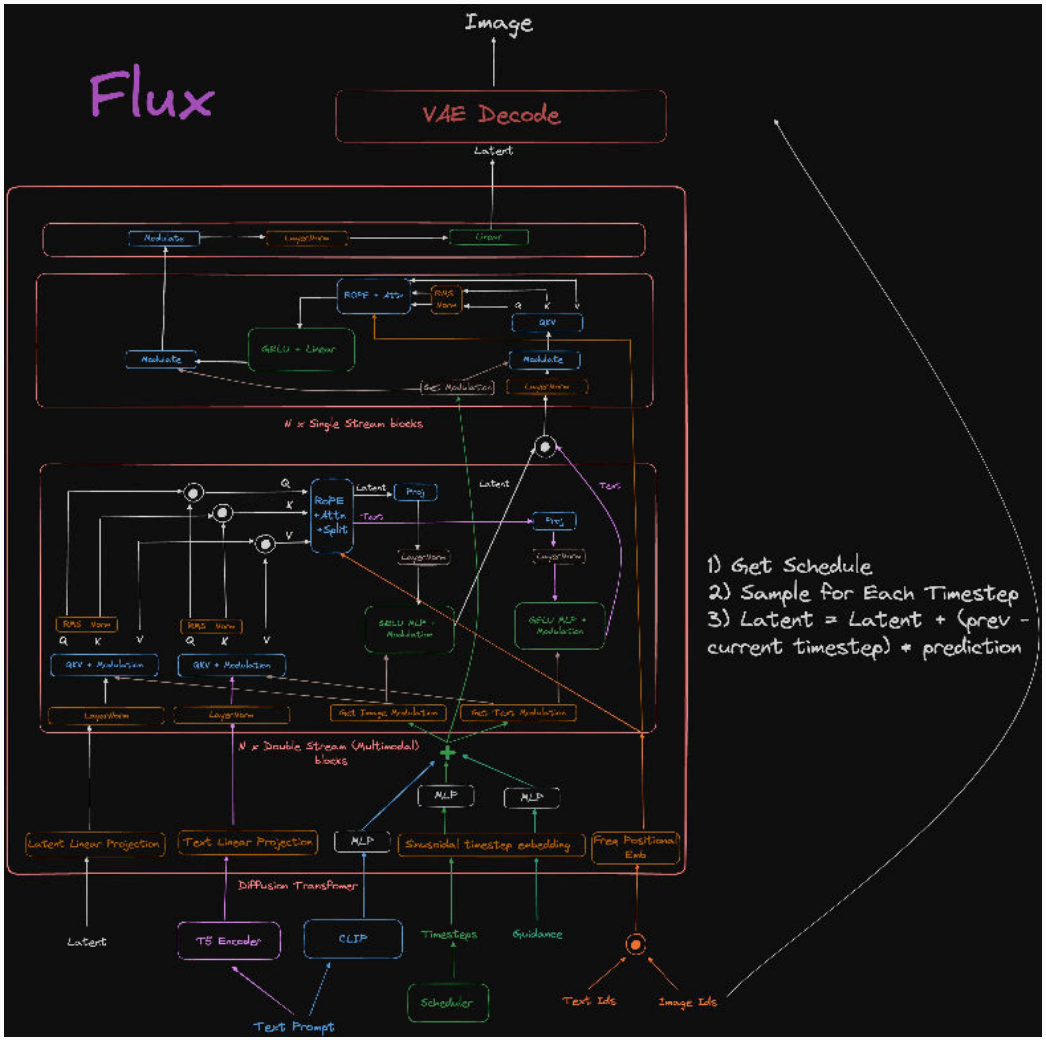
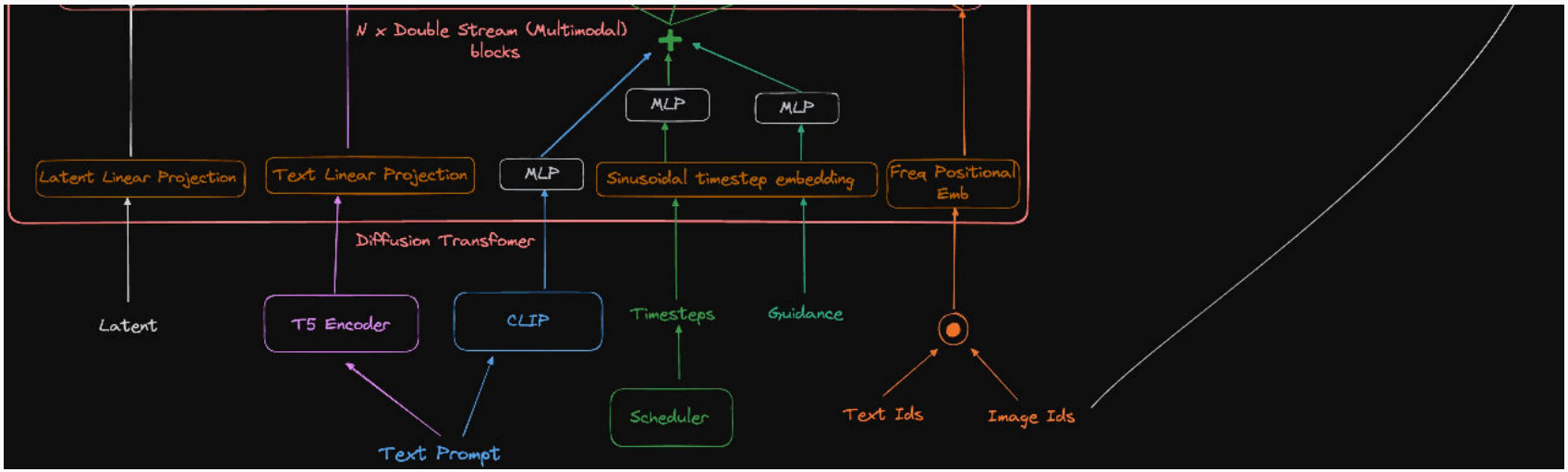
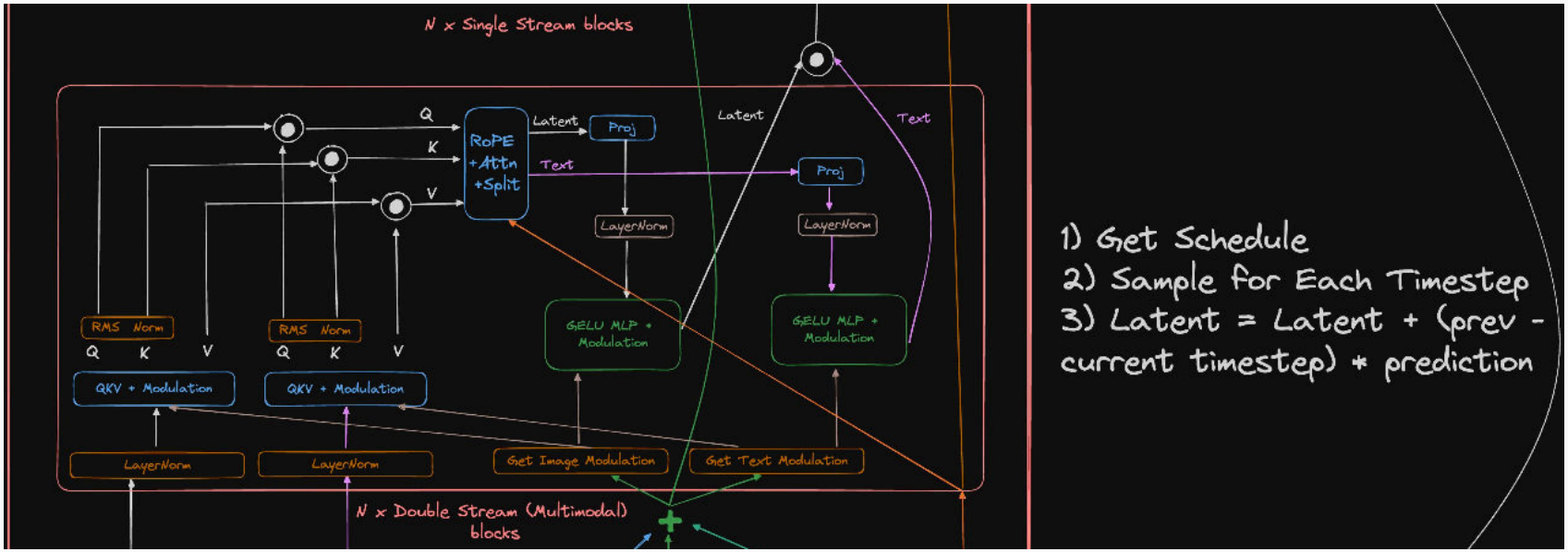


Image
Flux
VAE Decode
Latent
N x Single Stream blocks
RoPE +Attn +Split
Latent
Text
N x Double Stream (Multimodal) blocks
MLP
Latent Linear Projection
Text Linear Projection
Sinusoidal timestep embedding
Diffusion Transformer
Latent
T5 Encoder
CLIP
Timesteps
Guidance
Scheduler
Text Prompt
Text Ids
Image Ids
1) Get Schedule
2) Sample for Each Timestep
3) Latent = Latent + (prev - current timestep) * prediction

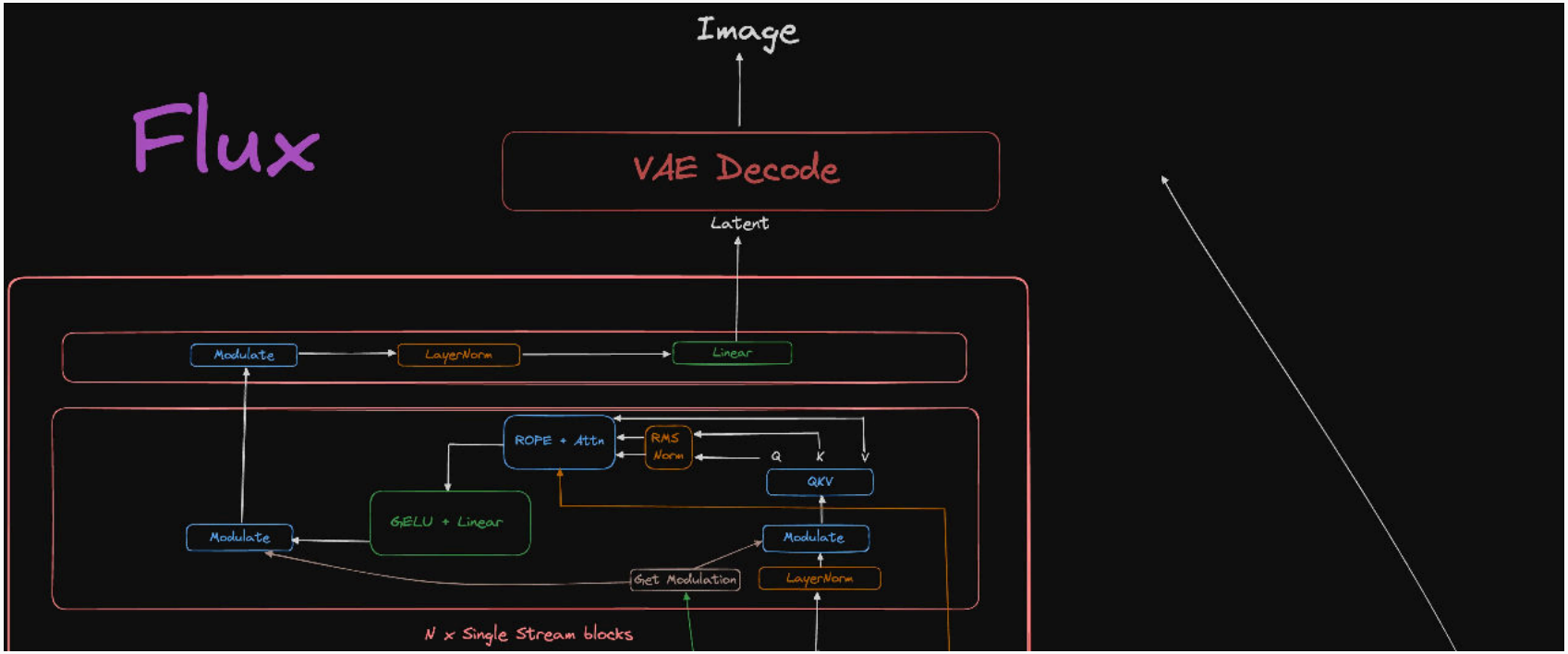
Flux


N x Double Stream (Multimodal) blocks
MLP
MLP
Latent Linear Projection
Text Linear Projection
MLP
Sinusoidal timestep embedding
Freq Positional Emb
Diffusion Transfomer
Latent
T5 Encoder
CLIP
Timesteps
Guidance
Scheduler
Text Ids
Image Ids
Text Prompt

Flux


N x Single Stream blocks
Q
K
V
RoPE
+Attn
+Split
Latent
Proj
Text
Proj
LayerNorm
LayerNorm
Latent
Text
RMS Norm
Q K V
RMS Norm
Q K V
GELU MLP + Modulation
GELU MLP + Modulation
QKV + Modulation
QKV + Modulation
LayerNorm
LayerNorm
Get Image Modulation
Get Text Modulation
N x Double Stream (Multimodal) blocks
1) Get Schedule
2) Sample for Each Timestep
3) Latent = Latent + (prev - current timestep) * prediction

Flux


Flux
Image
VAE Decode
Latent
Modulate
LayerNorm
Linear
ROPE + Attn
RMS Norm
Q
K
V
QKV
GELU + Linear
Modulate
Modulate
Get Modulation
LayerNorm
N x Single Stream blocks

Flux

Flux

Flux

Flux

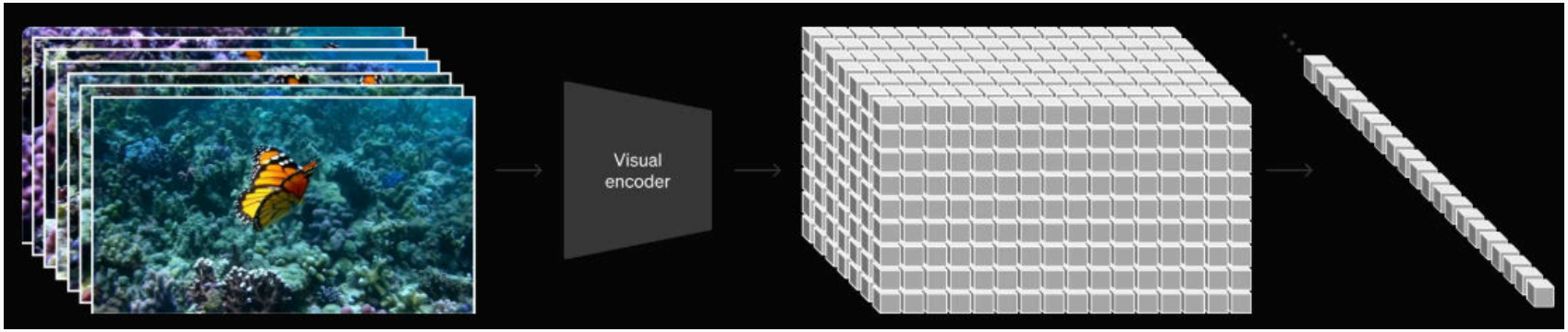
Flux

OpenAI Sora

- Текущее применение DiT: OpenAI Sora (Feb 2024, и вот ещё совсем недавно)

OpenAI Sora

- OpenAI всех деталей не раскрывает, но там точно есть DiT:



От дискретных процессов к непрерывным

От DDIM к непрерывному времени

- Начнём с идеи DDIM: вместо шага $\mathbf{x}_t \to \mathbf{x}_{t-1}$ мы приближаем направление сразу к $\mathbf{x}_0$.
- Пусть

$$\alpha_t = 1 - \beta_t, \quad A_t = \prod_{i=1}^{t} \alpha_i, \quad \hat{\mathbf{x}}_0(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{1 - A_t}\, \epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{A_t}}.$$

- Один шаг DDIM с параметром $\eta = 0$:

$$\mathbf{x}_{t-1} = \sqrt{A_{t-1}}\, \hat{\mathbf{x}}_0(x_t, t) + \sqrt{1 - A_{t-1}}\, \hat{\epsilon}(\mathbf{x}_t, t).$$

- Интуитивно это значит, что мы движемся по «детерминированной» траектории от $\mathbf{x}_t$ к $\mathbf{x}_0$.

От DDIM к непрерывному времени

- Дальше переходим к *непрерывному времени*: время $t \in [0, T]$, шаги $\Delta t \to 0$.
- Будем описывать прямую диффузию как стохастическое дифференциальное уравнение (SDE); обратный процесс — как *обратную* SDE.
- Зашумление — это стохастическое дифференциальное уравнение

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t, \quad \mathbf{f}(\mathbf{x}, t) = -\tfrac{1}{2}\beta(t)\,\mathbf{x}, \quad g(t) = \sqrt{\beta(t)},$$

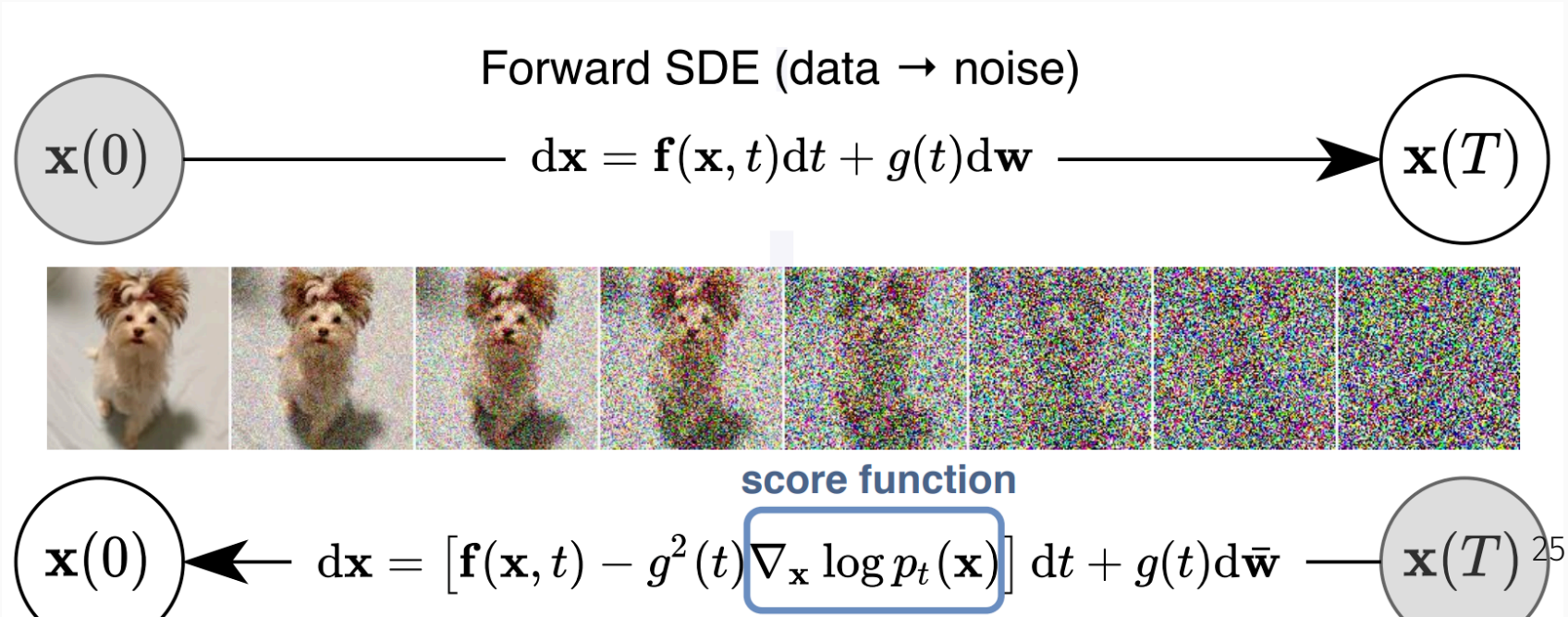
где $\mathbf{w}_t$ – винеровский процесс (броуновское движение), $\mathbf{f}$ – детерминированный дрейф, g – интенсивность шума; этот процесс постепенно превращает данные в шум (обычно стандартный гауссиан); пусть $p_t(\mathbf{x})$ – это плотность $\mathbf{x}(t)$

От DDIM к непрерывному времени

- Самый главный здесь результат — Anderson (1982): для такого процесса обратный тоже является диффузионным процессом и задаётся уравнением

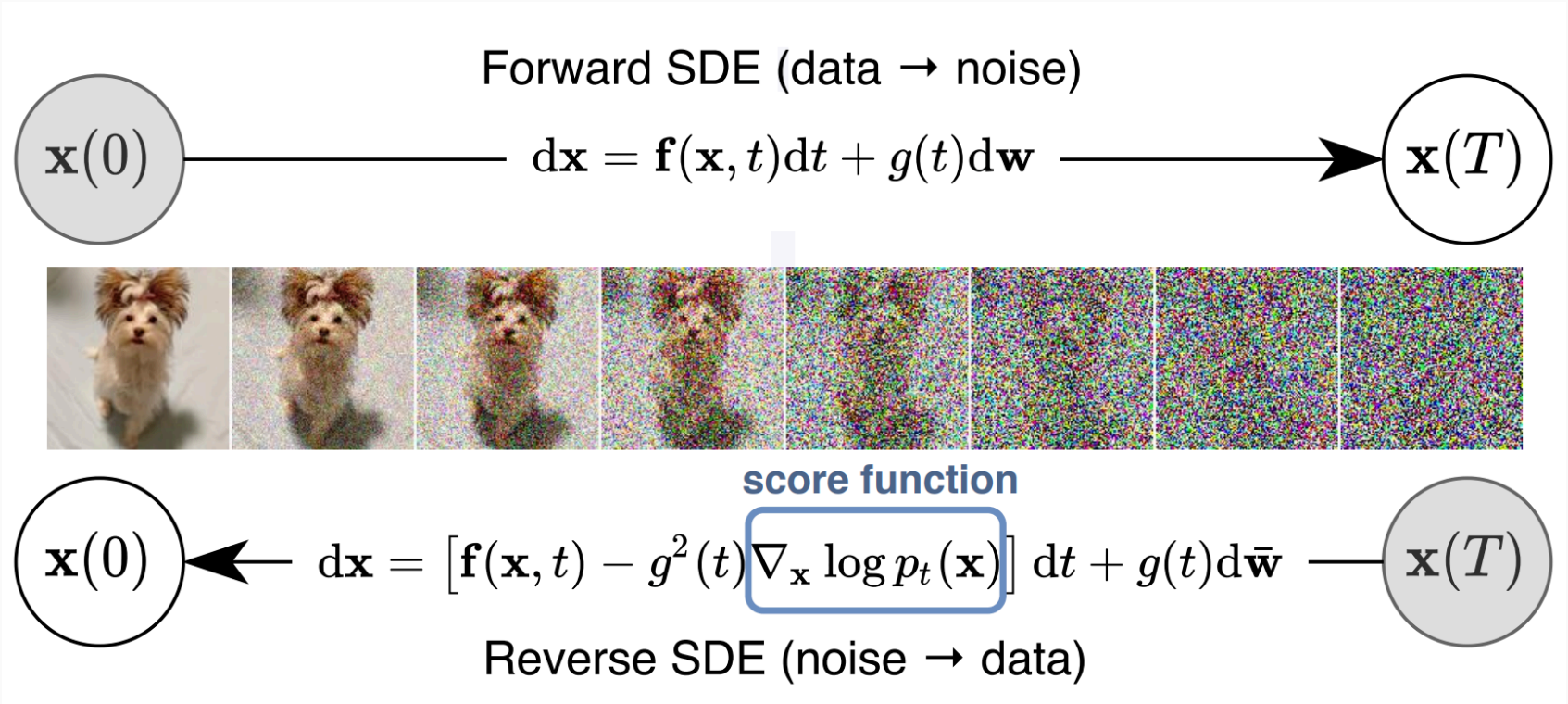
$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\, \mathrm{d}\bar{\mathbf{w}},$$

где $\bar{\mathbf{w}}$ – тоже стандартный винеровский процесс, но для времени от T до 0, а $\mathrm{d}t$ теперь отрицательное



От DDIM к непрерывному времени

- Функцию $s_*(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ в этой науке называют score; если score известен для каждого t, то процесс можно обратить
- Так что наш вопрос сводится к тому, откуда взять score

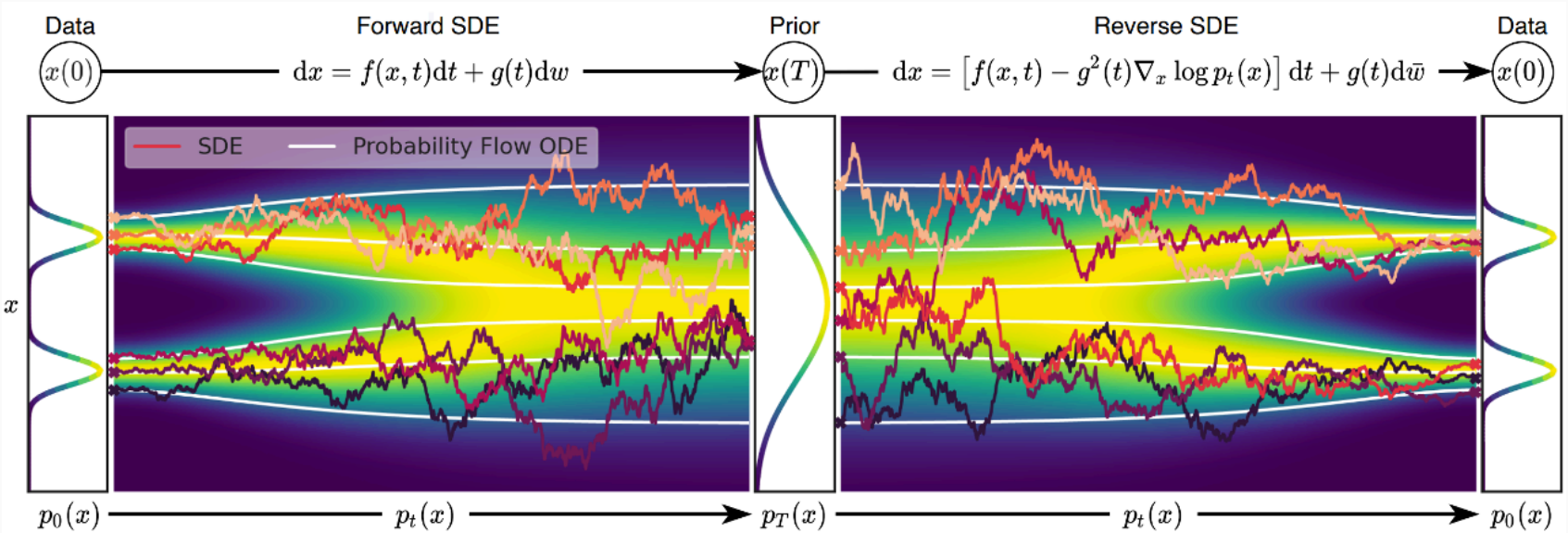


От DDIM к непрерывному времени

- A score, конечно же, можно и нужно обучать через score matching: ищем s_θ с параметрами θ, минимизируя

$$\mathbb{E}_t \left[\lambda(t) \mathbb{E}_{\mathbf{x}(0)} \left[\mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)} \left[\left\| s_\theta(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t)|\mathbf{x}(0)) \right\|_2^2 \right] \right] \right],$$

где λ – веса, а $p_t(\mathbf{x}(t)|\mathbf{x}(0))$ обычно можно так или иначе подсчитать (не будем уж в подробностях)



DDPM и DDIM как дискретизации SDE

- И теперь DDPM – это дискретизация такого процесса: наша марковская цепь $\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon$ сходится в пределе к стохастическому дифференциальному уравнению

$$\mathrm{d}\mathbf{x} = -\tfrac{1}{2}\beta(t)\mathbf{x}\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{w},$$

  т.е. как раз к такому процессу, который мы рассматривали
- Там есть два варианта: variance preserving (VP) как выше (дисперсия сохраняется единичной) и variance exploding (VE), в котором она всегда растёт, но это нам сейчас не важно
- Важно, что DDIM – это просто более грубая дискретизация того же процесса!

DDPM и DDIM как дискретизации SDE

- И даже более того – удивительно, но для любого диффузионного процесса есть и *детерминированный* процесс с теми же маргиналами $p_t(\mathbf{x})$! Для $s(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ это

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \tfrac{1}{2} g(t)^2 \, s(\mathbf{x}, t)\right] \mathrm{d}t$$

- Например, для DDPM как выше это будет

$$\mathrm{d}\mathbf{x} = -\tfrac{1}{2}\beta(t)\mathbf{x} - \tfrac{1}{2}\beta(t) s(\mathbf{x}, t)$$

- Интуитивно говоря, в SDE вероятностная масса течёт и размазывается, а в таком процессе течёт без шума, но так, что плотность p_t по времени остаётся той же

DDPM и DDIM как дискретизации SDE

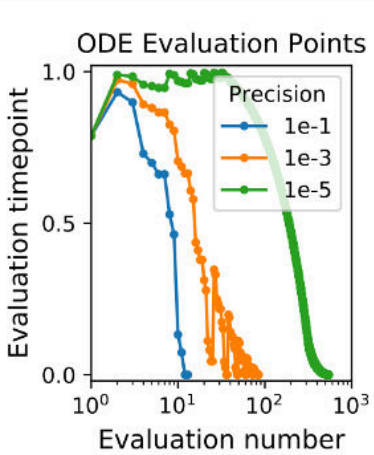
- И если теперь заменить s_* на s_θ, мы получим практическую формулу для порождения:

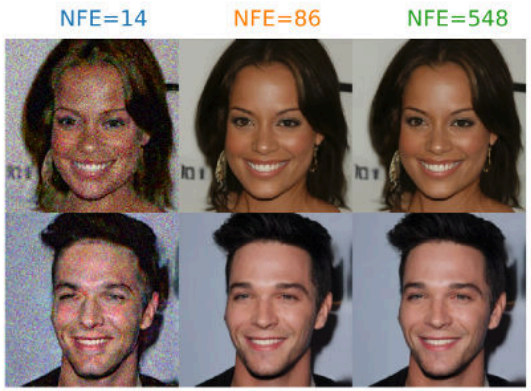
$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} \approx -\tfrac{1}{2}\beta(t)\Big(\mathbf{x} + s_\theta(\mathbf{x}, t)\Big).$$

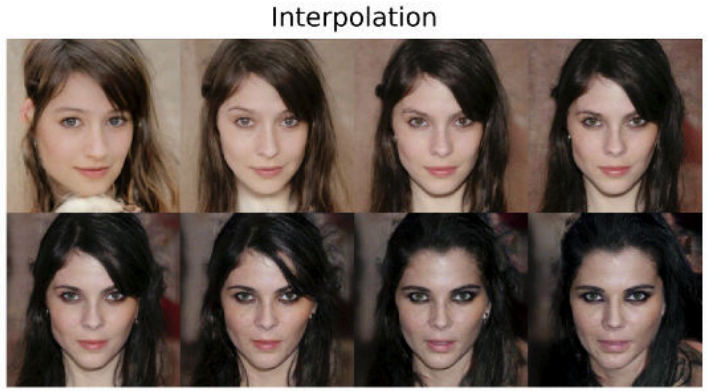
- Это и есть непрерывная версия DDIM: движение к $\mathbf{x}_0$ задаётся ОДУ, а не стохастическим процессом

DDPM и DDIM как дискретизации SDE

- Как обучать на практике: берём $\mathbf{x}_0$ из данных, получаем $\mathbf{x}_t$ из замкнутой формы распределения, минимизируем квадрат отклонения $s_\theta(\mathbf{x}, t)$ от истинного $\log p_t(\mathbf{x}(t)|\mathbf{x}(0))$
- Благодаря тому, что процесс можно считать детерминированным, появляются полезные эффекты:
  - удобные латентные коды и интерполяции;
  - пропуски шагов, как в DDIM: можно брать разреженную сетку по t и ускорять процесс;
  - траектория к $\mathbf{x}_0$ «более прямая», чем в стохастической обратной SDE.

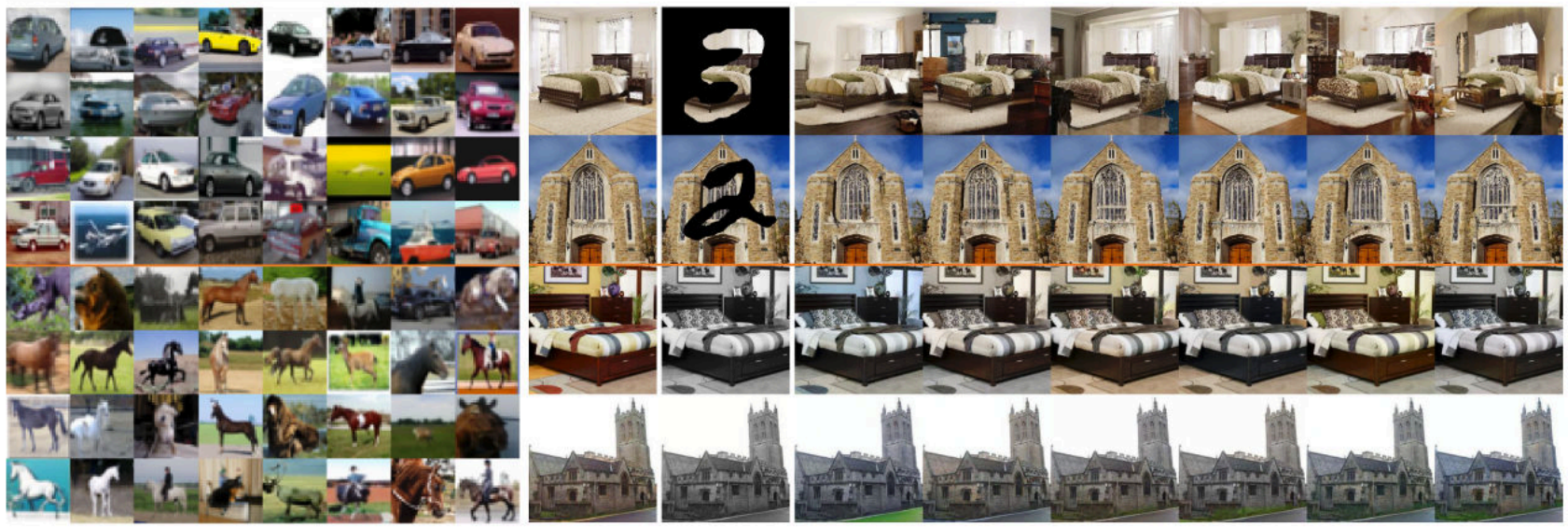






DDPM и DDIM как дискретизации SDE

- А ещё, кстати, всё это работает и для управляемого порождения: если нам известно $p_t\left(\mathbf{y}|\mathbf{x}(t)\right)$ для некоторого условия $\mathbf{y}$ (например, метки класса или частично заданного изображения), то мы можем так же сэмплировать из $p_0\left(\mathbf{x}(0)|y\right)$



Что дальше?

Последние новости диффузионных моделей

- Что произошло после DiT в диффузионных моделях?
- DPM-Solver++ (Lu et al., 2022): как дальше ускорить DDIM?
- Как мы видели выше, DDIM – это шаг первого порядка для интегрирования некоторого ОДУ
- Но ведь есть большая наука о том, как решать ОДУ лучше!
- Давайте выберем какую-нибудь многошаговую схему второго или третьего порядка, она будет понижать локальную ошибку, и в итоге можно будет шагать ещё шире, и шагов понадобится ещё меньше

Последние новости диффузионных моделей

- Например, DPM-Solver++ действительно стабильнее и лучше работает, чем другие варианты (о которых не будем):

DDIM (order = 1) (Song et al., 2021a)

PNDM (order = 2) (Liu et al., 2022b)

DEIS-1 (order = 2) (Zhang & Chen, 2022)

DEIS-2 (order = 3) (Zhang & Chen, 2022)

DPM-Solver (order = 2) (Lu et al., 2022)

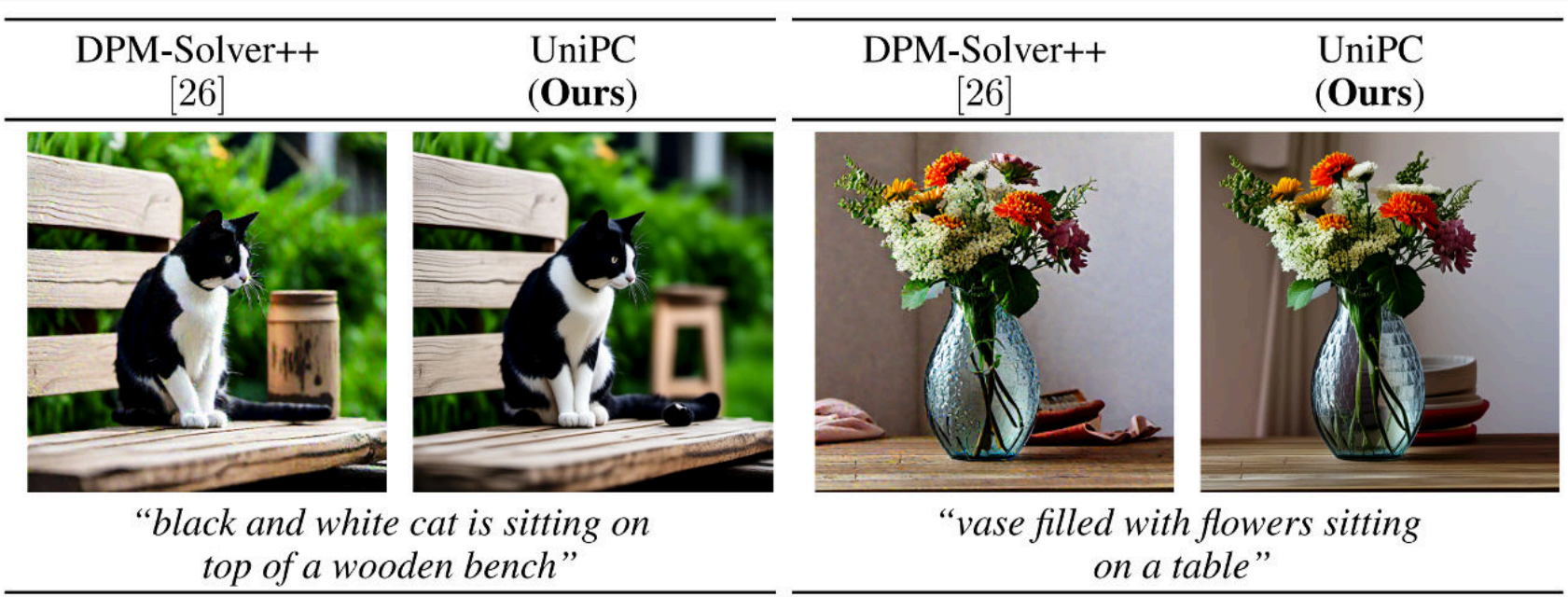
DPM-Solver-3 (order = 3) (Lu et al., 2022)

†DDIM (thresholding) (Saharia et al., 2022b)

DPM-Solver++ (order = 2) (**ours**)

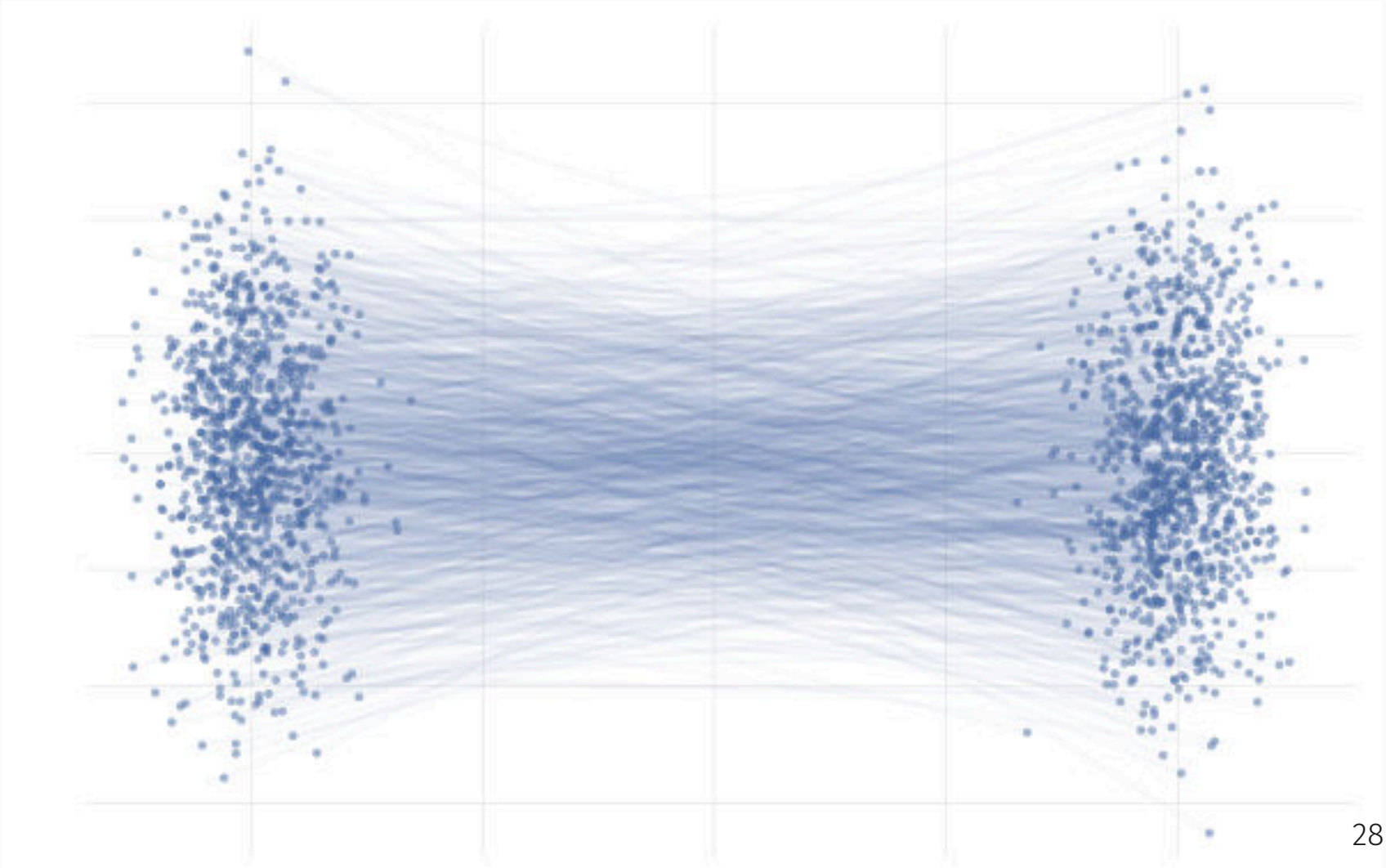
Последние новости диффузионных моделей

- UniPC (Zhao et al., 2023): метод типа «предиктор-корректор»
- Выводят универсальный корректирующий шаг (UniC), который можно добавить после любого базового предиктора, чтобы увеличить порядок точности без дополнительных вызовов модели, и выводят сопряжённый UniP-предиктор



Последние новости диффузионных моделей

- А самое интересное дальнейшее развитие событий – это, конечно, flow matching:



Последние новости диффузионных моделей

- На нём работают многие современные порождающие модели:

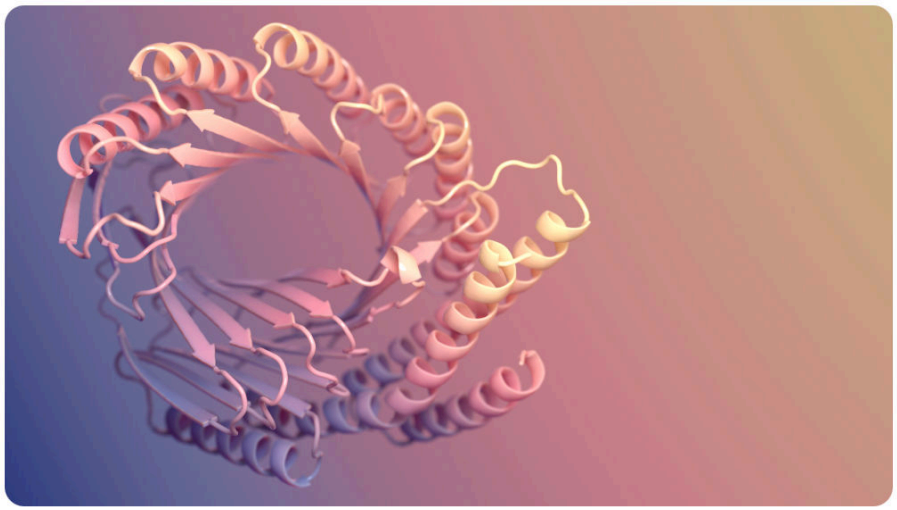


Figure 1: Protein generated by RFDiffusion (Watson et al., 2023).

Figure 2: Image from DALL-E 3 (Betker et al., 2023).

- Но это уже совсем другая история...

Спасибо!

Спасибо за внимание!