DDPM及其在用户建模中的应用

周红

2025/1/3

Outline

一、浅谈扩散模型

二、DDPM背后的数学原理

- Denoising Diffusion Probabilistic Models, NIPS 2020

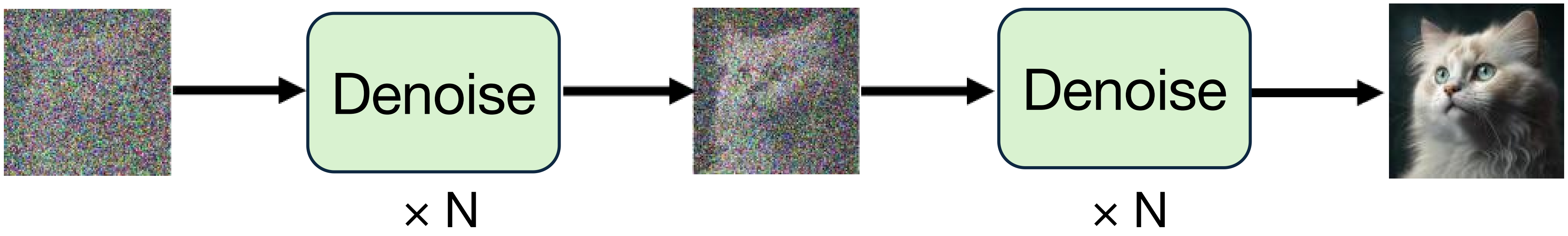
三、DDPM在用户建模中的应用

- Diffusion Recommender Model, SIGIR 2023
- Generate What You Prefer: Reshaping Sequential Recommendation via Guided Diffusion, NIPS 2023

一、浅谈扩散模型

Forward Process

Reverse Process

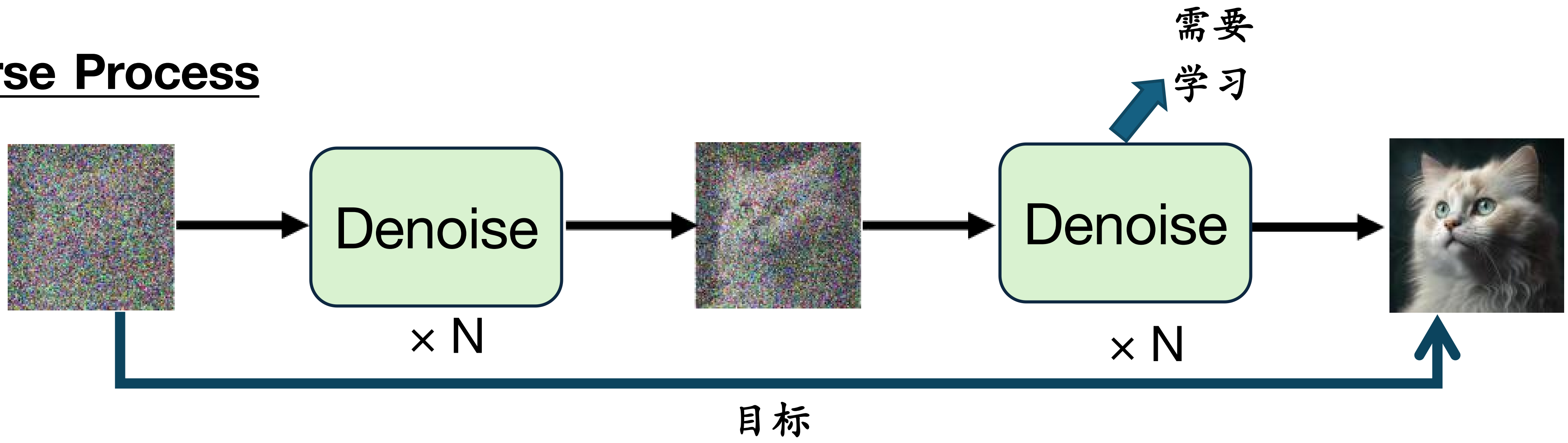


https://speech.ee.ntu.edu.tw/~hylee/ml/ml2023-course-data/DDPM%20(v7).pdf

- 目标：从随机噪声中生成符合目标分布的数据。
- 手段：加噪。
- 如何生成：对加噪后的数据进行建模，学习如何通过去噪得到符合目标分布的数据。

Forward Process

Reverse Process



二、DDPM背后的数学原理

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

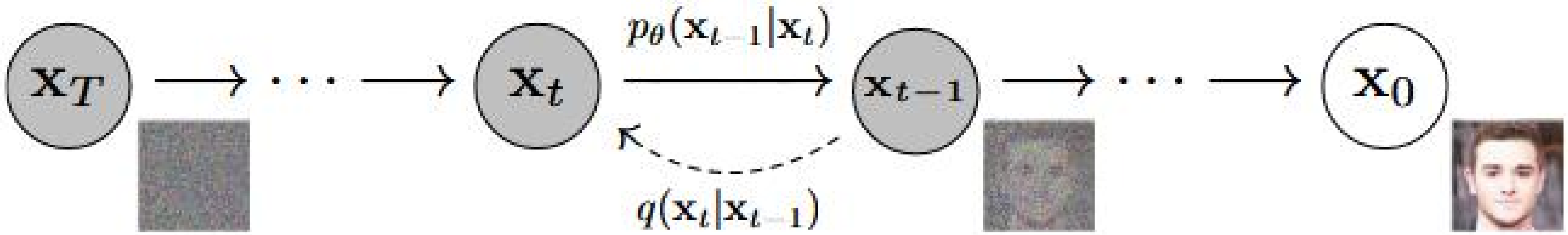
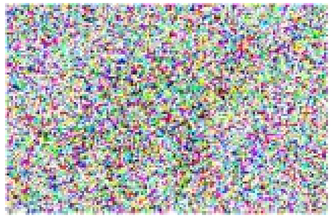


Figure 2: The directed graphical model considered in this work.

- 前向过程是一个初始状态为$\mathbf{x}_0$马尔可夫过程，其中$\mathbf{x}_0$由目标数据集中随机采样得到。
- 不断向分布中添加噪声，该加噪过程是一个高斯分布，均值和方差由 β_t 决定。
- 随着t的不断增大，最终数据分布 x_T 变为各向独立的高斯分布。

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

x_{t-1}



ε

$\sim N(0, I)$

$\xrightarrow{\beta_1, \beta_2, \ldots, \beta_T}$

固定值，越来越大

$\sqrt{1-\beta_t}$  $+ \; \beta_t$  $=$ 

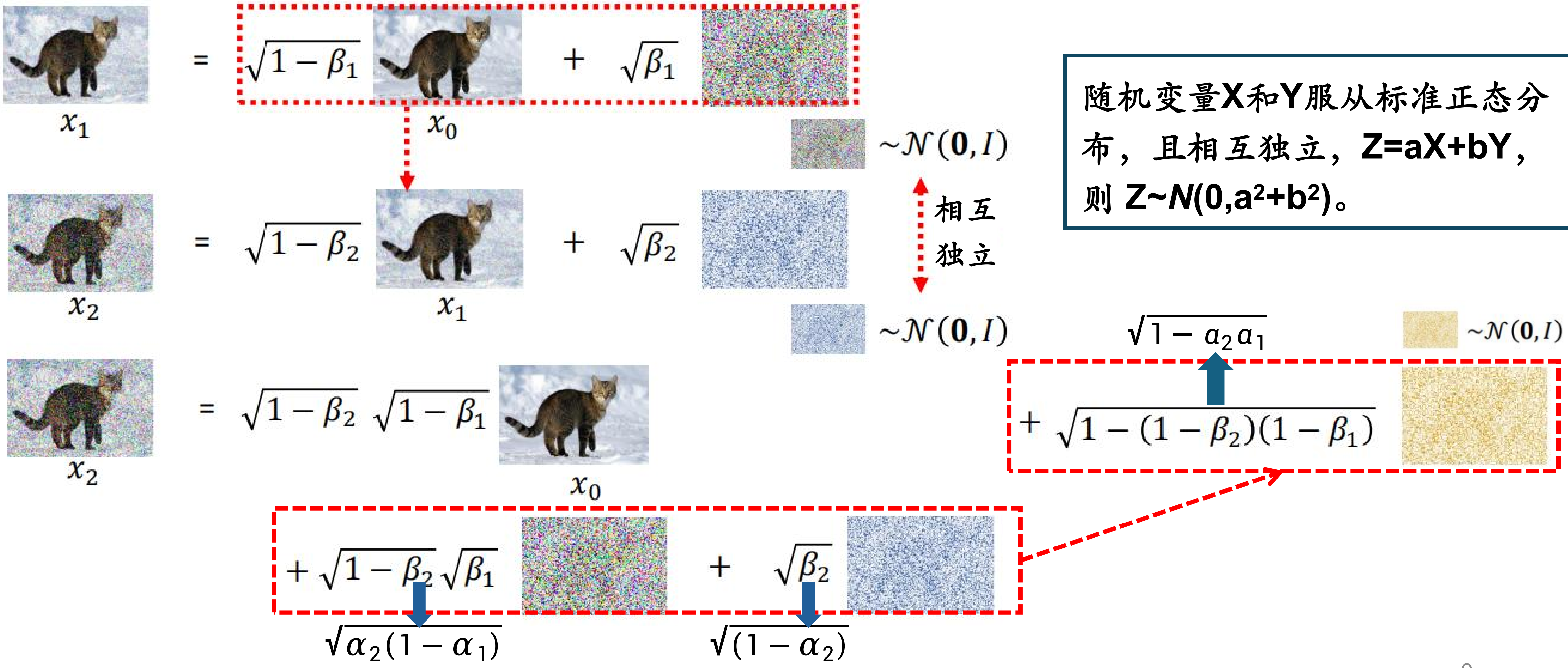
x_{t-1} $\quad$ ε $\quad$ x_t

Denoising Diffusion Probabilistic Models, NIPS 2020

1. Forward Process

- 任意时刻的分布可直接得出，不用进行迭代。

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \qquad \alpha_t := 1-\beta_t \text{ and } \bar{\alpha}_t := \prod_{s=1}^{t}\alpha_s$$



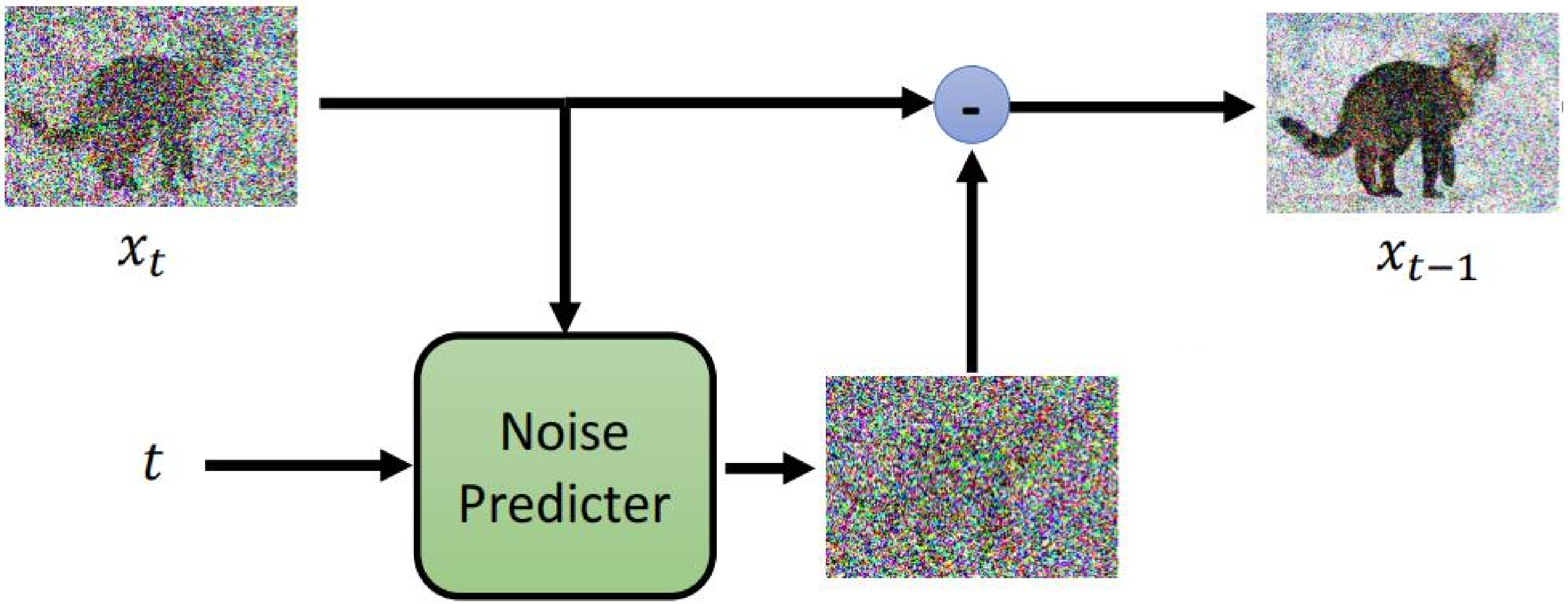
随机变量**X**和**Y**服从标准正态分布，且相互独立，**Z=aX+bY**，则 **Z~*N*(0,a²+b²)**。

Denoising Diffusion Probabilistic Models, NIPS 2020

2. Reverse Process

- 逆向过程是一个初始状态为x_T的马尔可夫过程，x_T是服从标准正态分布的高斯噪声。
- 不断从分布中去除噪声，假设该去噪过程也是一个高斯分布，均值和方差取决于模型参数θ。
- 随着t的不断减小，得到符合目标数据分布的样本。

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$



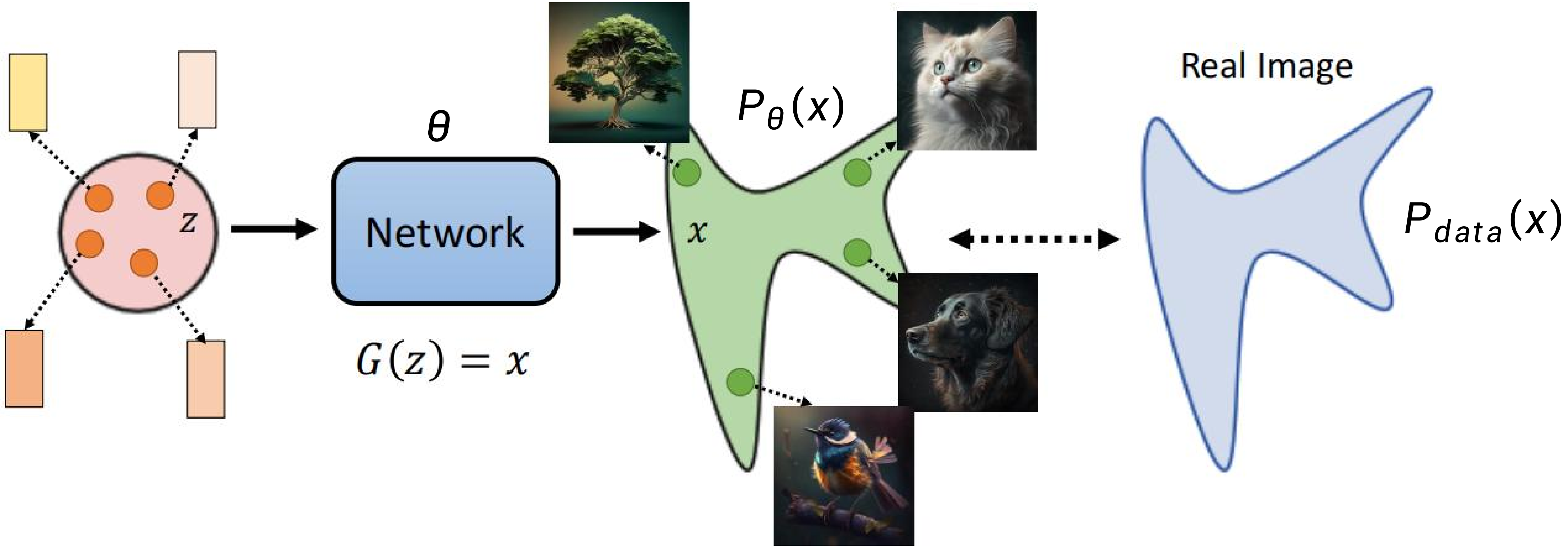
Denoising Diffusion Probabilistic Models, NIPS 2020

- 图像生成的本质目标：训练得到的模型生成的图像与真实图像是相似的。
- 采用最大似然估计，最大化$P_\theta(x_0)$。

Sample $\{x^1, x^2, \dots, x^m\}$ from $P_{data}(x)$

We can compute $P_\theta(x^i)$

$$\theta^* = arg\max_\theta \prod_{i=1}^{m} P_\theta(x^i)$$



https://speech.ee.ntu.edu.tw/~hylee/ml/ml2023-course-data/DDPM%20(v7).pdf

- 最大化对数似然函数的下界

$$\log p_\theta(x_0) = \log \int p_\theta(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{1:T}$$

$$= \log \int \frac{p_\theta(\boldsymbol{x}_{0:T}) \boxed{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} d\boldsymbol{x}_{1:T}$$

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

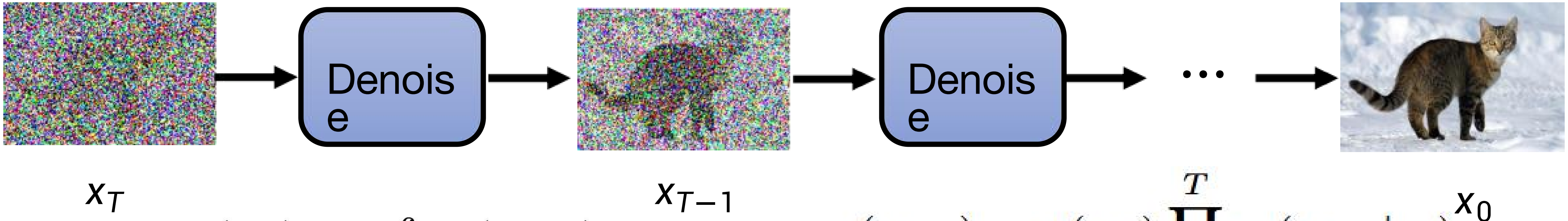
期望的定义

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

根据詹森不等式

对于一个凸函数 f 和一个随机变量 X

$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$



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

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

Understanding Diffusion Models: A Unified Perspective, arXiv 2022

- 最小化负对数似然函数的上界

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right] + \mathbb{E}_{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)}\right]$$

$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))\right]}_{\text{denoising matching term}}$$

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

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

提出t=1的项

$$= \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t>1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\right]$$

$$\boxed{q(x_t|x_{t-1})q(x_{t-1}|x_0) = q(x_t|x_0)q(x_{t-1}|x_t, x_0)}$$

$$= \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t>1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \cdot \boxed{\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}} - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{\boxed{q(\mathbf{x}_1|\mathbf{x}_0)}}\right]$$

在t>1的间隔项中消除，剩两项

①

$$= \mathbb{E}_q\left[-\log \frac{p(\mathbf{x}_T)}{\boxed{q(\mathbf{x}_T|\mathbf{x}_0)}} - \sum_{t>1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right]$$

KL散度度量两个分布差异

②

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

Denoising Diffusion Probabilistic Models, NIPS 2020

3. 学习目标

$$\mathbb{E}_q\Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\,\|\,p(\mathbf{x}_T))}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(\overset{①}{\boxed{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}}\,\|\,\overset{②}{\boxed{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}})}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Big]$$

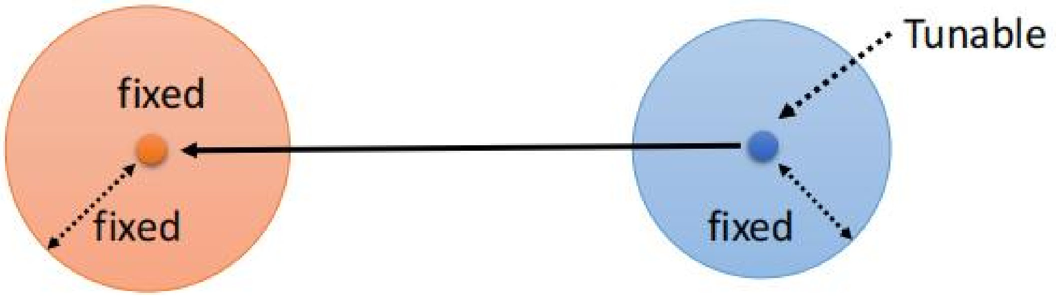
- **L_T**：前向过程确定，$p(X_T)$ 是标准正态分布，故为定值。
- **L_0**：可以理解为重构项，可以用**Monte Carlo**估计来近似和优化。
- **L_{t-1}**：需要重点关注的项。

三项均为已知的高斯分布

$$q(x_{t-1}|x_t,x_0) = \frac{q(x_{t-1},x_t,x_0)}{q(x_t,x_0)} = \frac{q(x_t|x_{t-1})q(x_{t-1}|x_0)q(x_0)}{q(x_t|x_0)q(x_0)} = \boxed{\frac{q(x_t|x_{t-1})q(x_{t-1}|x_0)}{q(x_t|x_0)}}$$

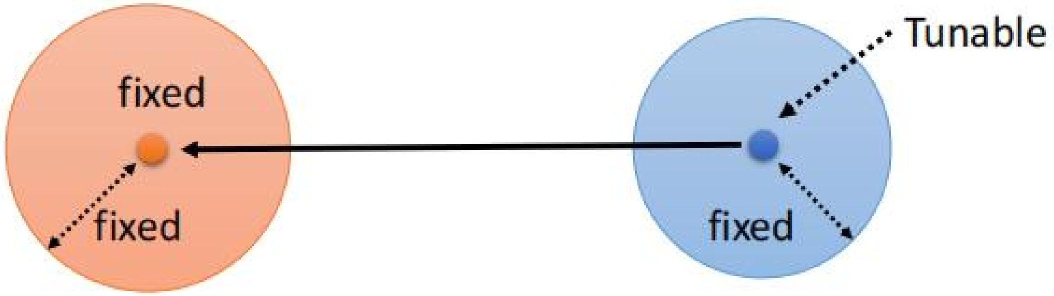
$$\propto \mathcal{N}(\boldsymbol{x}_{t-1};\underbrace{\frac{\sqrt{\alpha_t}(1-\bar\alpha_{t-1})\boldsymbol{x}_t+\sqrt{\bar\alpha_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar\alpha_t}}_{\mu_q(\boldsymbol{x}_t,\boldsymbol{x}_0)},\underbrace{\frac{(1-\alpha_t)(1-\bar\alpha_{t-1})}{1-\bar\alpha_t}\mathbf{I}}_{\boldsymbol{\Sigma}_q(t)})$$

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



- ①式均值方差固定。
- ②式中作者将方差设为固定值。
- 故两个分布的差异只与均值之差有关；直观来说，要想两个分布差异最小，就需要两分布的均值差异最小。换而言之，需要模型去预测分布①的均值。

Denoising Diffusion Probabilistic Models, NIPS 2020



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

分布①的均值

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

$$\frac{x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon}{\sqrt{\bar{\alpha}_t}} = x_0$$

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boxed{\varepsilon}\right)$$

- ①式均值方差固定。
- ②式中作者将方差设为固定值。
- 故两个分布的差异只与均值之差有关；直观来说，要想两个分布差异最小，就需要两分布的均差差异最小。换而言之，需要模型去预测分布①的均值。

- 在逆向过程中，$\mathbf{x}_t$ 是已知的，故可以只让模型去预测$\mathbf{x}_0$的值。
- 实际中直接预测$\mathbf{x}_0$的值，并将其作为最终生成结果，会导致生成质量较差。（一步一步来＞一步到位）

- 对$\mathbf{x}_0$进行转换，式子中只有噪声$\boldsymbol{\varepsilon}$是未知的，故最终让模型去预测$\boldsymbol{\varepsilon}$的值。

注：以上是直观解释，但均可通过证明得到，具体可参考
Understanding Diffusion Models: A Unified Perspective, arXiv 2022

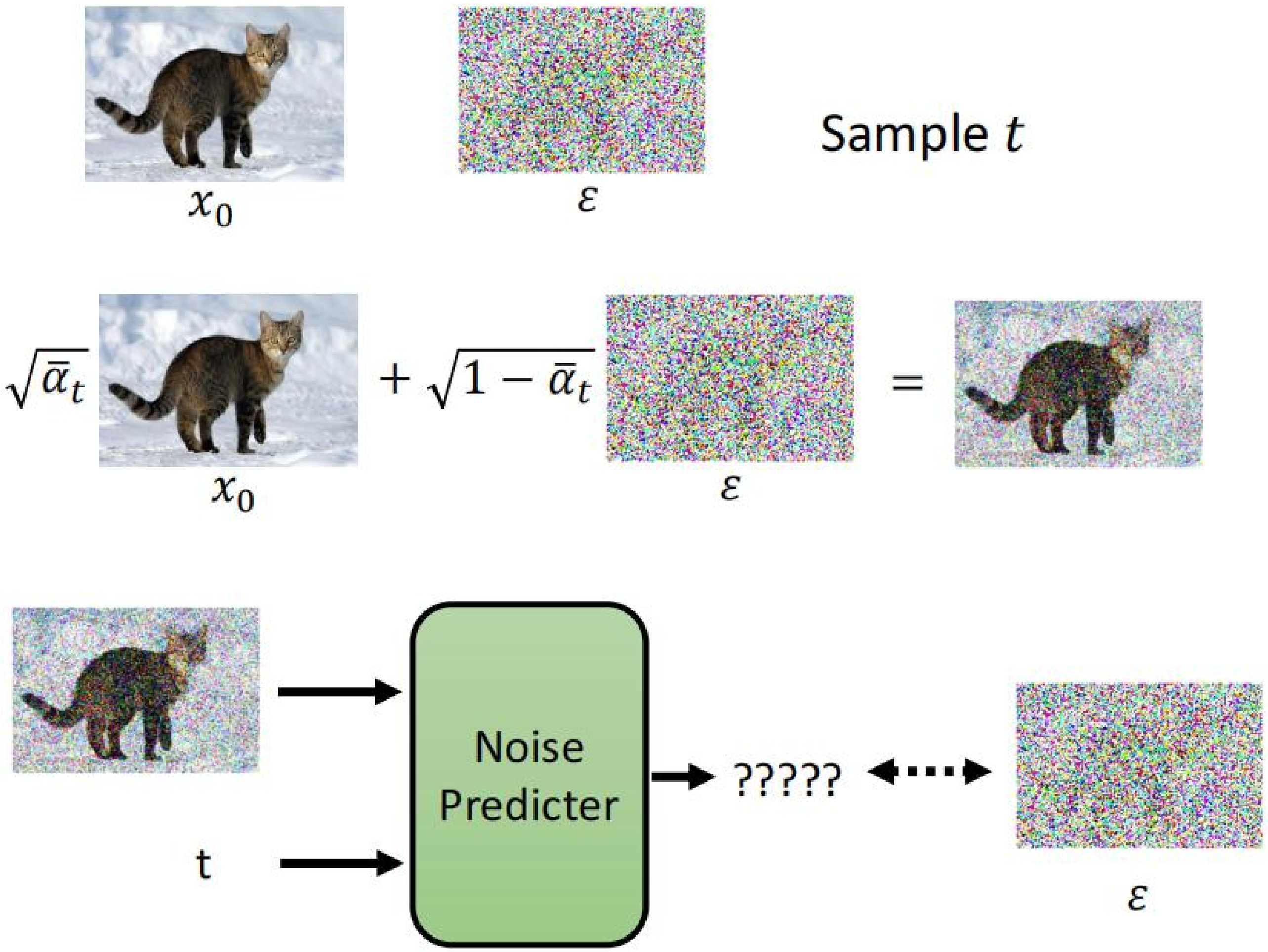
Algorithm 1 Training

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

加噪后的图像

目标噪声

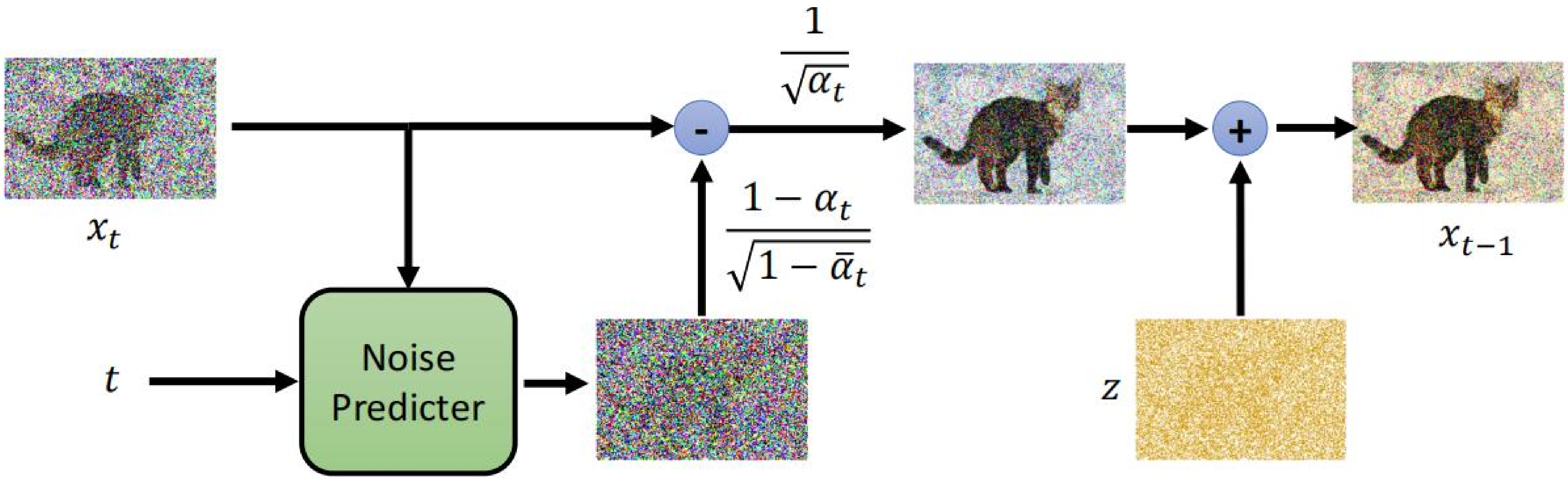
预测的噪声



Algorithm 2 Sampling

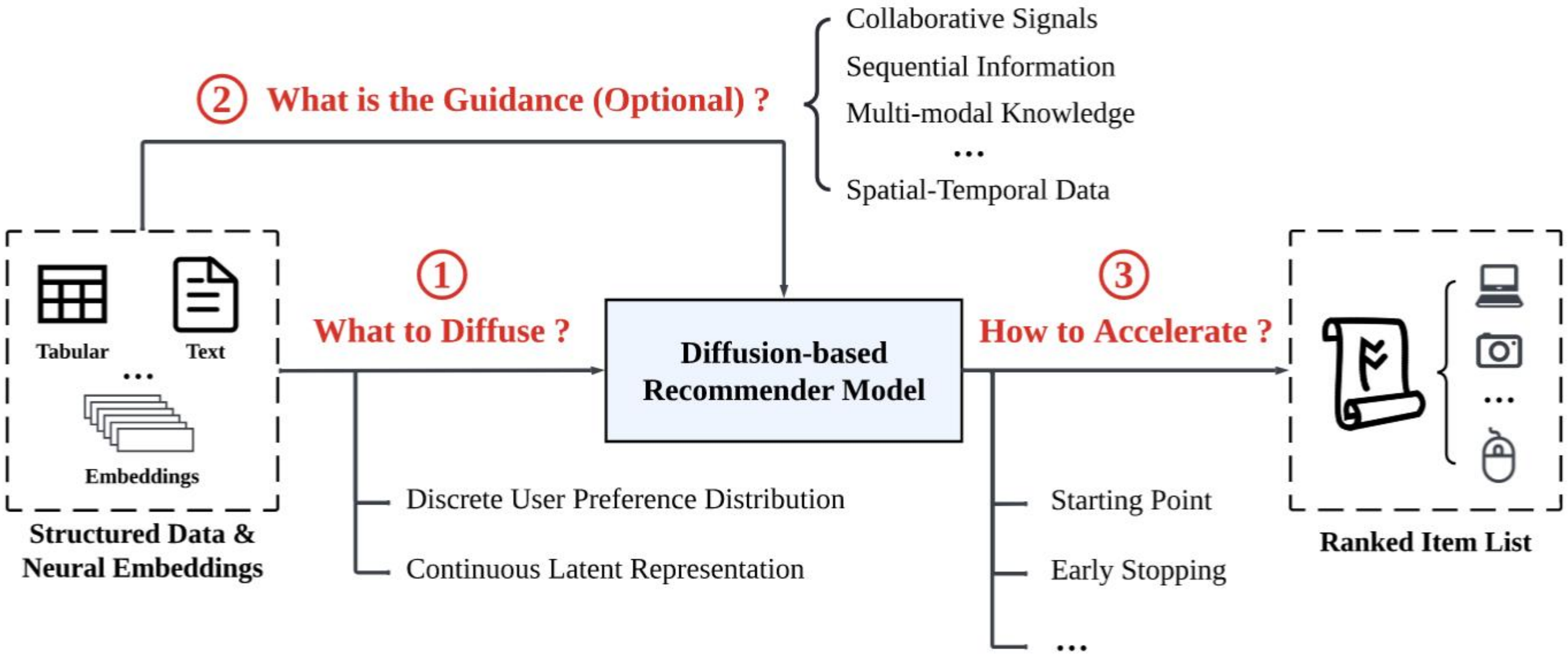
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}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

$$\boxed{\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boxed{\varepsilon} \right)}$$



三、DDPM在用户建模中的应用

1. 扩散什么？
   - 在整个项目空间上对用户-项目交互进行扩散，例如$\mathbf{x}_0 = \{0,1\}^{|\mathcal{I}|}$。
   - 在连续潜在空间中进行扩散，去噪后的潜在表示通常用作用户偏好向量或目标项目表示。
2. 用什么来指导？
   - 该指导充当条件信息，确保生成的是与每个用户相关的个性化推荐。
3. 如何加速？
   - 去噪时从有意义的输入（例如加噪后的交互矩阵）开始，而不是传统的纯高斯噪声。



A Survey on Diffusion Models for Recommender Systems, arXiv 2024

Diffusion Recommender Model

Wenjie Wang
wenjiewang96@gmail.com
National University of Singapore

Yiyan Xu
yiyanxu24@gmail.com
University of Science and Technology of China

Fuli Feng*
fulifeng93@gmail.com
University of Science and Technology of China

Xinyu Lin
xylin1028@gmail.com
National University of Singapore

Xiangnan He
xiangnanhe@gmail.com
University of Science and Technology of China

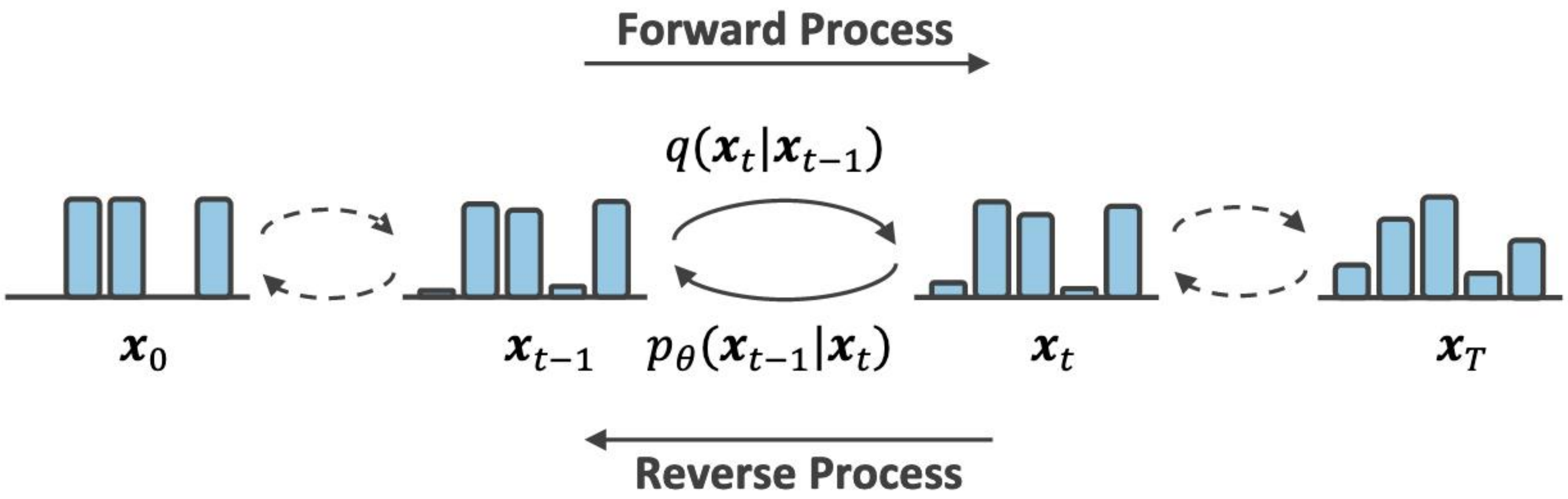
Tat-Seng Chua
dcscts@nus.edu.sg
National University of Singapore

论文动机:

- **现有生成模型的局限性:** 变分自编码器（VAE）和生成对抗网络（GAN）等生成模型常用于模拟用户交互的生成过程，但它们存在固有缺陷。
  - VAE：表示能力受限，可能无法捕捉用户交互的复杂性。
  - GAN：训练过程不稳定，难以收敛。
- **扩散模型强大的生成能力。**

扩散什么？

- 给定用户u，项目集I,交互历史为$x_u = \{x_u^1, x_u^2, ..., x_u^{|I|}\}$，$x_u^i$ 为1表示用户u与项目i进行了交互，为0则表示没有交互。设$x_0 = x_u$为初始状态。



个性化信息保留&加速

- 首先对x_0进行加噪，变为$x_{T'}$，然后设置 $\hat{x}_T = x_{T'}$,并以$\hat{x}_T$为输入（即不是从传统的纯高斯噪声开始）来执行反向去噪操作。

- 训练过程

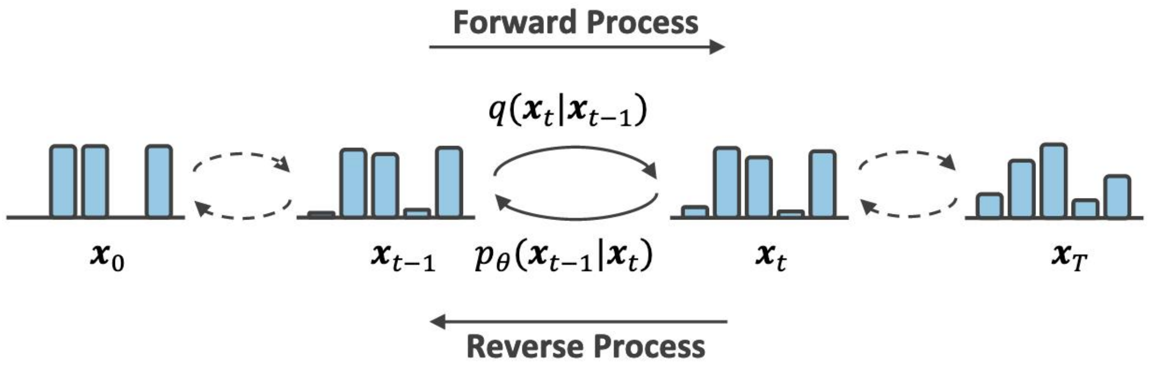
Algorithm 1 DiffRec Training

Input: all users' interactions $\bar{X}$ and randomly initialized θ.

repeat
  Sample a batch of users' interactions $X \subset \bar{X}$.
  for all $x_0 \in X$ **do**
    Sample $t \sim \mathcal{U}(1, T)$ or $t \sim p_t$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
    Compute $\boldsymbol{x}_t$ given $\boldsymbol{x}_0$, t, and ϵ via $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ in Eq. (3);
    Compute $\mathcal{L}_t$ by Eq. (11) if $t > 1$, otherwise by Eq. (12);
    Take gradient descent step on $\nabla_\theta \mathcal{L}_t$ to optimize θ;
until converged

Output: optimized θ.

对原始交互进行加噪



$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}), \tag{3}$$

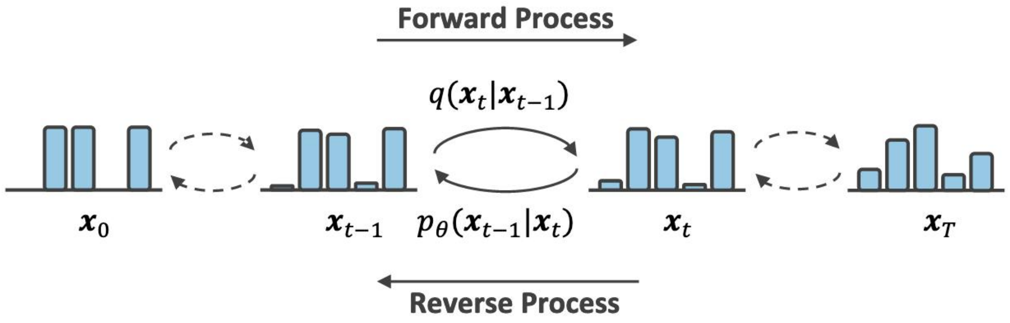
$$\mathcal{L}_t = \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\frac{1}{2}\left(\frac{\bar{\alpha}_{t-1}}{1-\bar{\alpha}_{t-1}} - \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}\right)\|\boxed{\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0}\|_2^2\right], \tag{11}$$

$$\begin{aligned}\mathcal{L}_1 &\triangleq -\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)] \\ &= \mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\|\boxed{\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_1, 1) - \boldsymbol{x}_0}\|_2^2],\end{aligned} \tag{12}$$

训练参数为$\boldsymbol{\theta}$的模型，模型根据加噪后的交互和时间步来对$\mathbf{x_0}$进行预测

Diffusion Recommender Model, SIGIR 2023

- 推理过程



Algorithm 2 DiffRec Inference

Input: θ and the interaction history $\boldsymbol{x}_0$ of user u.

1: Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

2: Compute $\boldsymbol{x}_{T'}$ given $\boldsymbol{x}_0$, T', and $\boldsymbol{\epsilon}$ via Eq. (3), and set $\hat{\boldsymbol{x}}_T = \boldsymbol{x}_{T'}$.

3: **for** $t = T, \dots, 1$ **do**

4: $\quad \hat{\boldsymbol{x}}_{t-1} = \boldsymbol{\mu}_\theta(\hat{\boldsymbol{x}}_t, t)$ calculated from $\hat{\boldsymbol{x}}_t$ and $\hat{\boldsymbol{x}}_\theta(\cdot)$ via Eq. (10);

Output: the interaction probabilities $\hat{\boldsymbol{x}}_0$ for user u.

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}), \tag{3}$$

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\boxed{\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t, t)}, \tag{10}$$

对 $\mathbf{x}_0$ 进行预测

使用对原始交互加噪后的样本作为去噪时的输入

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

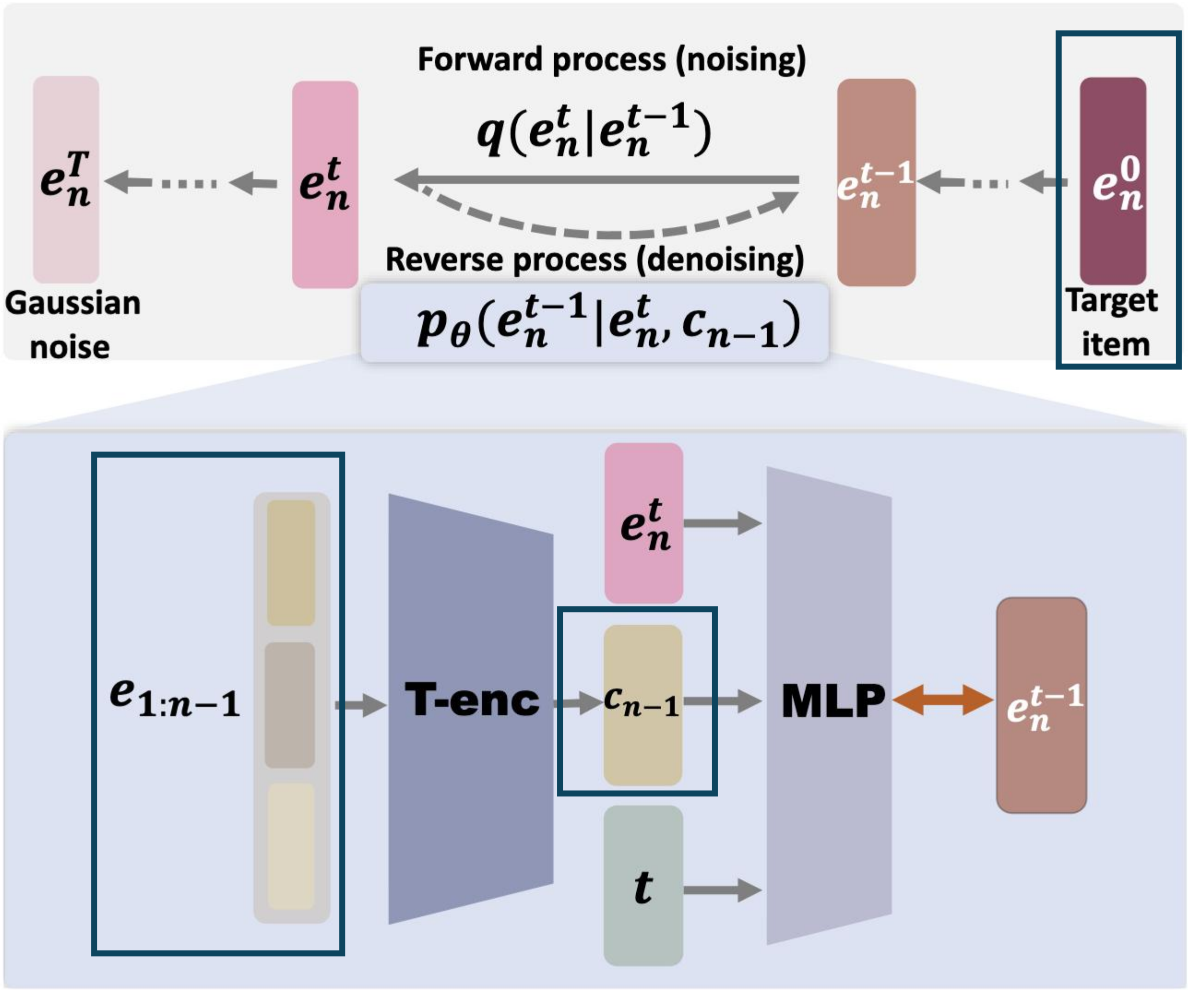
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

Generate What You Prefer: Reshaping Sequential Recommendation via Guided Diffusion

Zhengyi Yang[‡] Jiancan Wu[‡*] Zhicai Wang[‡] Yancheng Yuan[§*] Xiang Wang[‡†] Xiangnan He[‡†]
[‡]University of Science and Technology of China
[§]The Hong Kong Polytechnic University
{yangzhy,wangzhic}@mail.ustc.edu.cn
{wujcan,xiangwang1223,xiangnanhe}@gmail.com
yancheng.yuan@polyu.edu.hk

论文动机：解决当前序列推荐系统中的两大局限性。

- **与人类行为的差异：**传统的序列推荐模型采用"学习分类"的范式，通过负采样添加负例，训练模型区分用户是否偏好某项物品。而人类在选择时，往往会在脑海中构想出理想的物品，然后寻找与之匹配的选项。
- **负采样的局限性：**可能引入噪声，削弱对用户真实偏好的学习效果。



扩散什么？

- 通常，每个项目$v \in I$最初被转换为其相应的嵌入向量 e。因此交互序列可以表示为 $e_{1:n-1} = [e_1, e_2, ..., e_{n-1}]$，使用扩散模型来生成用户可能选择的下一个项目$e_n^0$。

用什么来指导?

- 需要解决个性化问题，提出用相应的历史交互序列来指导去噪过程。
- 用户在进行购买时，倾向于在他们的头脑中创建一个理想化的项目，而这个项目的生成与用户交互历史是高度相关的。

Generate What You Prefer: Reshaping Sequential Recommendation via Guided Diffusion, NIPS 2023

Algorithm 1 Training phase of DreamRec

repeat
  $\mathbf{e}_n^0, \mathbf{e}_{1:n-1} \sim \mathcal{D}$ ▷ Sample and embed a data from training set.
  $\mathbf{c}_{n-1} = \text{T-enc}(\mathbf{e}_{1:n-1})$ ▷ Encode interaction sequence.
  With probability p_u: $\mathbf{c}_{n-1} = \Phi$ ▷ Perform unconditional training with probability p_u.
  $t \sim \text{Uniform}(\{1, \ldots, T\})$ ▷ Sample diffusion step.
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample Gaussian noise.
  $\mathbf{e}_n^t = \sqrt{\bar{\alpha}_t}\mathbf{e}_n^0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ ▷ Corrupt the traget item with Gaussian noise.
  $\theta = \theta - \mu\nabla_\theta \left\|\mathbf{e}_n^0 - \boxed{f_\theta(\mathbf{e}_n^t, \mathbf{c}_{n-1}, t)}\right\|^2$ ▷ Take gradient descent step, μ is the step size.
until converged

对$\boldsymbol{e}_{\boldsymbol{n}}^{\boldsymbol{0}}$，即目标项目进行预测

- 同时训练无条件去噪扩散模型$\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{e}_{\boldsymbol{n}}^{\boldsymbol{t}}, \boldsymbol{t})$和利用历史交互作为条件的条件去噪扩散模型 $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{e}_{\boldsymbol{n}}^{\boldsymbol{t}}, \boldsymbol{c}_{\boldsymbol{n-1}}, \boldsymbol{t})$
- 有条件和无条件的权衡——个性化和多样化推荐的权衡

Generate What You Prefer: Reshaping Sequential Recommendation via Guided Diffusion, NIPS 2023

Algorithm 2 Generation phase of DreamRec

1: $\mathbf{e}_{1:n-1} \sim \mathcal{D}_t$ ▷ Sample and embed a data from testing set.
2: $\mathbf{e}_n^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample Gaussian noise.
3: $\mathbf{c}_{n-1} = \text{T-enc}(\mathbf{e}_{1:n-1})$ ▷ Encode interaction sequence.
4: **for** $t = T, \ldots, 1$ **do** ▷ Denoise for T steps.
5: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$ ▷ Sample denoising variance.
6: $\tilde{f}_\theta(\mathbf{e}_n^t, \mathbf{c}_{n-1}, t) = (1+w)\, f_\theta(\mathbf{e}_n^t, \mathbf{c}_{n-1}, t) - w\, f_\theta(\mathbf{e}_n^t, \Phi, t)$ ▷ Control the strength of guidance.
7: $\mathbf{e}_n^{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boxed{\tilde{f}_\theta(\mathbf{e}_n^t, \mathbf{c}_{n-1}, t)} + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{e}_n^t + \sqrt{\tilde{\beta}_t}\mathbf{z}$ ▷ Denoise for one step.
8: **end for**
9: **return** $\mathbf{e}_n^0$

对目标项目
进行预测

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

- 相应地，在推理阶段也需要进行有条件和无条件的权衡。
- w 是控制 c_{n-1} 强度的超参数，较高的w值可以增强个性化指导，但可能会破坏扩散的泛化能力，从而降低生成的目标项的质量。

Thanks