Fundamentos e Aplicações dos Modelos de Difusão

Modelos de Difusão

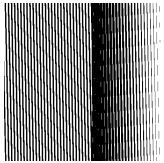
O que são?

- Modelos de difusão são uma classe de modelos generativos
- Aplicação principal: geração de imagens
  - Stable Diffusion
  - Dall-e
  - Sora (vídeos)
- Outras Aplicações
  - Geração de Áudios
  - Música
  - Robótica
  - etc

Modelos de Difusão

Modelos de Difusão

A fancy kangaroo wearing a tophat taking a mirror selfie on an elevator returning from a boxing session.



Modelos de Difusão

Modelos de Difusão

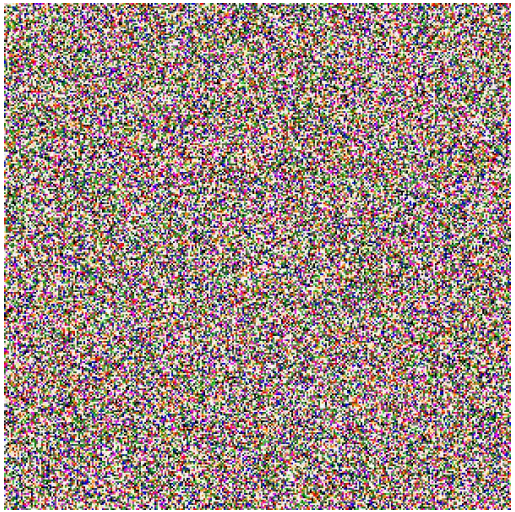
Modelos de Difusão

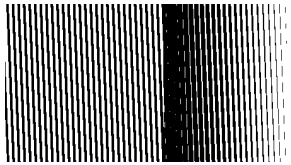
- Introdução
- **Visão Geral**
- Deep Denoising Probabilistic Models
- Melhorias
- Outras Aplicações

Visão Geral

- Queremos criar imagens do zero.
- Como garantir a diversidade das imagens geradas?
  - Partimos de um ruído!

Visão Geral





Essa transformação é muito difícil de ser aprendida

Visão Geral

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein JASCHA@STANFORD.EDU
Stanford University

Eric A. Weiss EAWEISS@BERKELEY.EDU
University of California, Berkeley

Niru Maheswaranathan NIRUM@STANFORD.EDU
Stanford University

Surya Ganguli SGANGULI@STANFORD.EDU
Stanford University

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International conference on machine learning*. PMLR, 2015.

Visão Geral

- Gradativamente perturbamos a imagem em pequenos passos.
- Nosso objetivo é encontrar uma função que a aprenda a reverter essas perturbações.
- Ao final temos um modelo que aprende a sair de um ruído aleatório, e gradualmente gerar uma imagem nova.

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International conference on machine learning*. PMLR, 2015.

Visão Geral

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International conference on machine learning*. PMLR, 2015.

Visão Geral

Generative Modeling by Estimating Gradients of the Data Distribution

Yang Song
Stanford University
yangsong@cs.stanford.edu

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Croitoru, Florinel-Alin & Hondru, Vlad & Ionescu, Radu Tudor & Shah, Mubarak. (2022). Diffusion Models in Vision: A Survey. 10.48550/arXiv.2209.04747.

DDPM

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

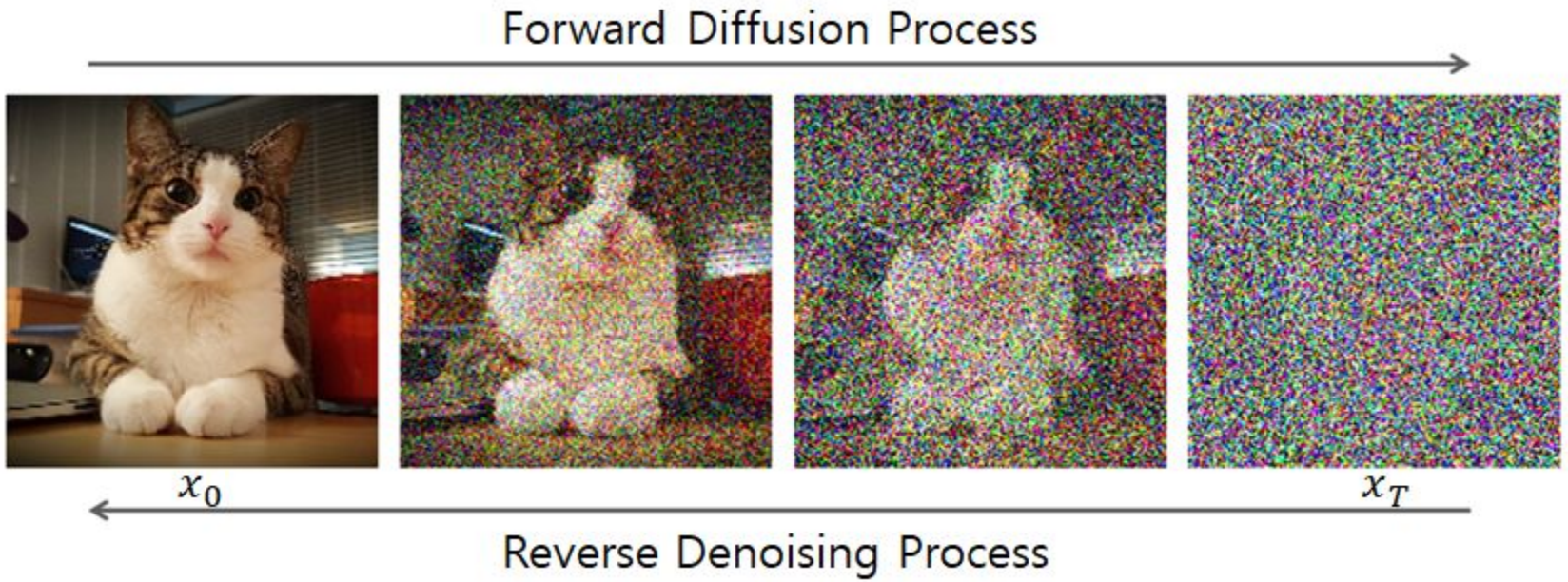
Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

Modelos de Difusão

- Introdução
- Visão Geral
- **Deep Denoising Probabilistic Models**
- Melhorias
- Outras Aplicações

DDPM - Overview



Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Overview

- Adicionamos ruído gaussiano seguindo uma distribuição linear constante.
- O modelo deve prever a distribuição do ruído:
  - Média
  - Variância -> fixada em valores constantes para simplificar o modelo
- 1000 time-steps
  - Partindo de um ruído inicial, a imagem passa 1000 vezes pela rede para chegar em sua versão final.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

- Queremos gerar dados que pertençam a uma distribuição inicial $p_{dados}(x)$
- Na prática, queremos que a distribuição $p_{\Theta}(x)$ gerada pelo nosso modelo seja similar à $p_{dados}(x)$.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

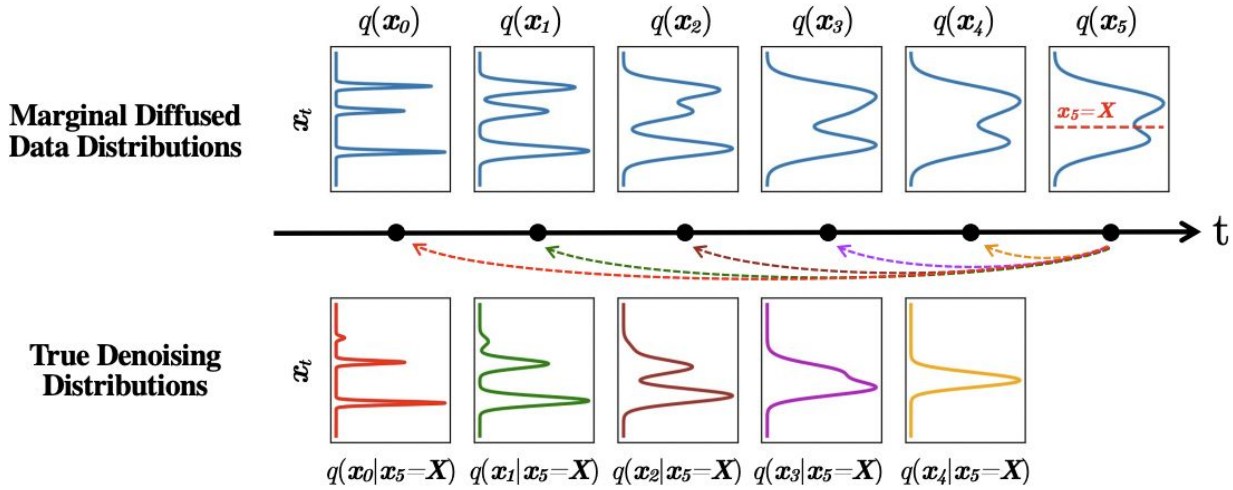
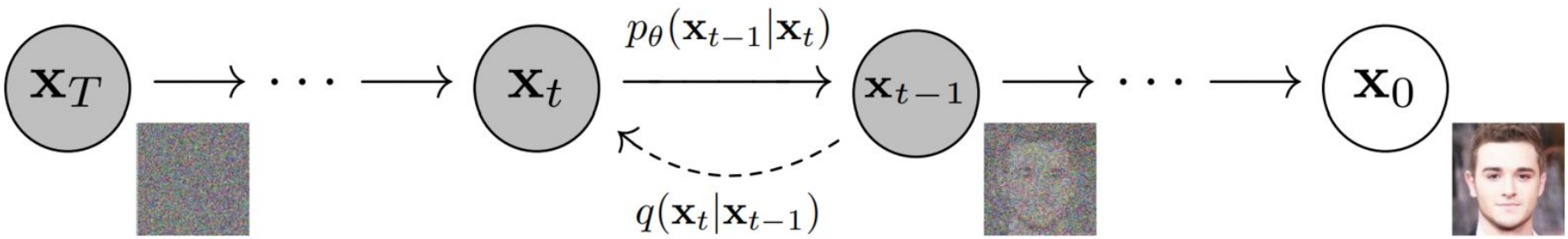


Figure 2: **Top**: The evolution of 1D data distribution $q(\mathbf{x}_0)$ through the diffusion process. **Bottom**:, The visualization of the true denoising distribution for varying step sizes conditioned on a fixed $\mathbf{x}_5$. The true denoising distribution for a small step size (i.e., $q(\mathbf{x}_4|\mathbf{x}_5 = X)$) is close to a Gaussian distribution. However, it becomes more complex and multimodal as the step size increases.

Imagem extraída de [10]

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive



X_0 ---------- Imagem original

X_i ---------- imagem com i passos de ruído adicionado

X_t ---------- imagem final (gaussiana aleatória)

$q(X_t \mid X_{t-1})$ ------ forward diffusion (adicionar ruído)

$p_\Theta(X_{t-1} \mid X_t)$ ---- backward diffusion (remover ruído)

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

Forward Diffusion

- Adicionamos ruídos seguindo um schedule de variância $\beta_1,...\beta_t$. O schedule usado é linear, com $\beta_1 = 0.0001$, $\beta_t = 0.02$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{2}$$

Podemos simplificar para:

$$X_t = \sqrt{1-\beta_t}X_{t-1} + \sqrt{\beta_t}\epsilon$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

$$X_t = \sqrt{1-\beta_t} X_{t-1} + \sqrt{\beta_t}\epsilon$$

Passar de X_0 para X_t requer t passos…

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

$$
\begin{aligned}
x_t &= \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\
&= \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_t \alpha_{t-1}}\bar{\epsilon}_{t-2} \\
&= \dots \\
&= \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \\
\alpha_t &= 1-\beta_t, \bar{\alpha}_t = \prod_{t=1}^{T} \alpha_t
\end{aligned}
$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

Backward Diffusion

- Lembre que, para passos, o processo reverso é modelado por uma gaussiana.

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

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

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

- Queremos aproximar a distribuição dos dados gerados da distribuição real.
  - $-\log(p_\Theta(X_0))$
  - Na prática, não conseguimos calcular $p_\Theta(X_0)$.
- Otimizamos, então, um limite inferior:

$$-\log p_\theta(\mathbf{x}_0) \leq -\log p_\theta(\mathbf{x}_0) + D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0)||p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)))$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

$$
\begin{aligned}
-\log p_\theta(\mathbf{x}_0) &\leq -\log p_\theta(\mathbf{x}_0) + D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0)\|p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)) \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{\mathbf{x}_{1:T}\sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})/p_\theta(\mathbf{x}_0)}\right] \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_q\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} + \log p_\theta(\mathbf{x}_0)\right] \\
&= \mathbb{E}_q\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \\
\text{Let } L_{\mathrm{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \geq -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0)
\end{aligned}
$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

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

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

$$
\begin{aligned}
L_{\text{VLB}} &= L_T + L_{T-1} + \cdots + L_0 \\
\text{where} L_T &= D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T)) \\
L_t &= D_{\text{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})) \text{for} 1 \leq t \leq T-1 \\
L_0 &= -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)
\end{aligned}
$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

$$L_t = D_{\mathrm{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}))\text{for}1 \leq t \leq T-1$$

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

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

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

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \left\| \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) \right\|^2 \right] + C$$

$$L_{t-1} - C = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{1}{2\sigma_t^2} \left\| \tilde{\boldsymbol{\mu}}_t \left(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) - \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}) \right) - \boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), t) \right\|^2 \right] \tag{9}$$

$$= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon} \right) - \boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), t) \right\|^2 \right] \tag{10}$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t))\right) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) \tag{11}$$

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \tag{12}$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \tag{14}$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

$$L_0 = -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$$

Como lidar com o fato da imagem ser discreta?

- Convertemos a imagem para [-1, 1]
- Criamos um decoder independente derivado da gaussiana $\mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}_\theta(\mathbf{x}_1, 1), \sigma_1^2\mathbf{I})$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

$$L_0 = -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$$

$$p_\theta(\mathbf{x}_0|\mathbf{x}_1) = \prod_{i=1}^{D} \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}(x; \mu_\theta^i(\mathbf{x}_1, 1), \sigma_1^2)\, dx \tag{13}$$

$$\delta_+(x) = \begin{cases} \infty & \text{if } x = 1 \\ x + \frac{1}{255} & \text{if } x < 1 \end{cases} \qquad \delta_-(x) = \begin{cases} -\infty & \text{if } x = -1 \\ x - \frac{1}{255} & \text{if } x > -1 \end{cases}$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \tag{14}$$

Para t=1, essa loss aproxima L_0, ignorando alguns fatores pouco relevantes.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

A loss é um MSE do ruído predito com o ruído real :)

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

Diffusion Models | Paper Explanation | Math Explained (youtube.com)

DDPM - Deep Dive

Resumindo:

- Partimos de um Variational Lower Bound.

$$\text{Let}\, L_{\text{VLB}} = \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \geq -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0)$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

Resumindo:

- Simplificamos até chegar em L_0, L_{t-1} e L_t.
- Como a variância foi fixada, L_t é constante e pode ser ignorado.

$$
\begin{aligned}
L_{\text{VLB}} &= L_T + L_{T-1} + \cdots + L_0 \\
\text{where} L_T &= D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T)) \\
L_t &= D_{\text{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})) \text{for} 1 \leq t \leq T-1 \\
L_0 &= -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)
\end{aligned}
$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

Resumindo:

- Reparametrizamos L_{t-1} para prever a média do ruído.

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \| \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) \|^2 \right] + C$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

Resumindo:

- E novamente para prever o ruído. Agora, chegamos em uma norma L_2 escalada.

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},t)\right\|^2\right] \tag{12}$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

Resumindo:

- Ignorar o termo constante é melhor na prática, pois faz a rede focar nos passos finais.

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \tag{14}$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Deep Dive

Resumindo:

- Com alguns truques, incluímos L_0 em $L_{simples}$.
- Convertemos a imagem de [0, 255] para [-1, 1]

$$p_\theta(\mathbf{x}_0|\mathbf{x}_1) = \prod_{i=1}^{D} \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}(x; \mu_\theta^i(\mathbf{x}_1, 1), \sigma_1^2)\, dx \tag{13}$$

$$\delta_+(x) = \begin{cases} \infty & \text{if } x = 1 \\ x + \frac{1}{255} & \text{if } x < 1 \end{cases} \qquad \delta_-(x) = \begin{cases} -\infty & \text{if } x = -1 \\ x - \frac{1}{255} & \text{if } x > -1 \end{cases}$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Implementação

Algorithm 1 Training

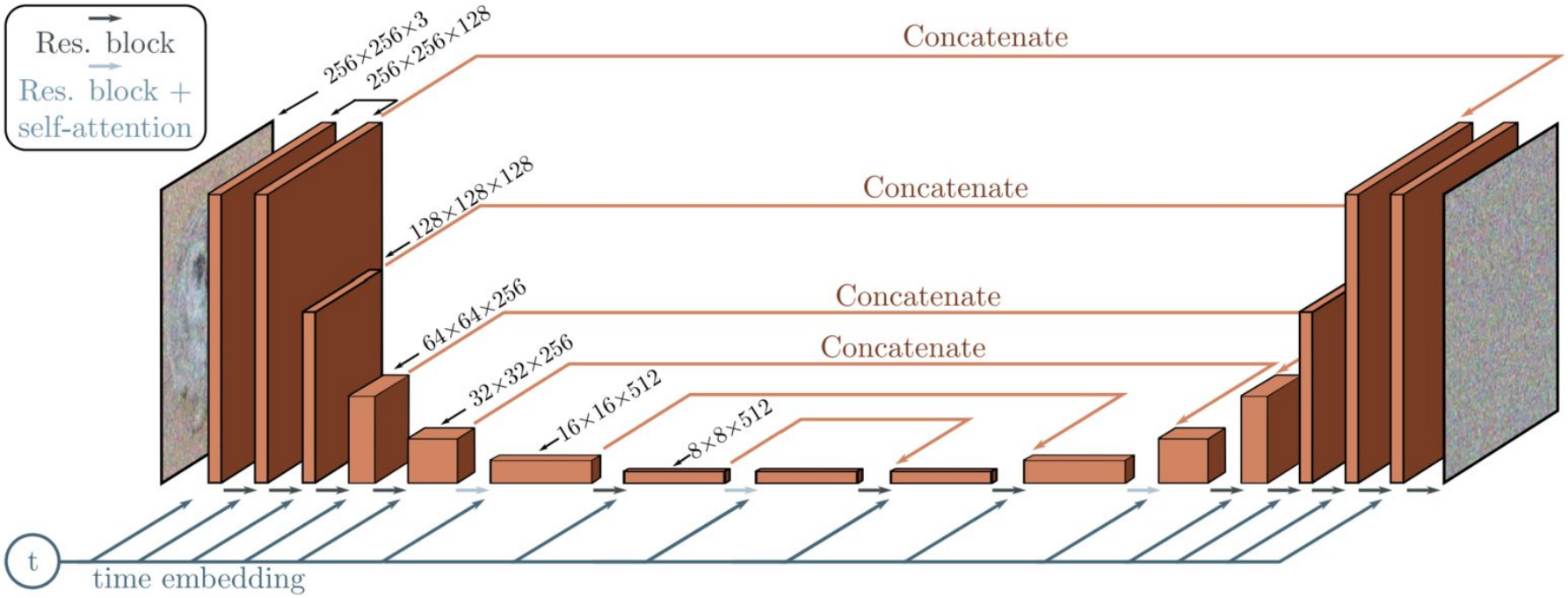
1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0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

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Implementação



Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

DDPM - Implementação

- Camadas residuais usando GroupNorm
- Self-attention entre blocos convolucionais na camada 16x16
- Modelo condicionado no timestep t:
  - Embedding sinusoidal dos transformers
- 36M parâmetros para CIFAR-10 e 114M para LSUN e CelebA-HQ

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

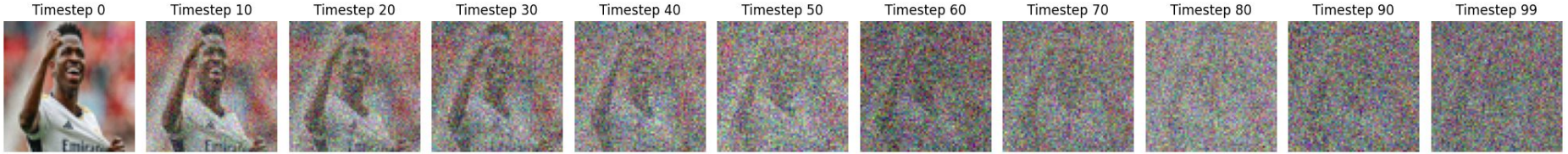
DDPM

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

Modelos de Difusão

- Introdução
- Visão Geral
- Deep Denoising Probabilistic Models
- **Melhorias**
- Outras Aplicações

DDPM - Melhorias



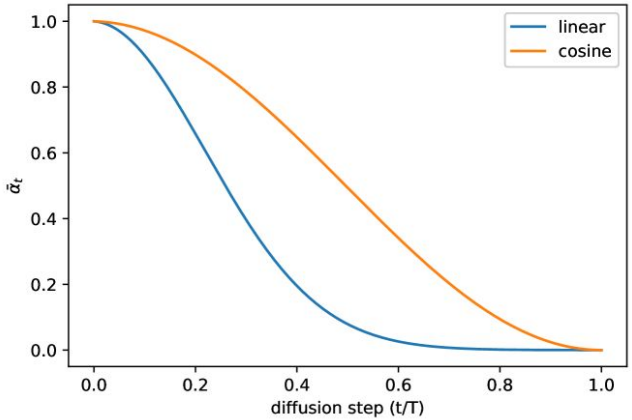
DDPM - Melhorias

Improved Denoising Diffusion Probabilistic Models

Alex Nichol [*1] **Prafulla Dhariwal** [*1]

Nichol, A.Q. , Dhariwal, P.. (2021). Improved Denoising Diffusion Probabilistic Models.Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research 139:8162-8171 Available from https://proceedings.mlr.press/v139/nichol21a.html.

Cosine Schedule



Nichol, A.Q. , Dhariwal, P.. (2021). Improved Denoising Diffusion Probabilistic Models.Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research 139:8162-8171 Available from https://proceedings.mlr.press/v139/nichol21a.html.

Cosine Schedule

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right)^2 \qquad (17)$$

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

β_t é clipado em 0.999.
s = 0.008

Nichol, A.Q. , Dhariwal, P.. (2021). Improved Denoising Diffusion Probabilistic Models.Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research 139:8162-8171 Available from https://proceedings.mlr.press/v139/nichol21a.html.

Aprender a Covariância

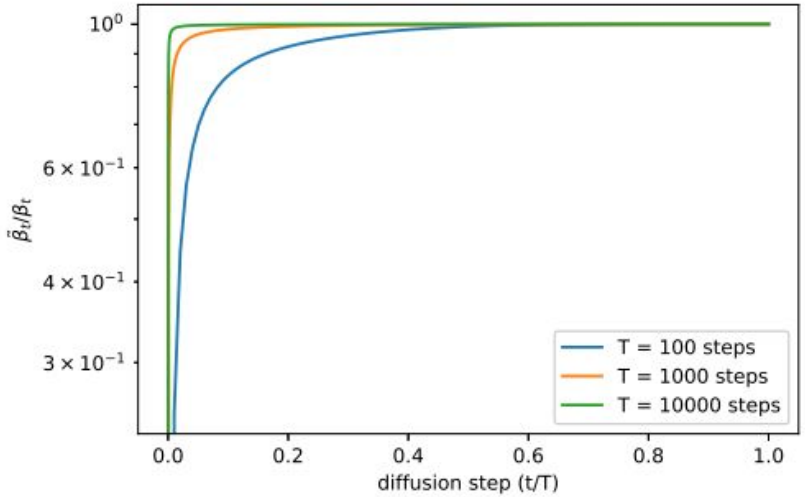
- A matriz de covariância é dada pela seguinte fórmula: $\mathbf{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I}$
- Em DDPM, os autores concluíram experimentalmente que fixar a variância em:
  - $\sigma_t^2 = \beta_t$ e
  - $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$

  Não impacta a qualidade das imagens.

Nichol, A.Q. , Dhariwal, P.. (2021). Improved Denoising Diffusion Probabilistic Models.Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research 139:8162-8171 Available from https://proceedings.mlr.press/v139/nichol21a.html.

Aprender a Covariância

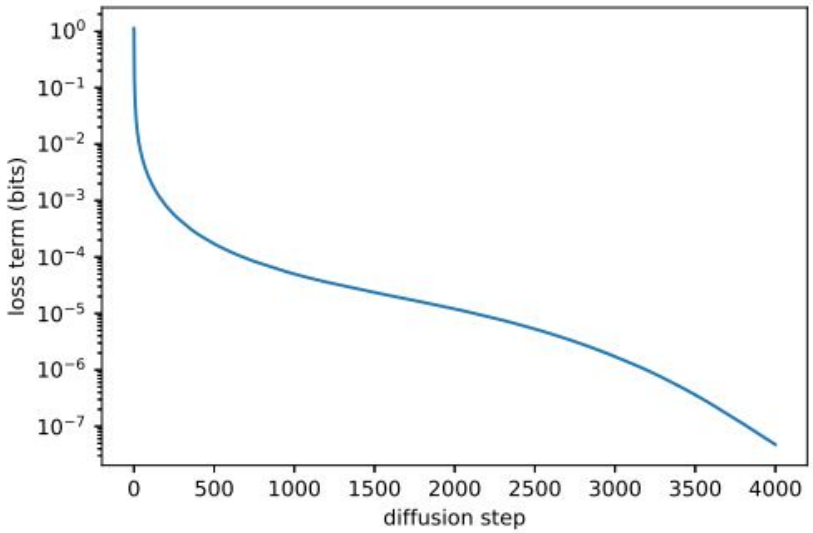
Intuitivamente, observamos que ambos os valores são próximos, com excessão para ts próximos a 0.



Nichol, A.Q. , Dhariwal, P.. (2021). Improved Denoising Diffusion Probabilistic Models.Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research 139:8162-8171 Available from https://proceedings.mlr.press/v139/nichol21a.html.

Aprender a Covariância

Porém, os primeiros passos são os mais importantes para a log-likelihood.



Nichol, A.Q. , Dhariwal, P.. (2021). Improved Denoising Diffusion Probabilistic Models.Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research 139:8162-8171 Available from https://proceedings.mlr.press/v139/nichol21a.html.

Aprender a Covariância

- Fixar a variância tem um impacto desprezível para a qualidade das imagens.
- No entanto, tem um impacto significativo na log-likelihood.
    - Em termos práticos, aumentar a log-likelihood implica em uma maior cobertura da distribuição original, ou seja, uma maior diversidade da rede.

Nichol, A.Q. , Dhariwal, P.. (2021). Improved Denoising Diffusion Probabilistic Models.Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research 139:8162-8171 Available from https://proceedings.mlr.press/v139/nichol21a.html.

Aprender a Covariância

- A rede agora prevê um vetor v com um componente por dimensão.
- Calculamos a variância com a fórmula

$$\Sigma_\theta(x_t, t) = \exp(v \log \beta_t + (1 - v) \log \tilde{\beta}_t) \qquad (15)$$

- Essa modelagem trata a variância como uma interpolação entre os betas.

Nichol, A.Q. , Dhariwal, P.. (2021). Improved Denoising Diffusion Probabilistic Models.Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research 139:8162-8171 Available from https://proceedings.mlr.press/v139/nichol21a.html.

Aprender a Covariância

- Temos uma nova loss:

$$L_{\text{hybrid}} = L_{\text{simple}} + \lambda L_{\text{vlb}} \qquad (16)$$

- Com $\lambda = 0.001$

Nichol, A.Q. , Dhariwal, P.. (2021). Improved Denoising Diffusion Probabilistic Models.Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research 139:8162-8171 Available from https://proceedings.mlr.press/v139/nichol21a.html.

DDPM - Melhorias

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal*
OpenAI
prafulla@openai.com

Alex Nichol*
OpenAI
alex@openai.com

Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." Advances in neural information processing systems 34 (2021): 8780-8794.

DDPM - Melhorias

- Aumentar profundidade x largura (mantendo o tamanho da rede constante)
- Mais attention heads
- Attention nas camadas 32x32, 16x16 e 8x8 em vez de apenas na 16x16
- Usar os blocos residuais da BigGAN
- Reescalar conexões residuais com $1/\sqrt{2}$

Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." Advances in neural information processing systems 34 (2021): 8780-8794.

Classifier Guidance

Queremos condicionar o modelo a gerar imagens de uma classe específica.

1. Começamos com um modelo de difusão já treinado incondicionalmente.
2. Treinamos um classificador em imagens ruidosas
3. Usamos os gradientes do classificador para guiar a geração de imagens.

Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." Advances in neural information processing systems 34 (2021): 8780-8794.

Classifier Guidance

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return x_0

Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." Advances in neural information processing systems 34 (2021): 8780-8794.

Classifier Guidance

- Pouco interpretável;
- Não é possível usar classificadores pré-treinados

Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." Advances in neural information processing systems 34 (2021): 8780-8794.

DDPM - Melhorias

Figure 6: Samples from BigGAN-deep with truncation 1.0 (FID 6.95, left) vs samples from our diffusion model with guidance (FID 4.59, middle) and samples from the training set (right).

Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." Advances in neural information processing systems 34 (2021): 8780-8794.

Classifier-Free Guidance

CLASSIFIER-FREE DIFFUSION GUIDANCE

Jonathan Ho & Tim Salimans
Google Research, Brain team
{jonathanho,salimans}@google.com

Ho, Jonathan, and Tim Salimans. "Classifier-free diffusion guidance." arXiv preprint arXiv:2207.12598 (2022).

Classifier-Free Guidance

Ideia básica: aplicar a regra de bayes para eliminar o classificador

$$
\begin{aligned}
& +w\nabla_x \log p(y|x) \\
= \quad & +w\nabla_x \log \frac{p(x|y)p(y)}{p(x)} \\
= \quad & +w\nabla_x \log p(x|y) + w\nabla_x p(y) - w\nabla_x \log p(x) \\
= \quad & +w\nabla_x \log p(x|y) + 0 - w\nabla_x \log p(x) \\
= \quad & w\epsilon_\theta(x, y) - w\epsilon_\theta(x)
\end{aligned}
$$

Ho, Jonathan, and Tim Salimans. "Classifier-free diffusion guidance." arXiv preprint arXiv:2207.12598 (2022).

Classifier-Free Guidance

- Treinamos dois modelos simultaneamente:
  - Um condicional
  - Um incondicional
- Usamos a mesma rede neural para os dois
  - Na prática, descartamos a informação condicional com alguma probabilidade.

Ho, Jonathan, and Tim Salimans. "Classifier-free diffusion guidance." arXiv preprint arXiv:2207.12598 (2022).

Classifier-Free Guidance

Ho, Jonathan, and Tim Salimans. "Classifier-free diffusion guidance." arXiv preprint arXiv:2207.12598 (2022).

Classifier-Free Guidance

- Não precisamos de classe para condicionar nosso modelo!

Ho, Jonathan, and Tim Salimans. "Classifier-free diffusion guidance." arXiv preprint arXiv:2207.12598 (2022).

Imagen

Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding

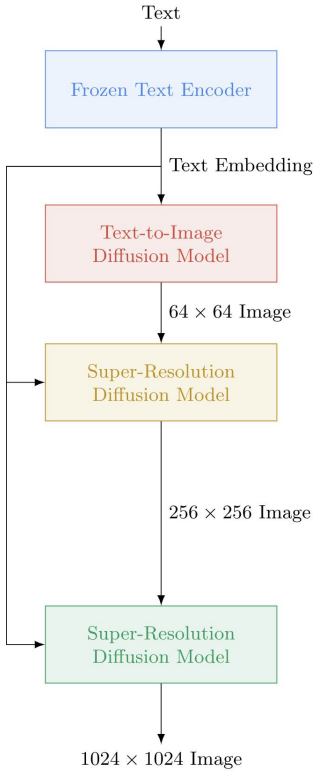
Chitwan Saharia*, William Chan*, Saurabh Saxena†, Lala Li†, Jay Whang†,
Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan,
S. Sara Mahdavi, Raphael Gontijo-Lopes, Tim Salimans,
Jonathan Ho†, David J Fleet†‡, Mohammad Norouzi*

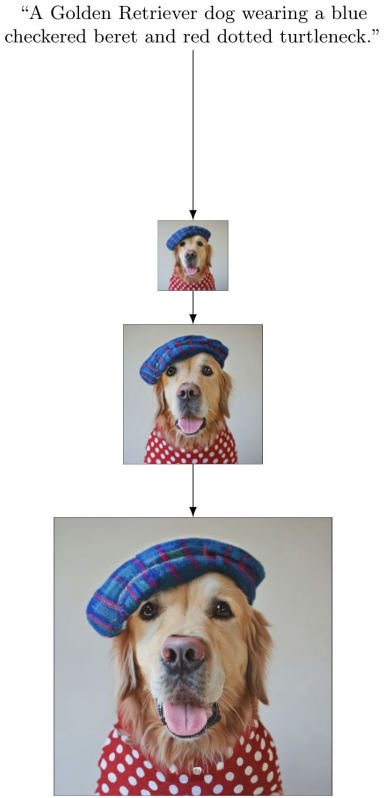
{sahariac,williamchan,mnorouzi}@google.com
{srbs,lala,jwhang,jonathanho,davidfleet}@google.com

Google Research, Brain Team
Toronto, Ontario, Canada

Saharia, Chitwan, et al. "Photorealistic text-to-image diffusion models with deep language understanding." Advances in neural information processing systems 35 (2022): 36479-36494.

Imagen





Saharia, Chitwan, et al. "Photorealistic text-to-image diffusion models with deep language understanding." Advances in neural information processing systems 35 (2022): 36479-36494.

Imagen

Sprouts in the shape of text 'Imagen' coming out of a fairytale book.

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

A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.

Teddy bears swimming at the Olympics 400m Butterfly event.

A cute corgi lives in a house made out of sushi.

A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

Saharia, Chitwan, et al. "Photorealistic text-to-image diffusion models with deep language understanding." Advances in neural information processing systems 35 (2022): 36479-36494.

DDIM

DENOISING DIFFUSION IMPLICIT MODELS

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlin,ermon}@cs.stanford.edu

Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models." arXiv preprint arXiv:2010.02502 (2020).

DDIM

- No DDPM, temos que rodar o processo de sampling t vezes.
  - Sampling demora muito.
- Como agilizar esse processo?
  - “Pulando” passos de sampling.

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}{\sqrt{\alpha_t}} \right)}_{\text{“ predicted } \boldsymbol{x}_0\text{”}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(\boldsymbol{x}_t)}_{\text{“direction pointing to } \boldsymbol{x}_t\text{”}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}} \tag{12}$$

Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models." arXiv preprint arXiv:2010.02502 (2020).

DDIM

- Reformulamos o processo de sampling pra depender de X_0, além de apenas X_{t-1}

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}{\sqrt{\alpha_t}} \right)}_{\text{“ predicted } \boldsymbol{x}_0\text{”}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(\boldsymbol{x}_t)}_{\text{“direction pointing to } \boldsymbol{x}_t\text{”}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}} \tag{12}$$

- Podemos reescrever a mesma fórmula para depender de X_{t-i}, efetivamente “pulando” i timesteps.

Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models." arXiv preprint arXiv:2010.02502 (2020).

DDIM

Figure 9: CelebA samples from DDIM with the same random $\boldsymbol{x}_T$ and different number of steps.

Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models." arXiv preprint arXiv:2010.02502 (2020).

Latent Diffusion

High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach[1] * Andreas Blattmann[1] * Dominik Lorenz[1] Patrick Esser[R] Björn Ommer[1]

[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [R]Runway ML

https://github.com/CompVis/latent-diffusion

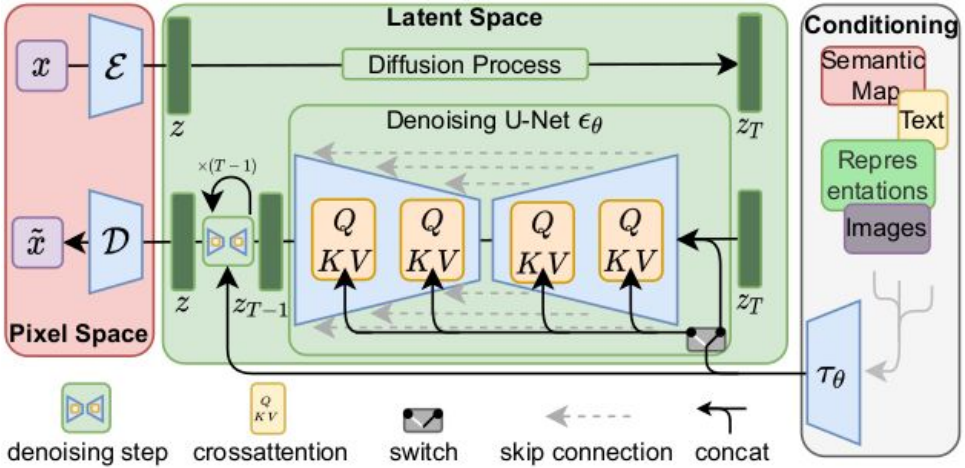
Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

Latent Diffusion

- Trabalhar no espaço de pixels é custoso.
- Fazemos nosso modelo trabalhar em um espaço latente.

Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

Latent Diffusion



Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

Latent Diffusion

Figure 27. Random samples from *LDM-4* trained on the ImageNet dataset. Sampled with classifier-free guidance [32] scale $s = 3.0$ and 200 DDIM steps with $\eta = 1.0$.

Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

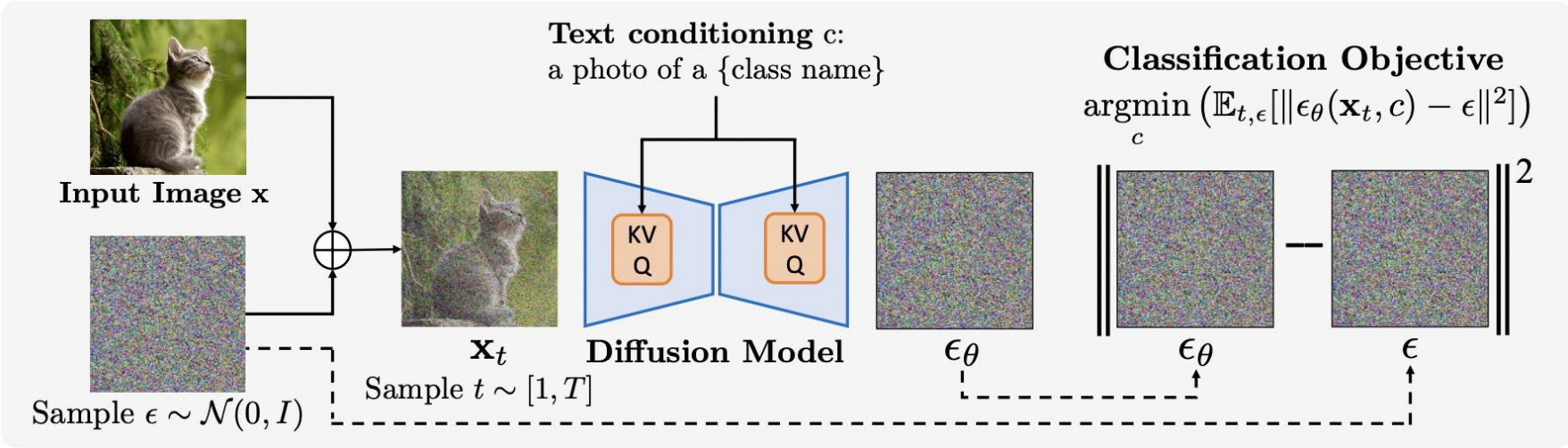
Outras Melhorias

- v-space loss [7]
  - reparametrização da loss para prever $v \equiv \alpha_t - \sigma_t x$
- Zero-PSNR schedule [8]
  - garante que X_t seja completamente ruidosa(o que não ocorre em demais schedules)
- Progressive Distillation [7]
  - Usar uma Teacher Network para aprender a realizar a difusão em N/2 passos.
- Denoising Diffusion GAN [9]
  - Substituímos o processo reverso com uma GAN
  - Podemos dar passos maiores de uma só vez.

Modelos de Difusão

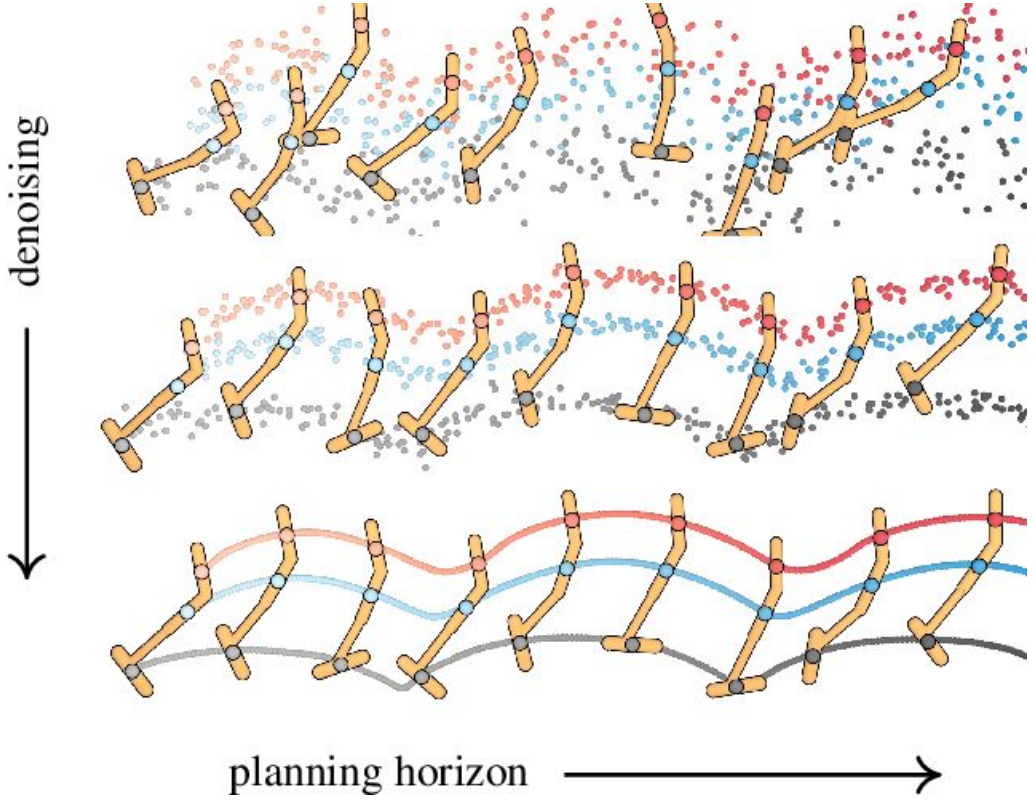
- Introdução
- Visão Geral
- Deep Denoising Probabilistic Models
- Melhorias
- **Outras Aplicações**

Your diffusion model is secretly a zero-shot classifier.



Li, Alexander C., et al. "Your diffusion model is secretly a zero-shot classifier." arXiv preprint arXiv:2303.16203 (2023).

Planning with Diffusion for Flexible Behavior Synthesis



Janner, Michael, et al. "Planning with diffusion for flexible behavior synthesis." arXiv preprint arXiv:2205.09991 (2022).

Audio Generation-DiffWave

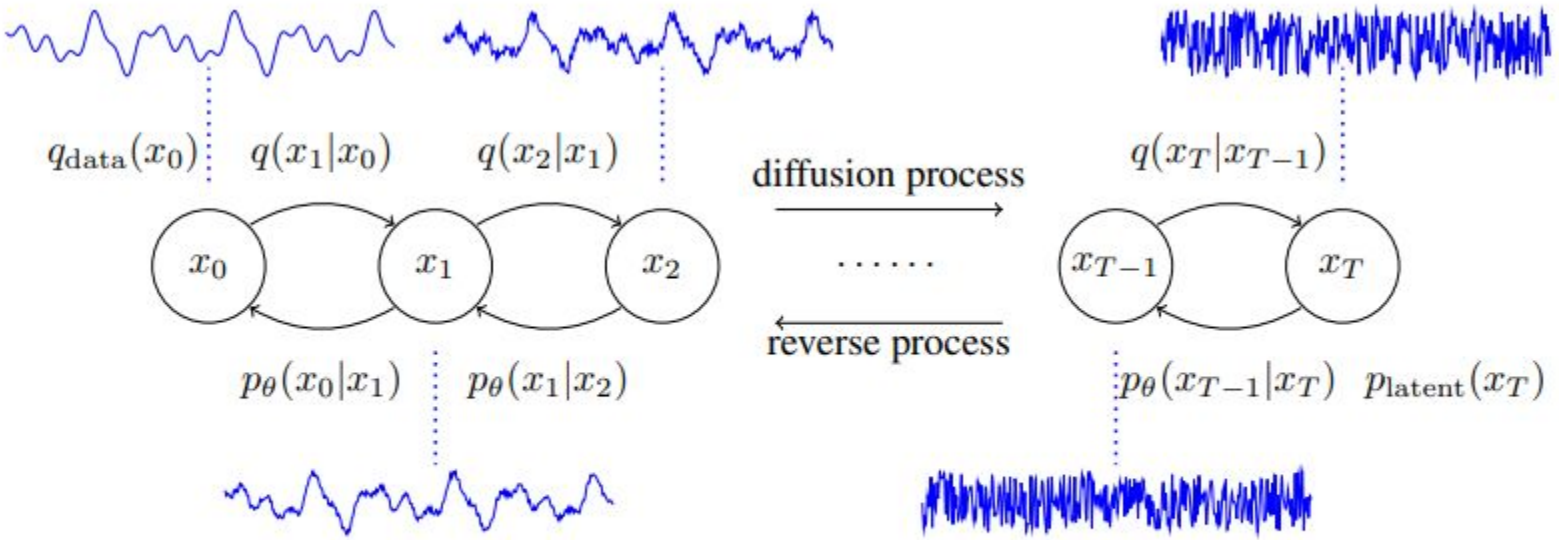
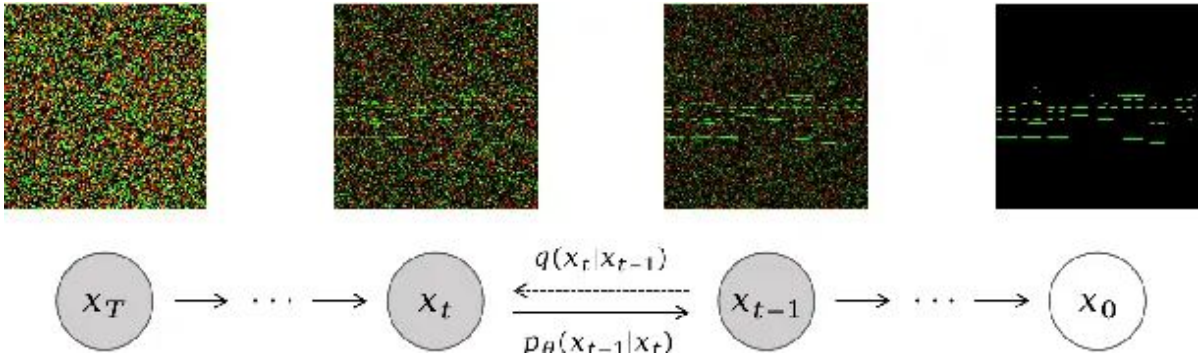


Figure 1: The diffusion and reverse process in diffusion probabilistic models. The reverse process gradually converts the white noise signal into speech waveform through a Markov chain $p_\theta(x_{t-1}|x_t)$.

Sound demos for "DiffWave: A Versatile Diffusion Model fo Audio Synthersis" (diffwave-demo.github.io)

Kong, Zhifeng, et al. "Diffwave: A versatile diffusion model for audio synthesis." arXiv preprint arXiv:2009.09761 (2020).

Polyffusion



Polyffusion: A Diffusion Model for Polyphonic Score Generation with Internal and External Controls

Min, Lejun, et al. "Polyffusion: A diffusion model for polyphonic score generation with internal and external controls." arXiv preprint arXiv:2307.10304 (2023).

DiffTransfer

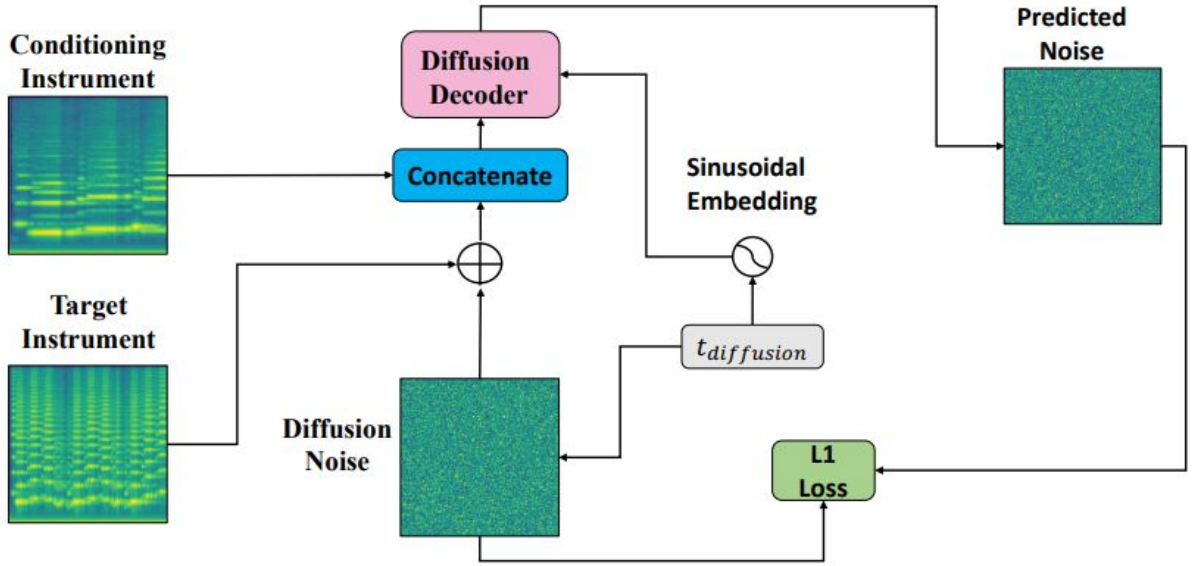


Figure 1: Training scheme of the proposed DiffTransfer technique. The target instrument spectrogram is summed with noise following a simplified cosine schedule. The decoder, conditioned on the conditioning instrument spectrogram and on the sinusoidal embedding representing the current time instant estimates the added noise. The decoder parameters are estimated by computing the L1 loss between the ground truth and the estimated diffusion noise.

DiffTransfer | Timbre Transfer using Denoising Diffusion Implicit Models (ISMIR 2023) (lucacoma.github.io)

Comanducci, Luca, Fabio Antonacci, and Augusto Sarti. "Timbre transfer using image-to-image denoising diffusion implicit models." Proceedings of the 24th International Society for Music Information Retrieval Conference, Milan, Italy, November 5-9, 2023 (ISBN: 978-1-7327299-3-3). 2024.

Referências

Algumas aulas e tutoriais:

L6 Diffusion Models (SP24) (youtube.com)

CS 198-126: Lecture 12 - Diffusion Models (youtube.com)

Diffusion Models | Paper Explanation | Math Explained - YouTube

InDepth Guide to Denoising Diffusion Probabilistic Models DDPM (learnopencv.com)

What are Diffusion Models? | Lil'Log (lilianweng.github.io)

Referências

[0] Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

[1] Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." International conference on machine learning. PMLR, 2015.

[2] Song, Yang, and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution." Advances in neural information processing systems 32 (2019).

[3] Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." Advances in neural information processing systems 34 (2021): 8780-8794.

[4] Ho, Jonathan, and Tim Salimans. "Classifier-free diffusion guidance." arXiv preprint arXiv:2207.12598 (2022).

[5] Saharia, Chitwan, et al. "Photorealistic text-to-image diffusion models with deep language understanding." Advances in Neural Information Processing Systems 35 (2022): 36479-36494.

[6] Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models." arXiv preprint arXiv:2010.02502 (2020).

[7] Salimans, Tim, and Jonathan Ho. "Progressive distillation for fast sampling of diffusion models." arXiv preprint arXiv:2202.00512 (2022).

[8] Lin, Shanchuan, et al. "Common diffusion noise schedules and sample steps are flawed." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2024.

[9] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022

[10] Xiao, Zhisheng, Karsten Kreis, and Arash Vahdat. "Tackling the generative learning trilemma with denoising diffusion gans." arXiv preprint arXiv:2112.07804 (2021).

[11] Janner, Michael, et al. "Planning with diffusion for flexible behavior synthesis." arXiv preprint arXiv:2205.09991 (2022).

[12] Kong, Zhifeng, et al. "Diffwave: A versatile diffusion model for audio synthesis." arXiv preprint arXiv:2009.09761 (2020).

[13] Min, Lejun, et al. "Polyffusion: A diffusion model for polyphonic score generation with internal and external controls." arXiv preprint arXiv:2307.10304 (2023).

[14] Comanducci, Luca, Fabio Antonacci, and Augusto Sarti. "Timbre transfer using image-to-image denoising diffusion implicit models." Proceedings of the 24th International Society for Music Information Retrieval Conference, Milan, Italy, November 5-9, 2023 (ISBN: 978-1-7327299-3-3). 2024.