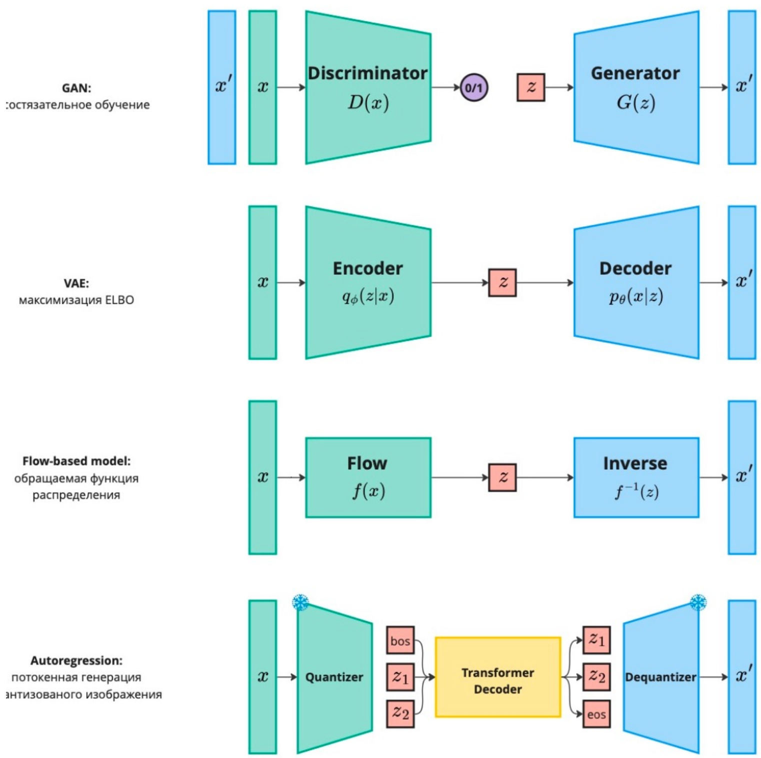
SBER AI

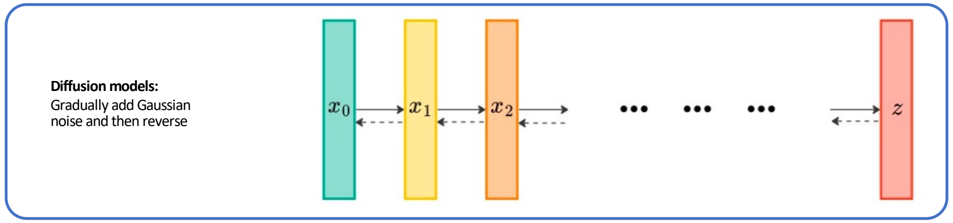
Adversarial Diffusion Distillation

Lev Novitskiy
ML Researcher, Sber AI

Generative architectures

SBER AI





Advantages

- Show better performance
- No adversarial training
- No mode collapse

Disadvantages

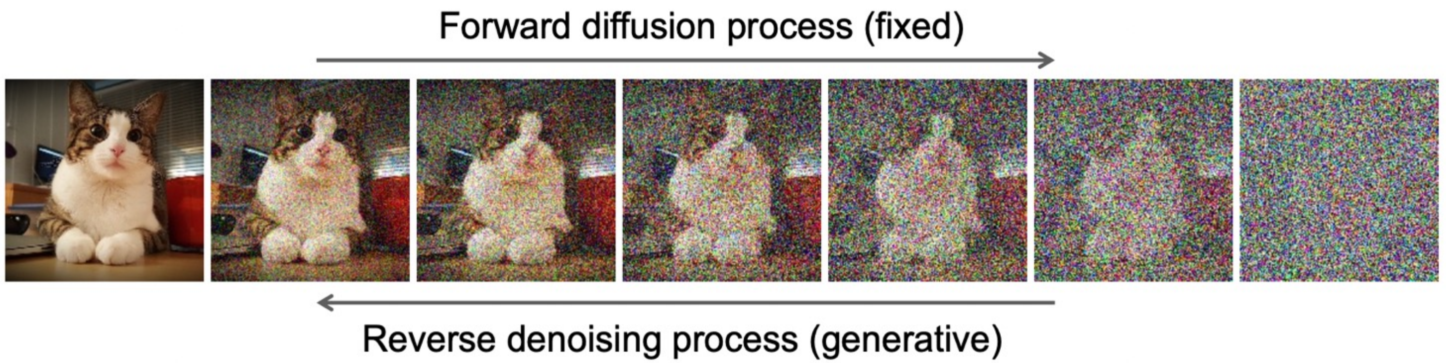
- Large inference time

Denoising Diffusion Probabilistic Models (DDPM)

SBER AI

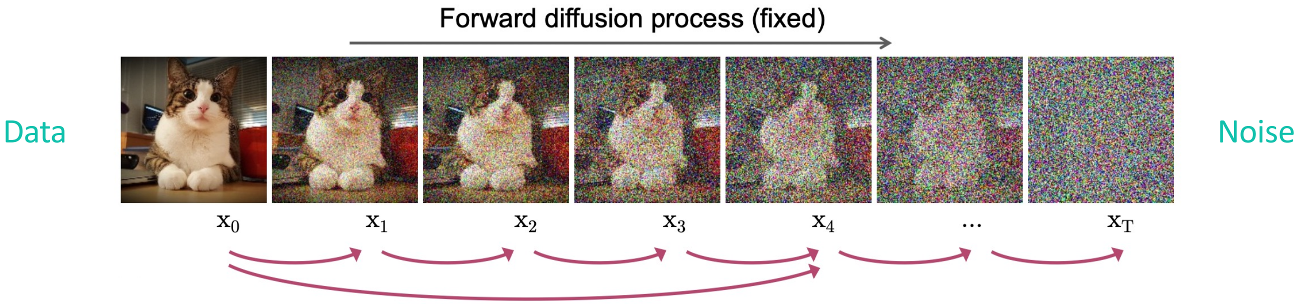
- **Forward** diffusion process – iterative **noise addition**
- **Reverse** diffusion process – iterative **noise removal**

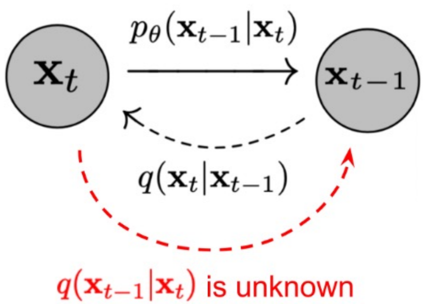
Data



Noise

Forward diffusion process





$\mathbf{x}_0 \sim q(\mathbf{x})$ — a probability density of the real data (images in our case)

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

$\{\beta_t \in (0,1)\}_{t=1}^{T}$ — dispersion of the size of each step t

Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$:

$$\begin{aligned}\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\ &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2} \\ &= \dots \\ &= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\end{aligned}$$

, where

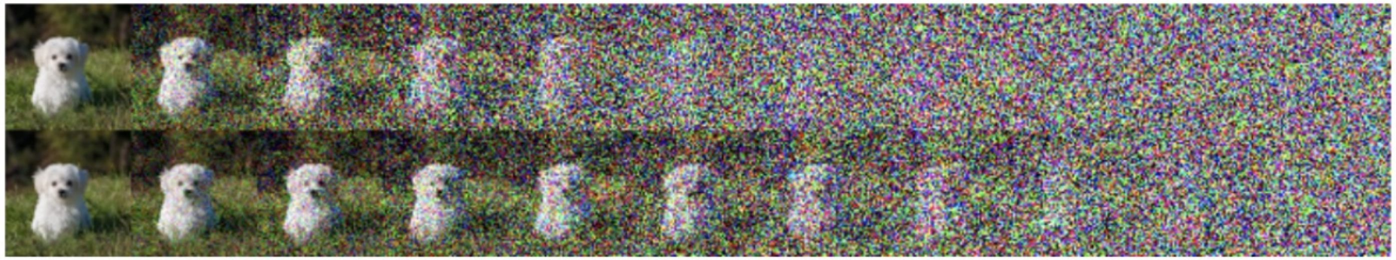
$\boldsymbol{\epsilon}_{t-1}, \boldsymbol{\epsilon}_{t-2}, \dots \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$\bar{\boldsymbol{\epsilon}}_{t-2}$ merges to Gaussians

$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\, \epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

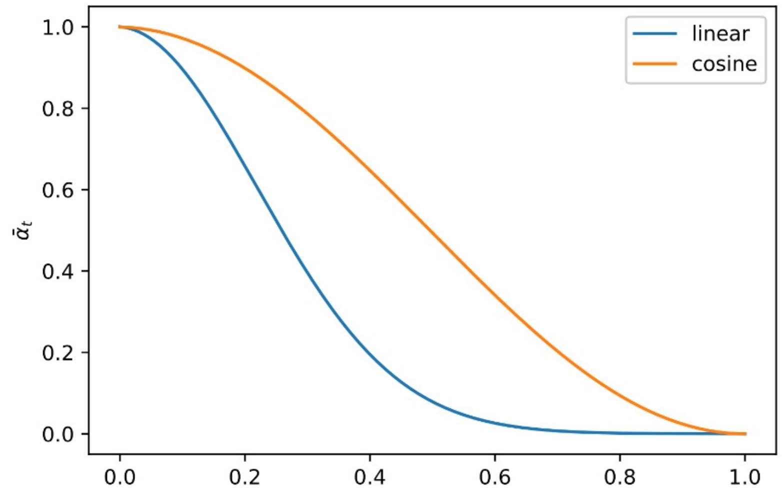
$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$ — probability of transition from $\mathbf{x}_0$ $\mathbf{x}_t$

Noise scheduler



β_t are chosen so that $\bar{\alpha}_T \to 0$

$$q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$$

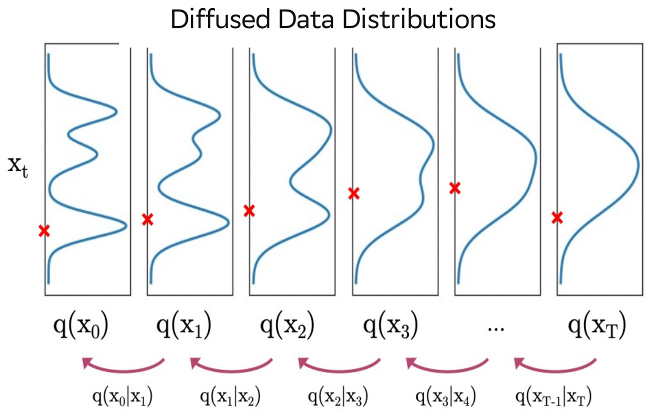


Reverse diffusion process

SBER AI

Process of generation (ideal situation):

1. Sampling $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$
2. Iterative sampling $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\mathbf{x}_t)$



We need to approximate:

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

Probability of a step in the reverse process:

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

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

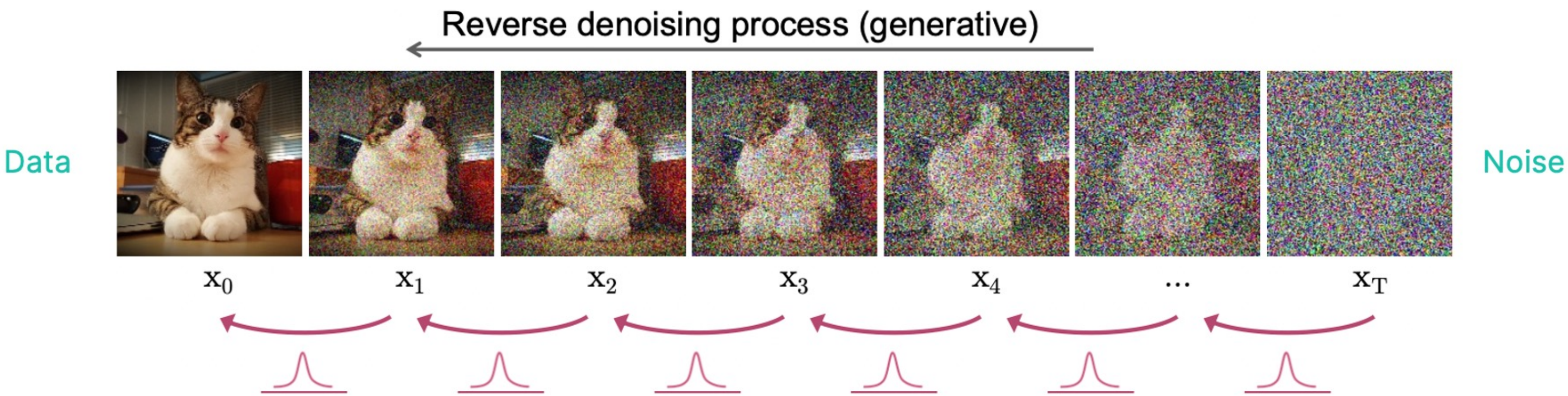
Parameters of the distribution:

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

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

Training of a reverse diffusion process

SBER AI



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

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

Trainable network
(U-net, Denoising Autoencoder)

$$\begin{aligned}
& -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0) = \\
& = -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \right) \\
& = -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} d\mathbf{x}_{1:T} \right) \\
& = -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right) \\
& \leq -\mathbb{E}_{q(\mathbf{x}_{0:T})} \log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \\
& = \mathbb{E}_{q(\mathbf{x}_{0:T})} \Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \Big] = L_\text{VLB}
\end{aligned}$$

$$\begin{aligned}
L_\text{VLB} = \mathbb{E}_q [& \underbrace{D_\text{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} \\
& + \sum_{t=2}^{T} \underbrace{D_\text{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]
\end{aligned}$$

$$D_\text{KL}(\mathcal{N}_0 \parallel \mathcal{N}_1) = \frac{1}{2}\left(\operatorname{tr}(\Sigma_1^{-1}\Sigma_0) - k + (\mu_1 - \mu_0)^\mathsf{T} \Sigma_1^{-1} (\mu_1 - \mu_0) + \ln\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right) \right)$$

$$D_\text{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) = \frac{\|\tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2}{2\tilde{\beta}_t}$$

Training and sampling strategy

SBER AI

As we know $x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_t)$

$$\tilde{\mu}(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right)$$

Here ϵ_t is a noise added to the sample x_0. Let our neural network approximate this noise by $\epsilon_\theta(x_t, t)$

$$L_{t-1} = \mathbb{E}_{x_0,\epsilon}\left[\frac{1}{2\tilde{\beta}_t}\|\tilde{\boldsymbol{\mu}}(x_t, x_0) - \boldsymbol{\mu}_\theta(x_t, t)\|^2\right]$$

$$= \mathbb{E}_{x_0,\epsilon}\left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\tilde{\beta}_t}\|\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)\|^2\right]$$

Simplification [5]

$$L_{\text{simple}} = \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\left[\|\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)\|^2\right]$$

Algorithm 1 Training

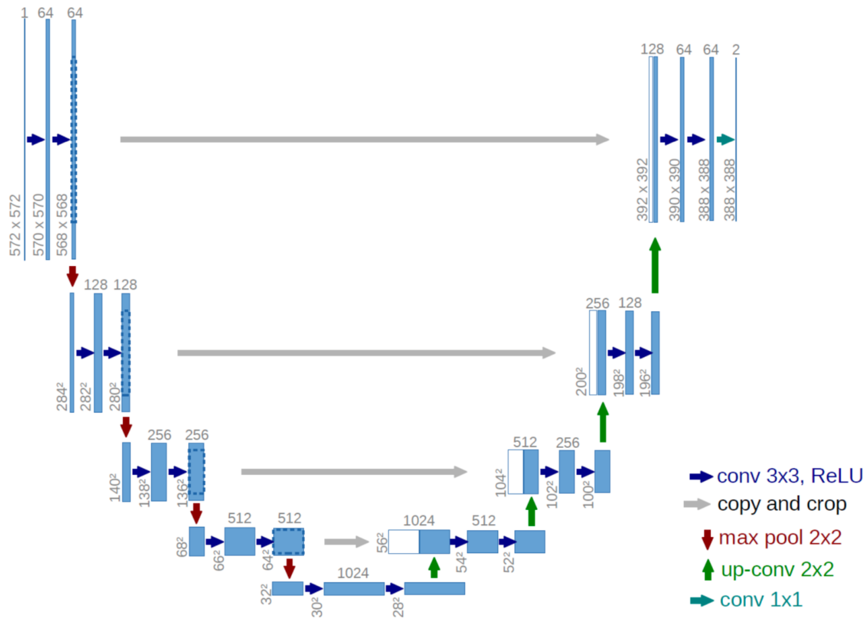
1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0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

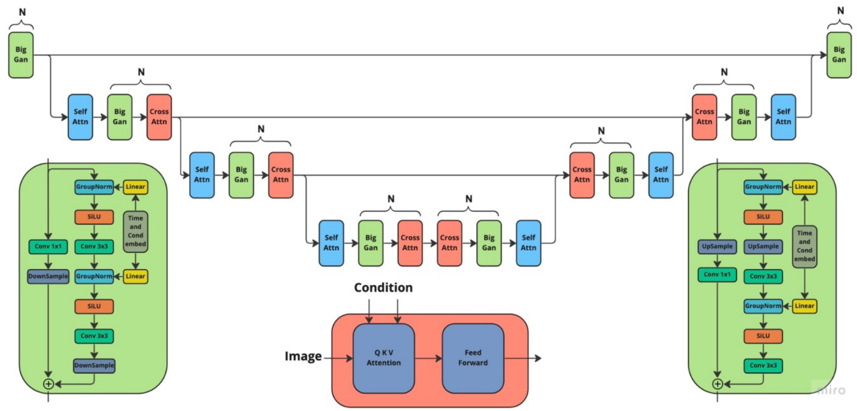
Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **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** $\mathbf{x}_0$

Base architecture – U-Net

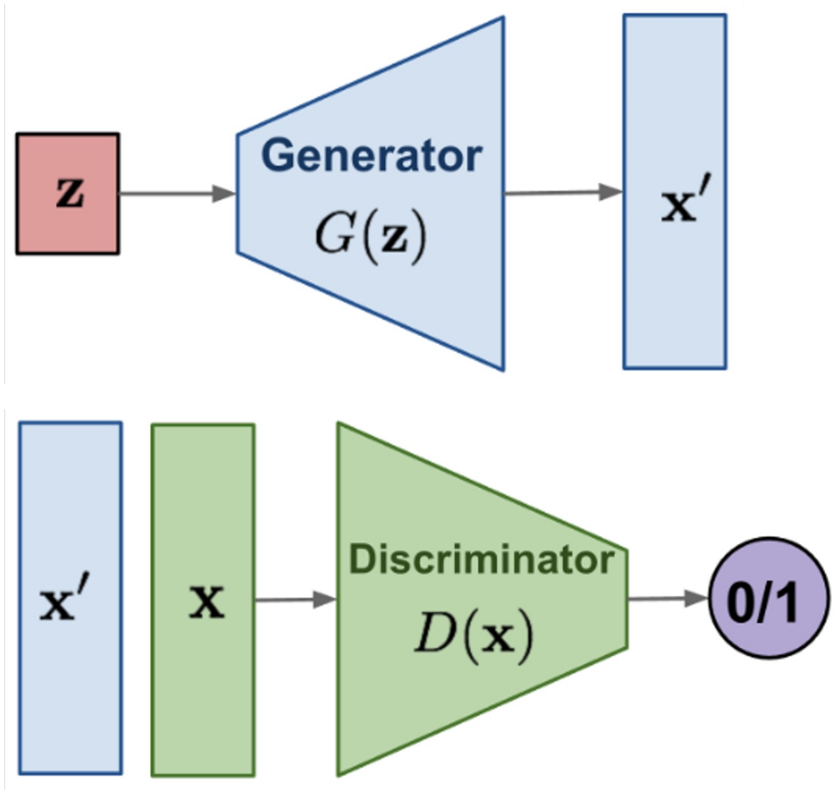
SBER AI





GAN

SBER AI



$$\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G\left(z^{(i)}\right)\right)\right)$$

$$\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(x^{(i)}\right)+\log\left(1-D\left(G\left(z^{(i)}\right)\right)\right)\right]$$

Проблемы GAN'ов

SBER AI

01

Low diversity (mode collapse)

02

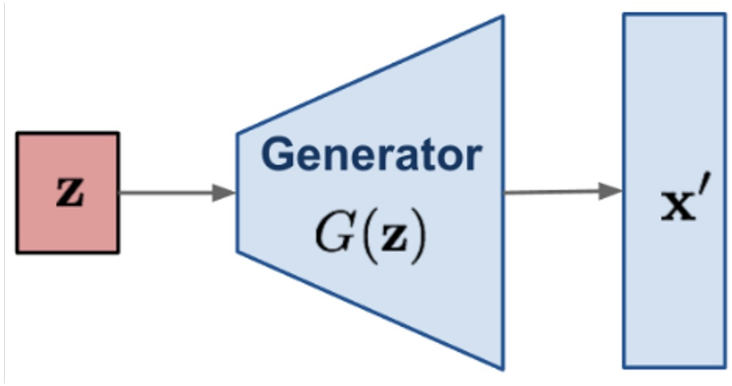
Non-convergence

03

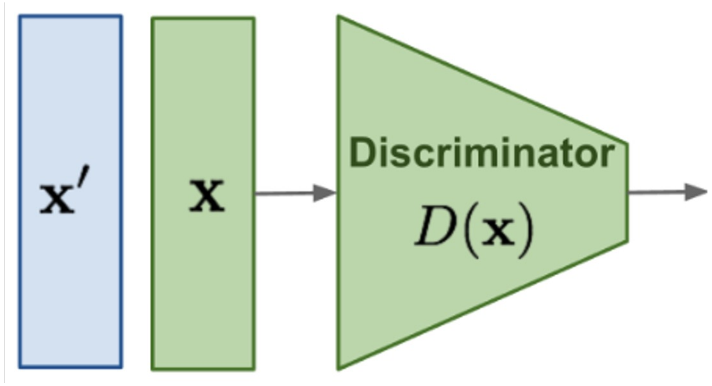
Instability

Wasserstein GAN

SBER AI



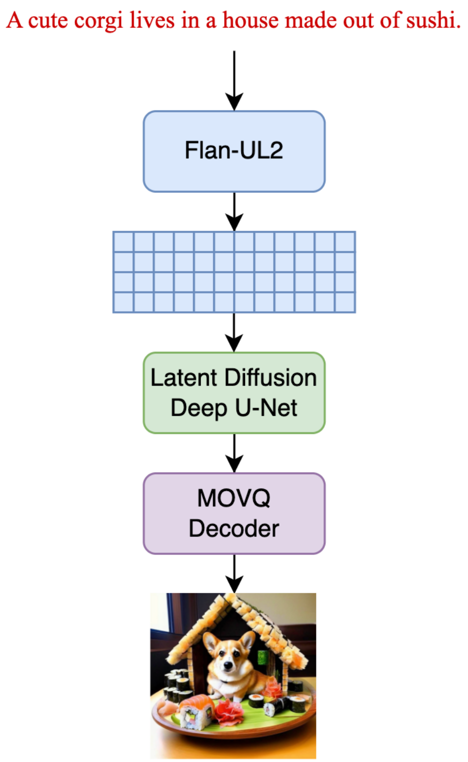
$$\frac{1}{m}\sum_{i=1}^{m}\left[f\left(G\left(\mathbf{z}^{(i)}\right)\right)\right]$$

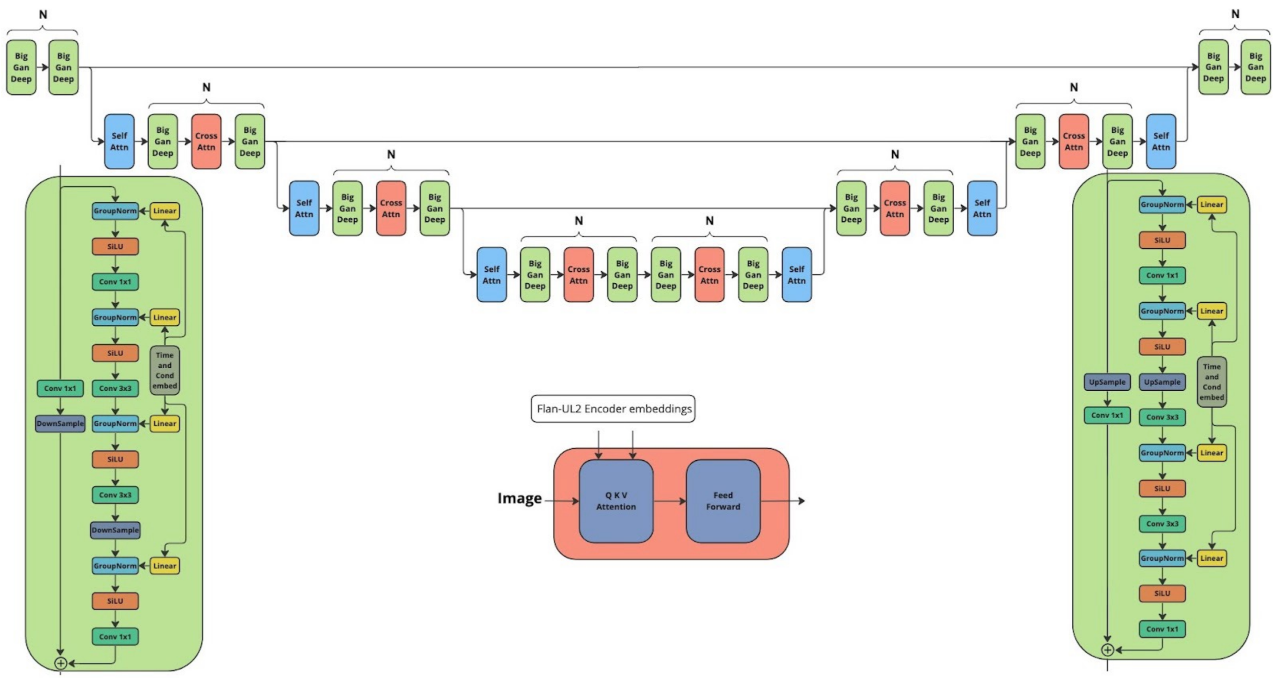


$$\frac{1}{m}\sum_{i=1}^{m}\left[f\left(\mathbf{x}^{(i)}\right)-f\left(G\left(\mathbf{z}^{(i)}\right)\right)\right]$$

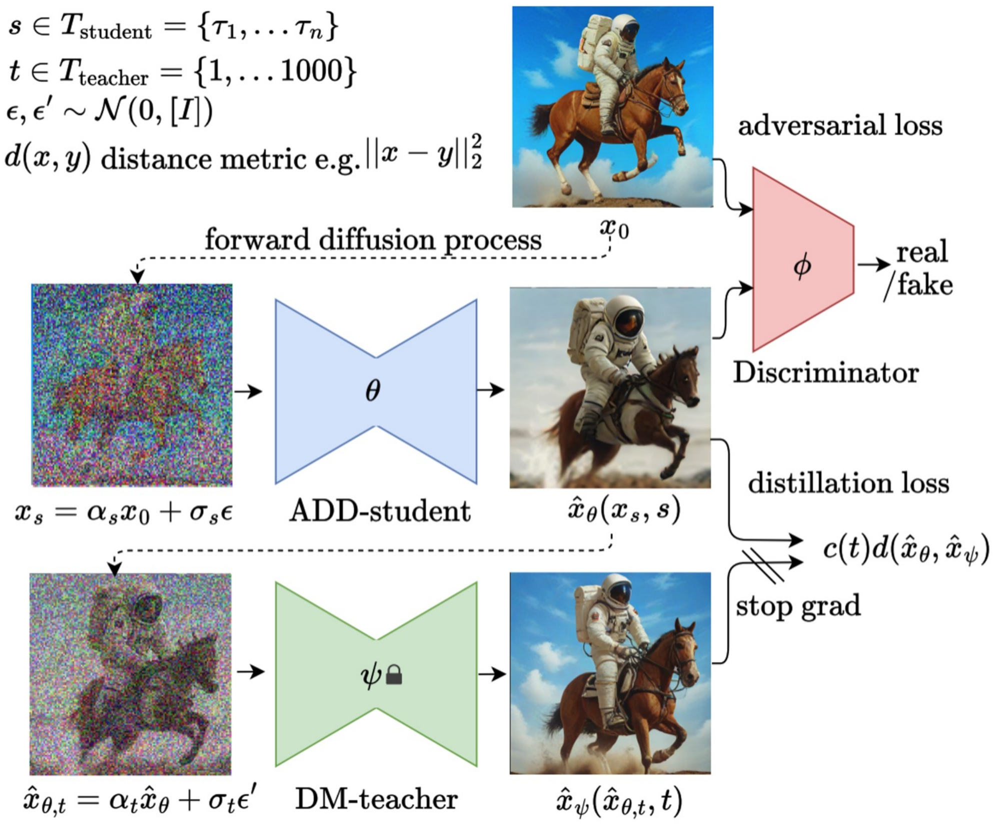
Kandinsky 3.0

SBER AI





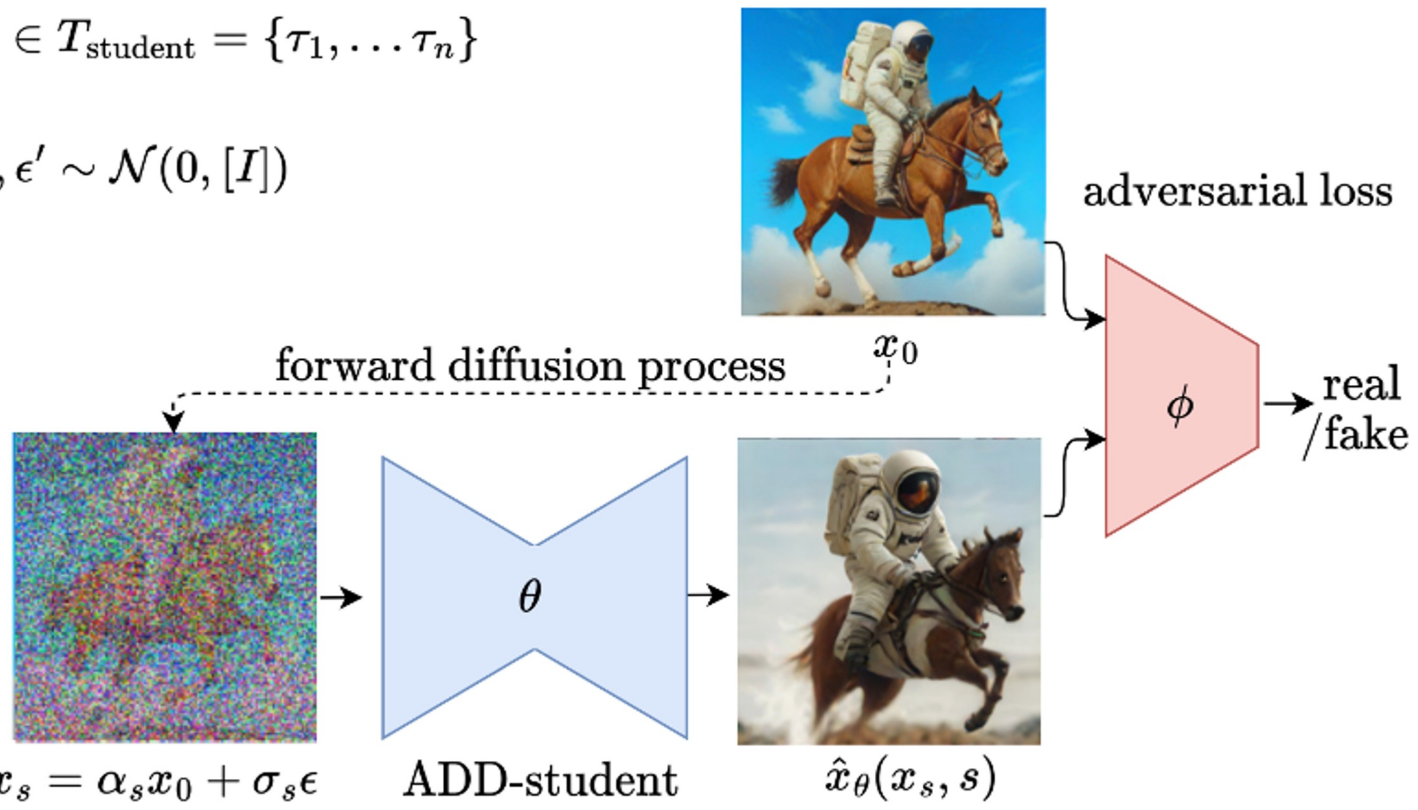
Adversarial Diffusion Distillation



Adversarial Diffusion Distillation

$s \in T_{\text{student}} = \{\tau_1, \dots \tau_n\}$

$\epsilon, \epsilon' \sim \mathcal{N}(0, [I])$



Kandinsky 3.1

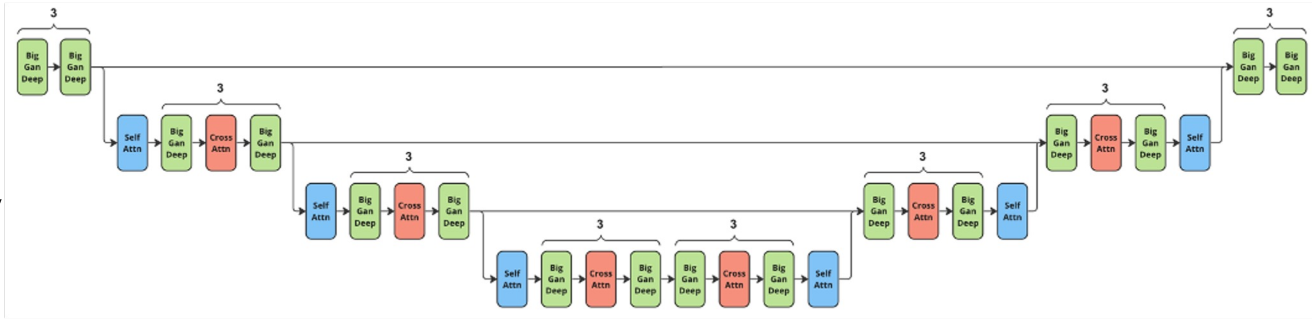
SBER AI

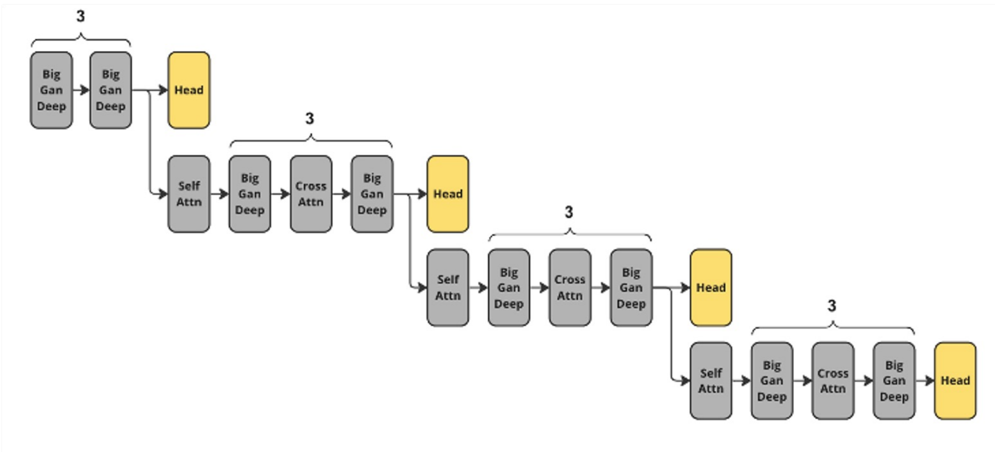
Adversarial loss on every step

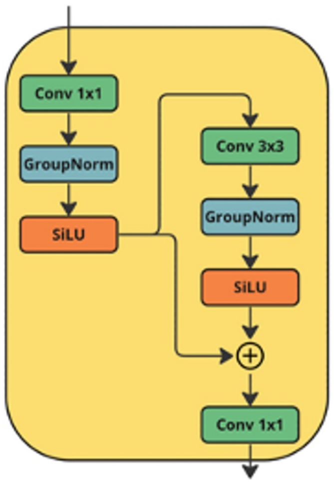
Generator - Kandinsky 3.0

Discriminator – Freezed Kandinsky 3.0 DownSample part + Heads

20x inference time boost







Примеры генерации

SBER AI

	A Pikachu with an angry expression and red eyes, with lightning around it, hyper realistic style	A panda is playing a guitar	A Pomeranian is sitting on the Kings throne wearing a crown. Two tiger soldiers are standing next to the throne
Kandinsky 3.0			
Distillated Kandinsky 3.0			

Quiz

SBER AI

Thank you!

SBER AI

Adversarial Diffusion Distillation

Lev Novitskiy
ML Researcher, Sber AI
@lefffffffffff