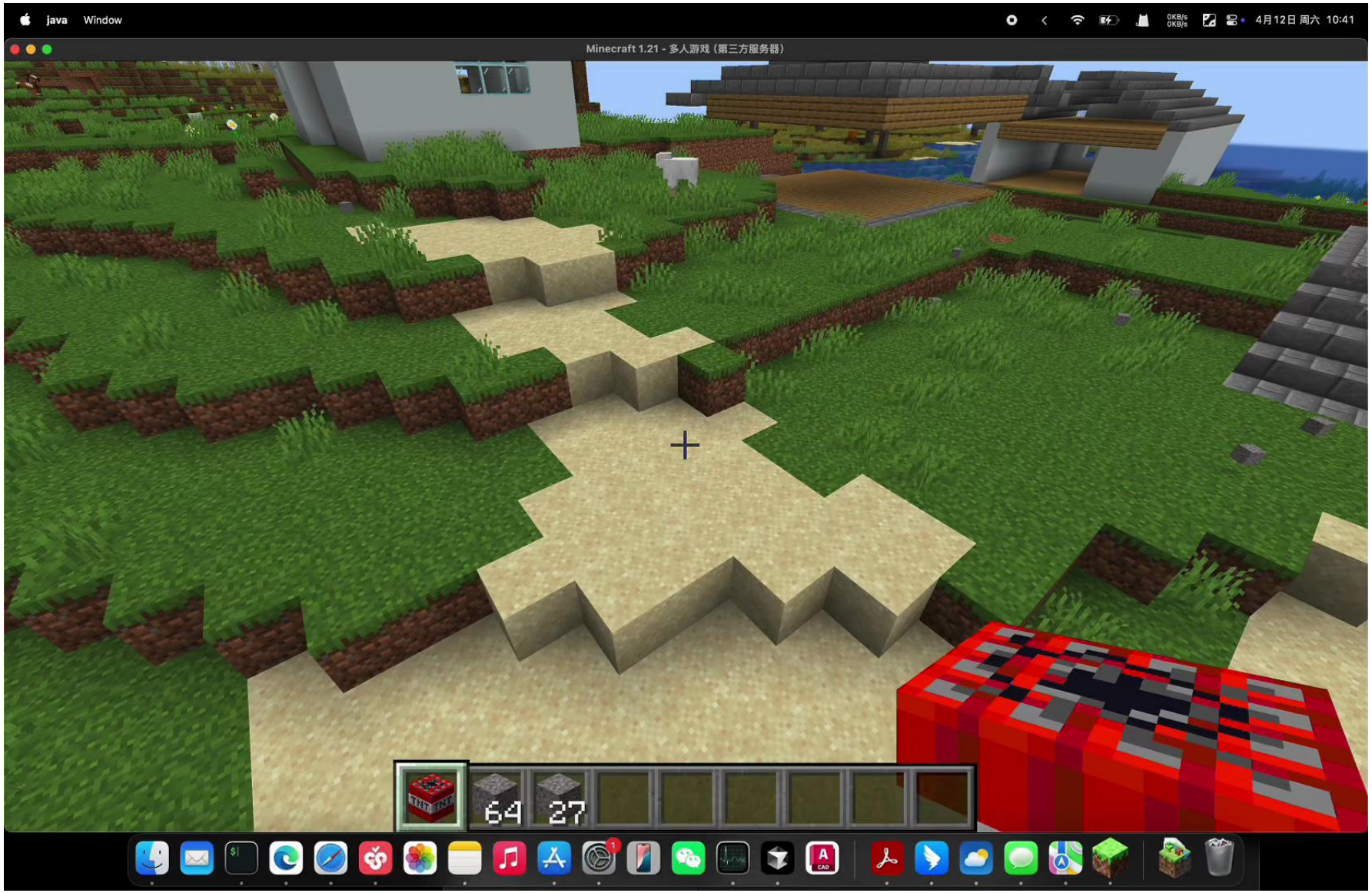

Diffusion Model
A brief introduction & Applications

Inspiration



Structure

Introduction

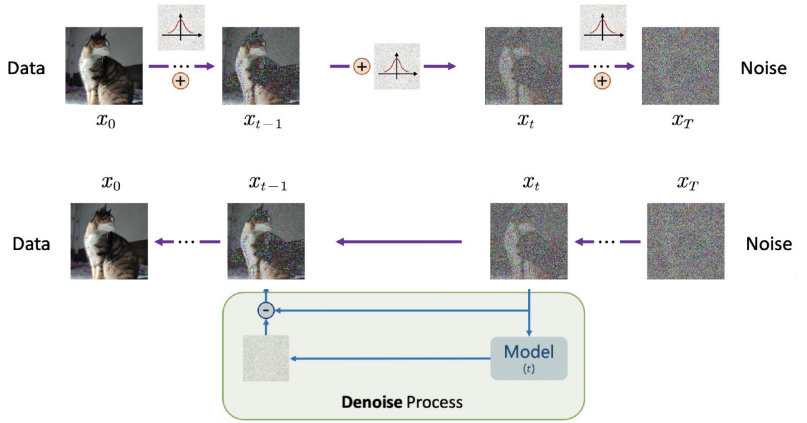
Application

What is Diffusion Model?

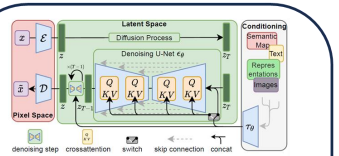
前向加噪

反向去噪

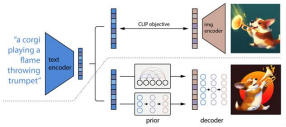
模型训练



How Diffusion Model can be applied in different fields?

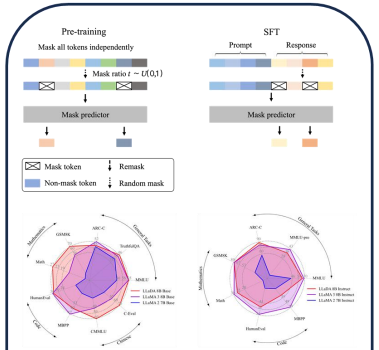


Stable Diffusio(LDM)

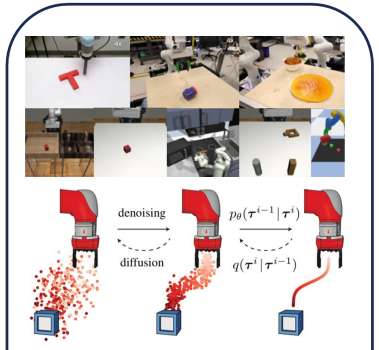


DALLE 2 By OpenAI

TXT2IMG



LLM



Robotics

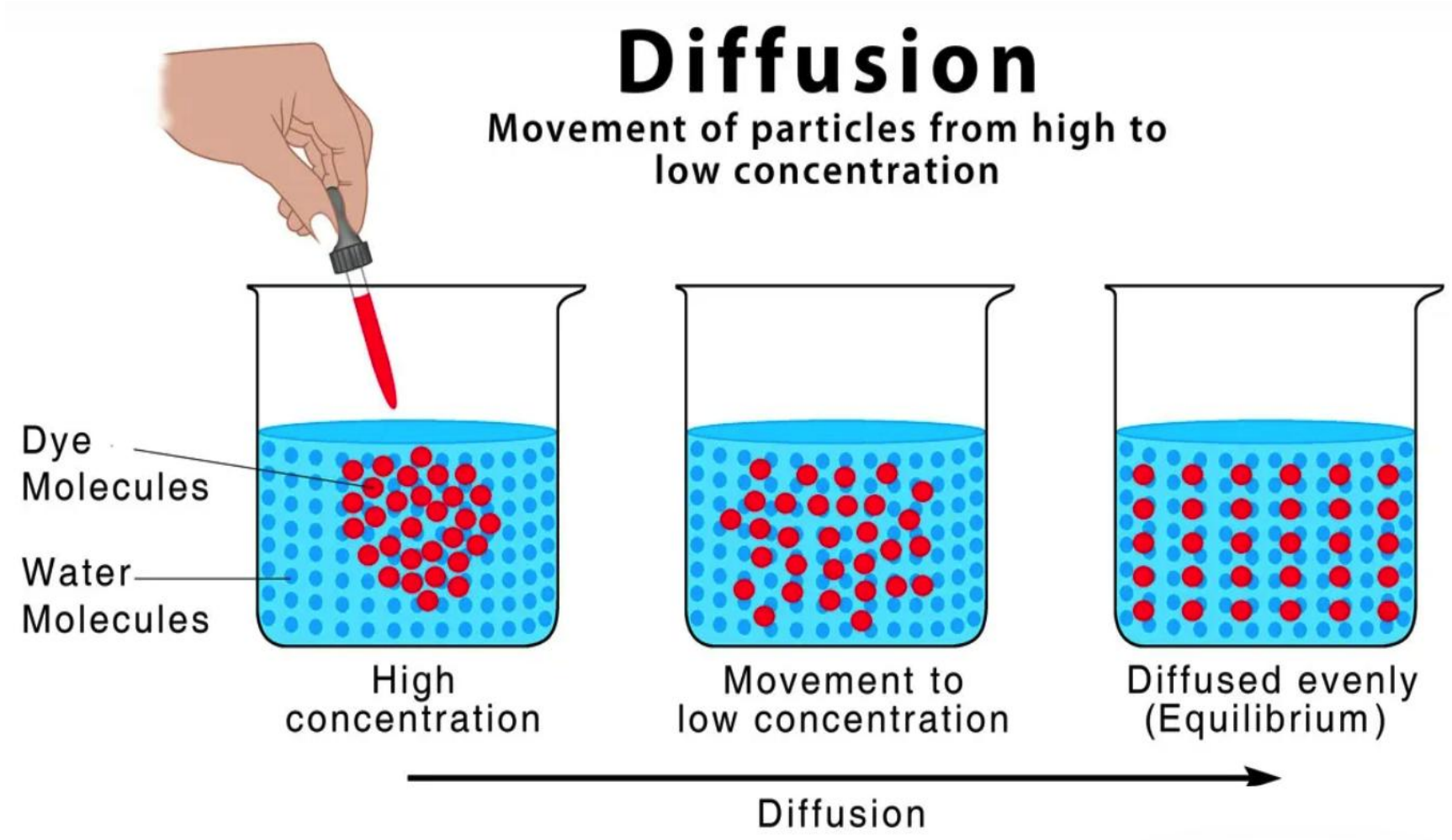
Brief Introduction

前向加噪

反向去噪

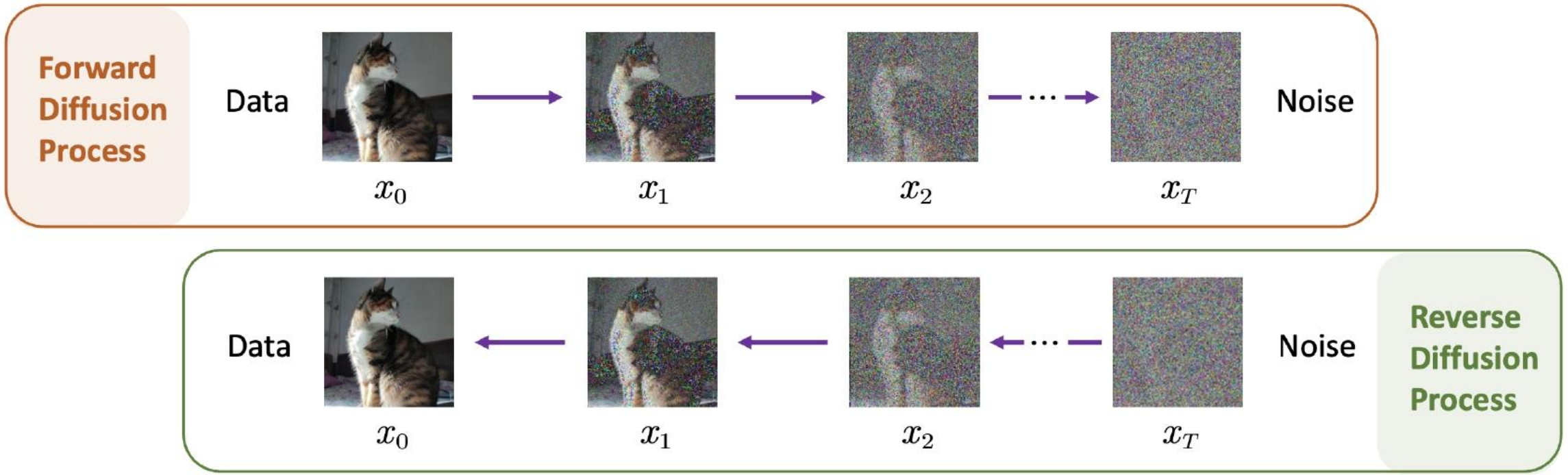
模型训练

什么是diffusion

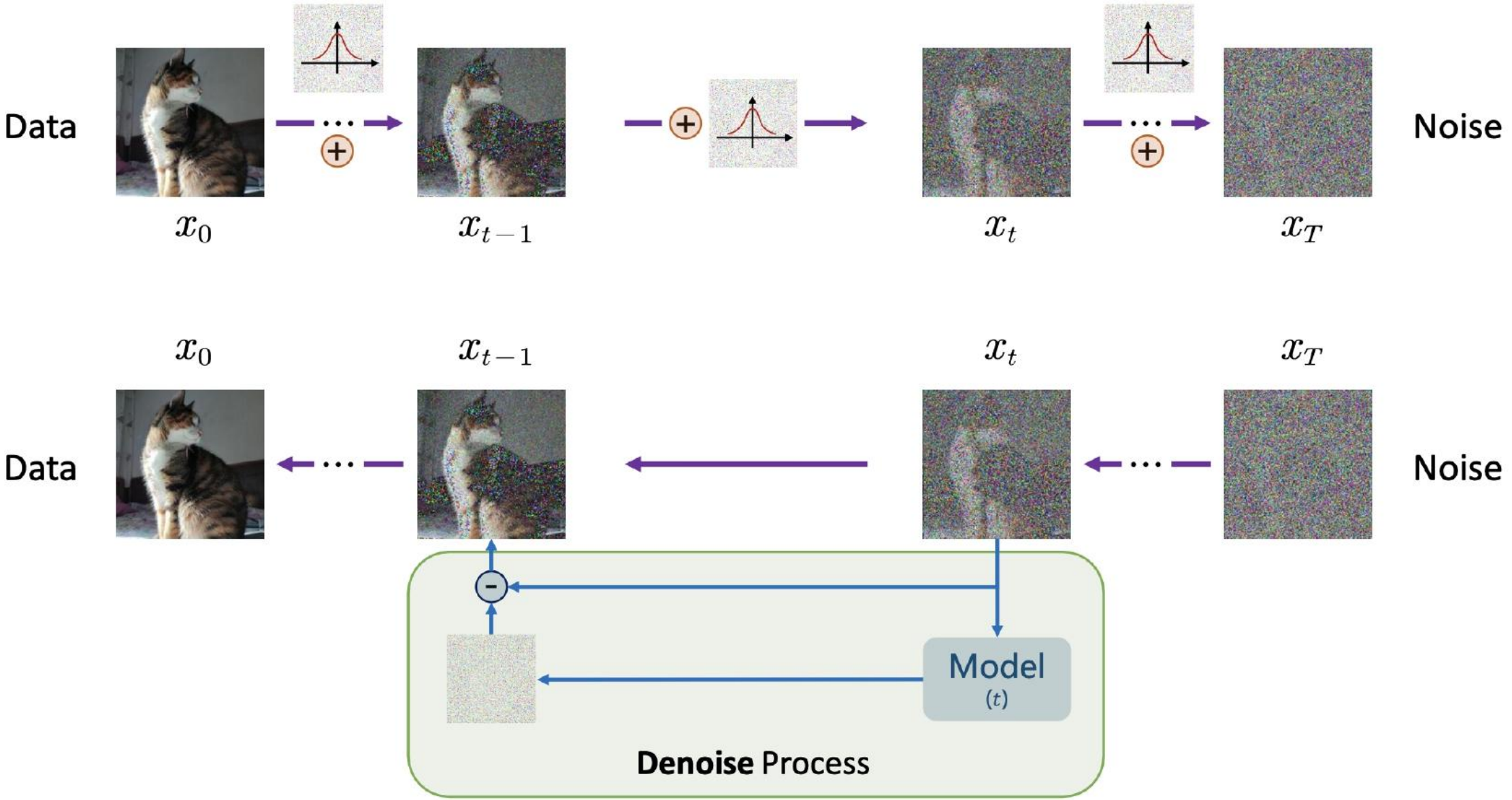


What is Diffusion Model?

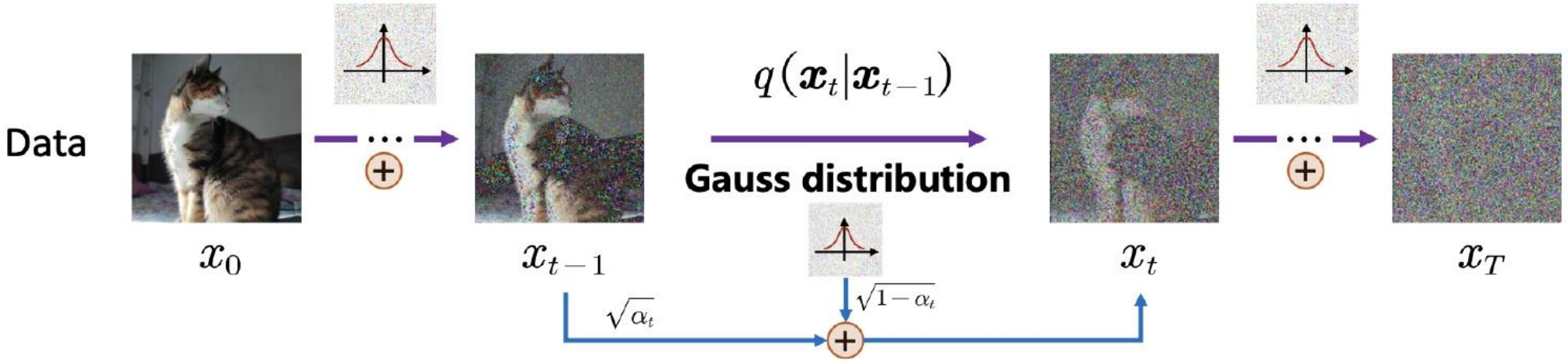
Denoising Diffusion Probabilistic Models

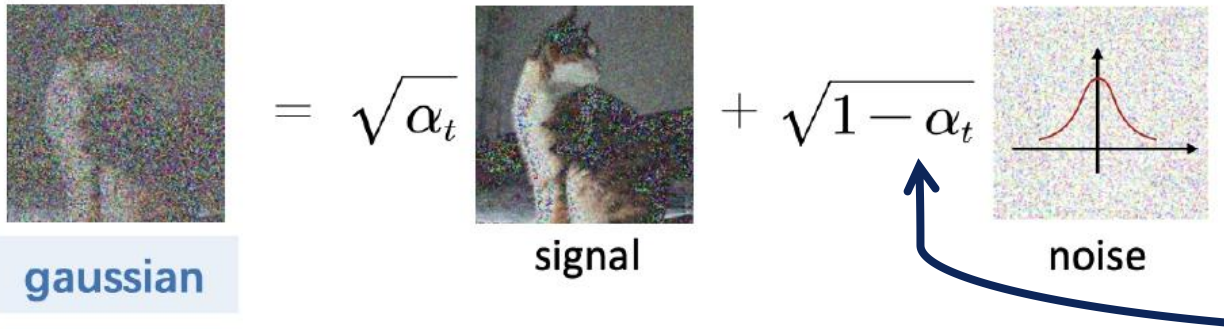


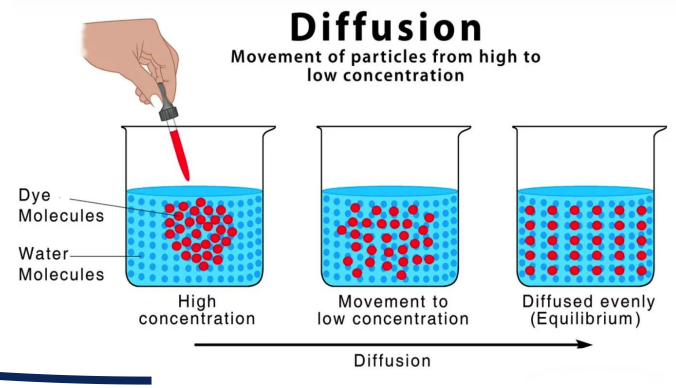
What is Diffusion Model?



Forward Diffusion Process







$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\varepsilon_{t-1}, \quad \varepsilon \sim \mathcal{N}(0, I)$$

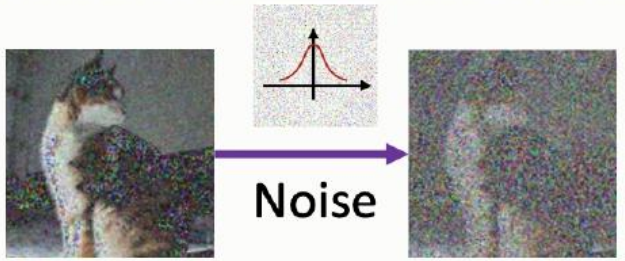
$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t) I)$$

Forward Diffusion Process

①Forward (closed-form)

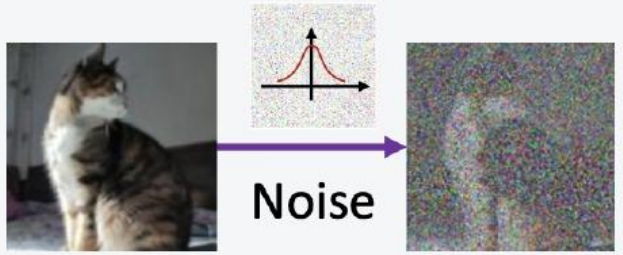
$$
\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\varepsilon}_{t-1} \\
&= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\varepsilon}_{t-2}\right) + \sqrt{1-\alpha_t}\boldsymbol{\varepsilon}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \left(\sqrt{\alpha_t(1-\alpha_{t-1})}\boldsymbol{\varepsilon}_{t-2} + \sqrt{1-\alpha_t}\boldsymbol{\varepsilon}_{t-1}\right) \\
&\vdots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, I)
\end{aligned}
$$

Where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$



x_{t-1} x_t

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\,\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\,\boldsymbol{\varepsilon}_t$$

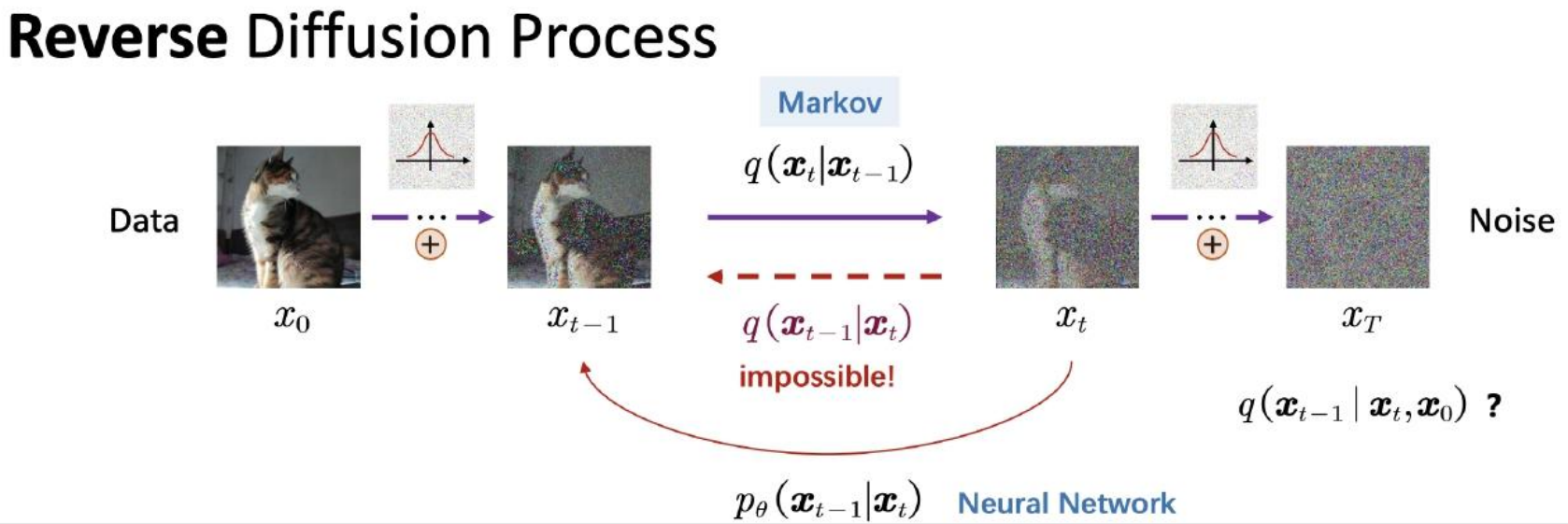


x_0 x_t

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\varepsilon}_t$$

$$\bar{\alpha}_t = \alpha_1\alpha_2\ldots\alpha_t$$

Reverse Diffusion Process



Assume: the output is gaussian

Target Distribution: $q(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1}; \mu_t(x_t), \Sigma_t(x_t)\right)$

Approximated Distribution: $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$

What is $q(x_{t-1} \mid x_t, x_0)$

x_t x_{t-1} x_0

Markov性+贝叶斯

$$P(A|B) = \frac{P(AB)}{P(B)}$$

$$P(ABC) = P(A)P(B|A)P(C|AB)$$

Assume: Markov

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

$$\propto \exp\left(-\frac{1}{2}\left(\frac{(x_t - \sqrt{\alpha_t}x_{t-1})^2}{\beta_t} + \frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t}\right)\right)$$

$$= \exp\left(-\frac{1}{2}\left(\underbrace{\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)}_{x_{t-1}\text{方差}}x_{t-1}^2 - \underbrace{\left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)}_{x_{t-1}\text{均值}}x_{t-1} + \underbrace{C(x_t, x_0)}_{\text{与}x_{t-1}\text{无关}}\right)\right)$$

$$q(x_t \mid x_{t-1}) \sim \mathcal{N}\left(x_t; \sqrt{\alpha_t}x_{t-1}, 1-\alpha_t\right)$$

$$q(x_{t-1} \mid x_0) \sim \mathcal{N}\left(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}}x_0, 1-\bar{\alpha}_{t-1}\right)$$

$$q(x_t \mid x_0) \sim \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, 1-\bar{\alpha}_t\right)$$

高斯概率密度函数 $\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \exp\left(-\frac{1}{2}\left(\frac{1}{\sigma^2}x^2 - \frac{2\mu}{\sigma^2}x + \frac{\mu^2}{\sigma^2}\right)\right)$

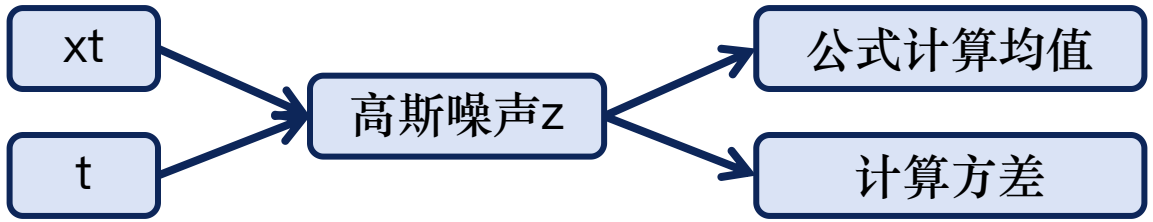
Remove x_0

③Reverse If we know x_0

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t}}_{\mu_q(\mathbf{x}_t, t)}, \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}_{\sum_q(t)} I\right)$$

$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon_t \Rightarrow x_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon_t}{\sqrt{\bar{\alpha}_t}}$ 正向加噪过程的公式移项

$\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t} \Rightarrow \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_t\right)$ 神经网络预测的噪声



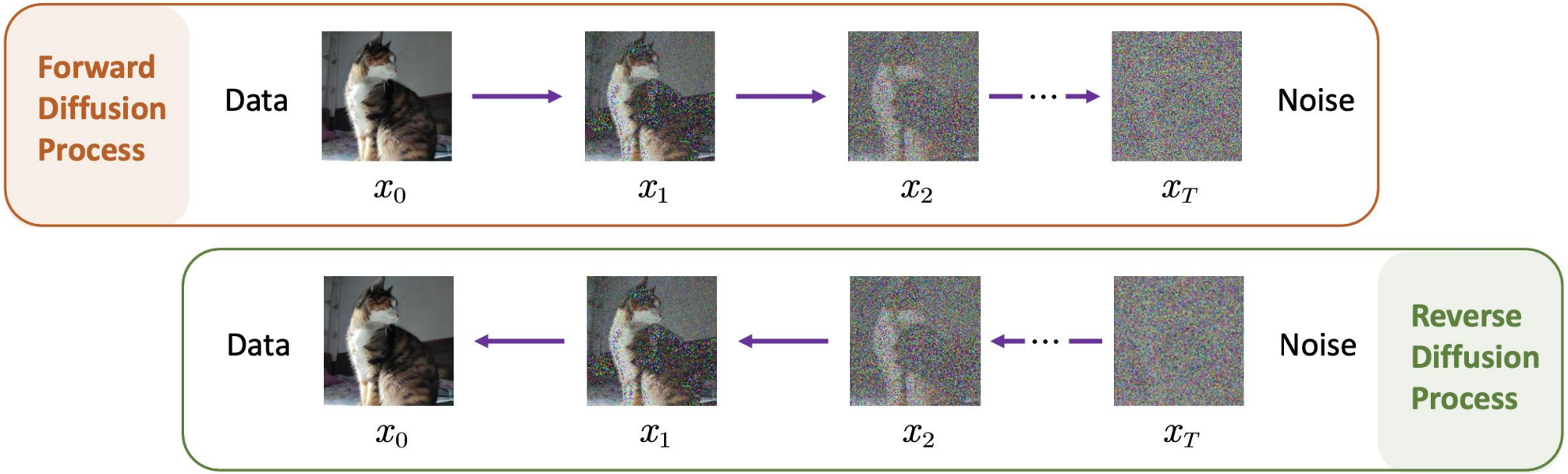
$$\tilde{\mu}_t = \frac{1}{\sqrt{a_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{a}_t}}\bar{z}_t)$$

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

组合得到反向公式

Summary

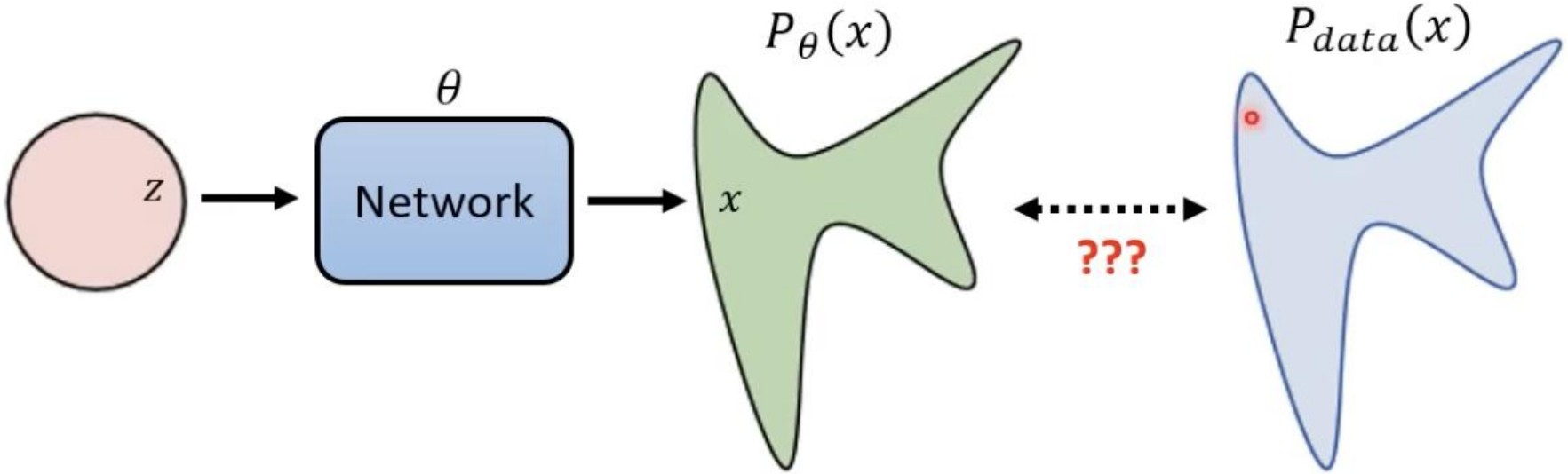
前向过程 $q(x_t|x_{t-1}) = N(x_t; \sqrt{a_t}x_{t-1}, (1-a_t)I)$



反向过程 $p(x_{t-1}|x_t) = N(x_{t-1}; \frac{1}{\sqrt{a_t}}(x_t - \frac{1-a_t}{\sqrt{1-a_t}}\bar{z}(x_t, t)), \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t I)$

Reverse Diffusion Process

Maximum Likelihood Estimation



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

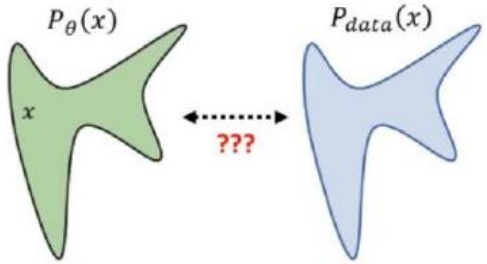
We can compute $P_\theta(x^i)$

???

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

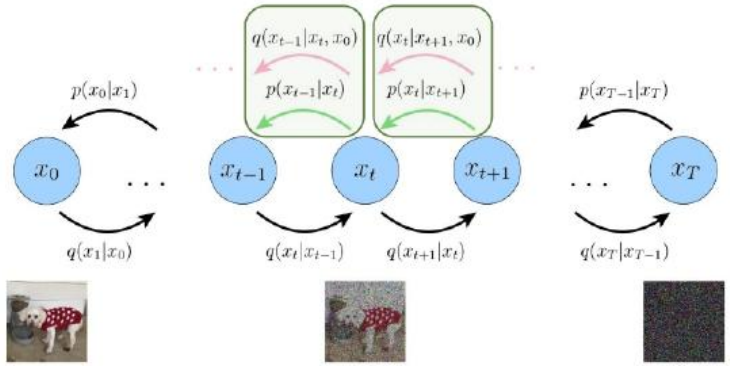
p -> approximated distribution

Maximum Likelihood Estimation



$$\arg\max_\theta \prod_{i=1}^{t} p_\theta(\boldsymbol{x}_i)$$

$$\arg\max_\theta \sum_{i=1}^{t} \log p_\theta(\boldsymbol{x}_i)$$



②Optimization (view 1)

$$\min \; -\log p_\theta(x_0) \le -\log p_\theta(x_0) + D_{KL}\left(q(x_{1:T} \mid x_0) \,\|\, p_\theta(x_{1:T} \mid x_0)\right)$$

加上非负项放缩

见证明1

$$\min \; -\log p_\theta(x_0) \le \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T} \mid x_0)}{p_\theta(x_{0:T})}\right] \quad \textbf{(ELBO)}$$

改变优化目标，类似于VAE

见证明2

$$\min \; -\log p_\theta(x_0) \le \mathbb{E}_{q(x_{1:T}|x_0)}\left[\underbrace{D_{KL}\left(q(x_T \mid x_0) \,\|\, p_\theta(x_T)\right)}_{\text{prior term}}\right] + \sum_{t=2}^{T} \underbrace{D_{KL}\left(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t)\right)}_{\text{reconstruction term}} - \log p_\theta(x_0 \mid x_1)$$

常数，可以忽略

没有可学习参数

证明1

$$-\log p_\theta(x_0) \le -\log p_\theta(x_0) + D_{KL}(q(x_{1:T}|x_0)||p_\theta(x_{1:T}|x_0)) \quad \text{KL 散度非负，进行放缩}$$

$$= -\log p_\theta(x_0) + \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})/p_\theta(x_0)}\right]; \quad \text{where} \quad p_\theta(x_{1:T}|x_0) = \frac{p_\theta(x_{0:T})}{p_\theta(x_0)}$$

$$= -\log p_\theta(x_0) + \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} + \underbrace{\log p_\theta(x_0)}_{\text{与}q\text{无关}}\right]$$

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right].$$

左右取期望，利用重积分中的Fubini定理

$$\mathcal{L}_{VLB} = \underbrace{\mathbb{E}_{q(x_0)}\left(\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right]\right) = \mathbb{E}_{q(x_{0:T})}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right]}_{Fubini\text{定理}} \ge \mathbb{E}_{q(x_0)}[-\log p_\theta(x_0)]$$

Jensen不等式也可以推导得出一样的结论

证明2

$$
\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] \quad \text{对数性质，乘变加，提取一项} \\
&= \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] \\
&\qquad \text{扩散终态的损失} \qquad \text{各时间步的KL散度之和} \qquad \text{初始状态的损失} \\
&= \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] \quad \text{重新整理log项} \\
&= \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}
$$

q: 前向过程
p: 反向过程
pθ：预测

贝叶斯定理 $q(x_{t-1}|x_t,x_0) = \dfrac{q(x_t|x_{t-1},x_0)\cdot q(x_{t-1}|x_0)}{q(x_t|x_0)}$

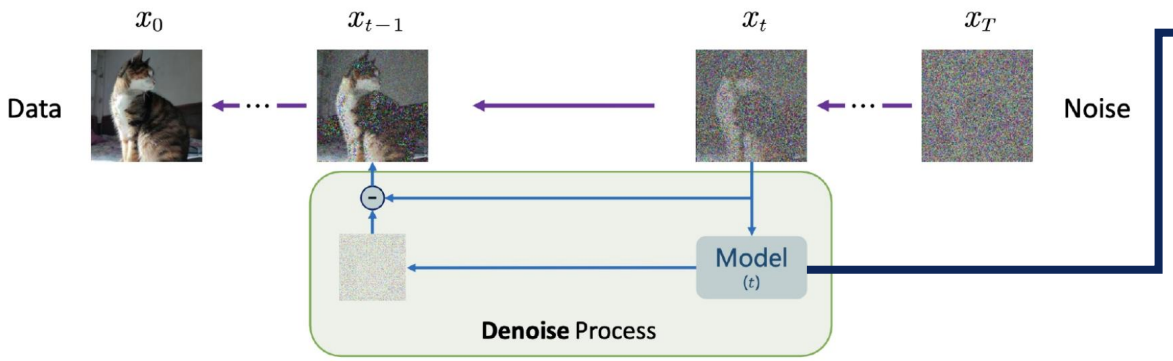
Markov性 $q(x_t|x_{t-1},x_0) = q(x_t|x_{t-1})$

利用KL散度，我们可以精确地计算出当我们近似一个分布与另一个分布时损失了多少信息

优化目标——各时间步的KL散度

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

q没有学习参数，Xt是高斯噪声，所以是常数



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

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

x_t已知，把ε当作参数
训练的Loss 就是惩罚这两个分布之间的差异
(两个都是正态分布，KL散度可以直接套公式)

$$
\begin{aligned}
L_t &= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t)\|_2^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t,t)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right) - \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\right)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t)\|^2\right]
\end{aligned}
$$

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

$$KL(p,q) = \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma^2 + (\mu_1-\mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

Brief Introduction

前向加噪

重参数技巧
任意时刻xt可以由x0表出

反向去噪

马尔可夫性质
三个高斯进行运算

计算均值以及方差

模型训练

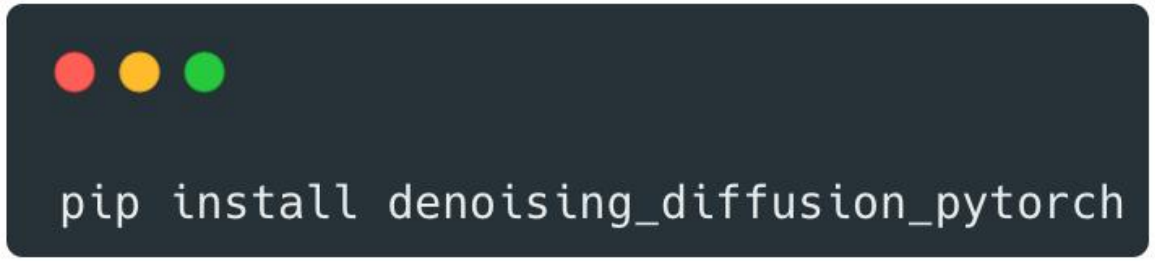
使用MLE进行估计
使用KL放缩
Loss是两个分布KL散度

Diffusion Model with Pytorch

自己搭一个简易的框架

有趣的开源项目

denoising-diffusion-pytorch



minDiffusion

build-with-bombs

Applications

TXT2IMG

LDM&DALLE

LLM

LLaDA

Robotics

Diffusion Policy

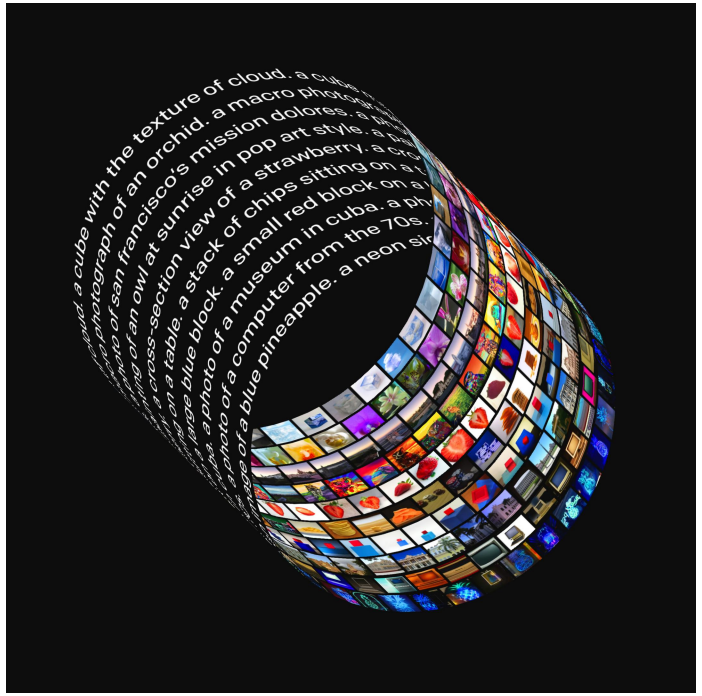
Application 01 —— 图像生成 txt2img

什么是txt2img?

通过给定文本提示词（text prompt）
输出一张匹配提示词的图片。

Stable Diffusio(LDM)

High-Resolution Image Synthesis with Latent Diffusion Models

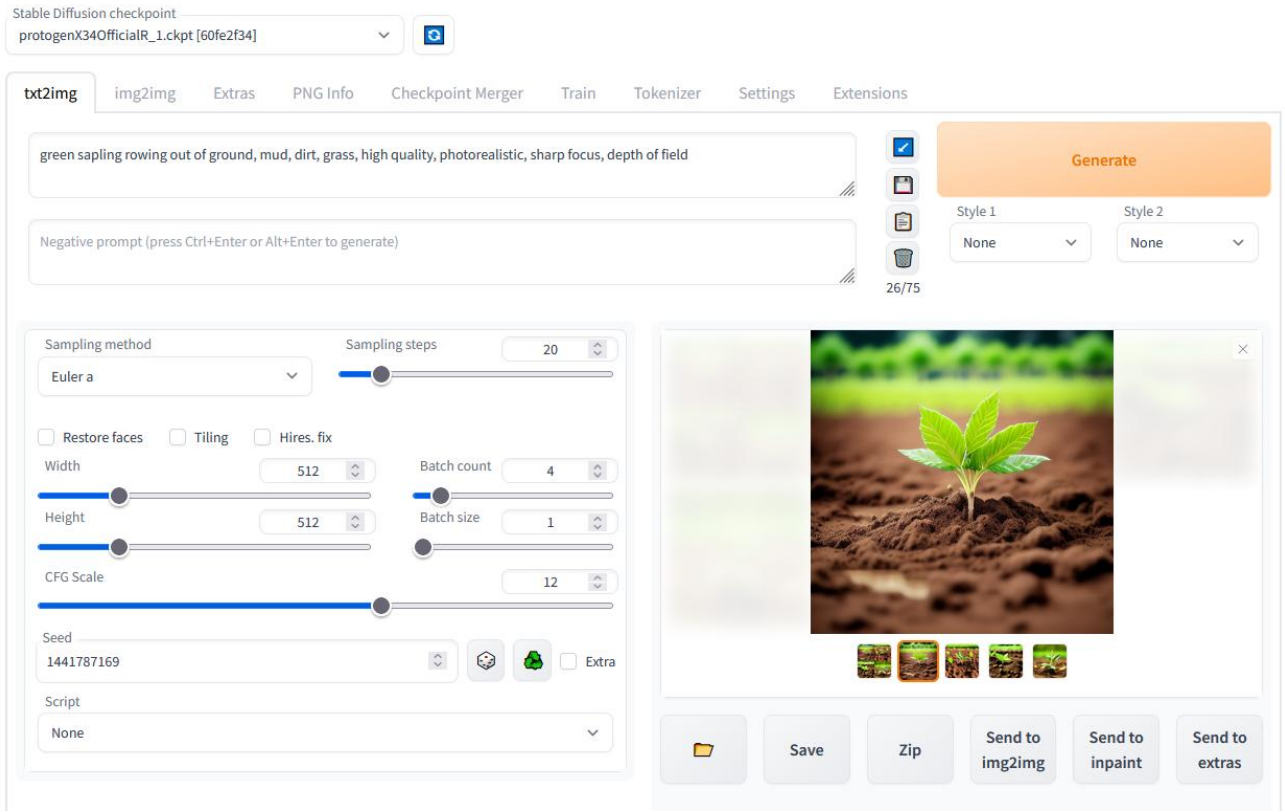


DALLE 2 By OpenAI

Hierarchical Text-Conditional Image Generation with CLIP Latents

Application 01 —— 图像生成 LDM

Have a try! Open Source

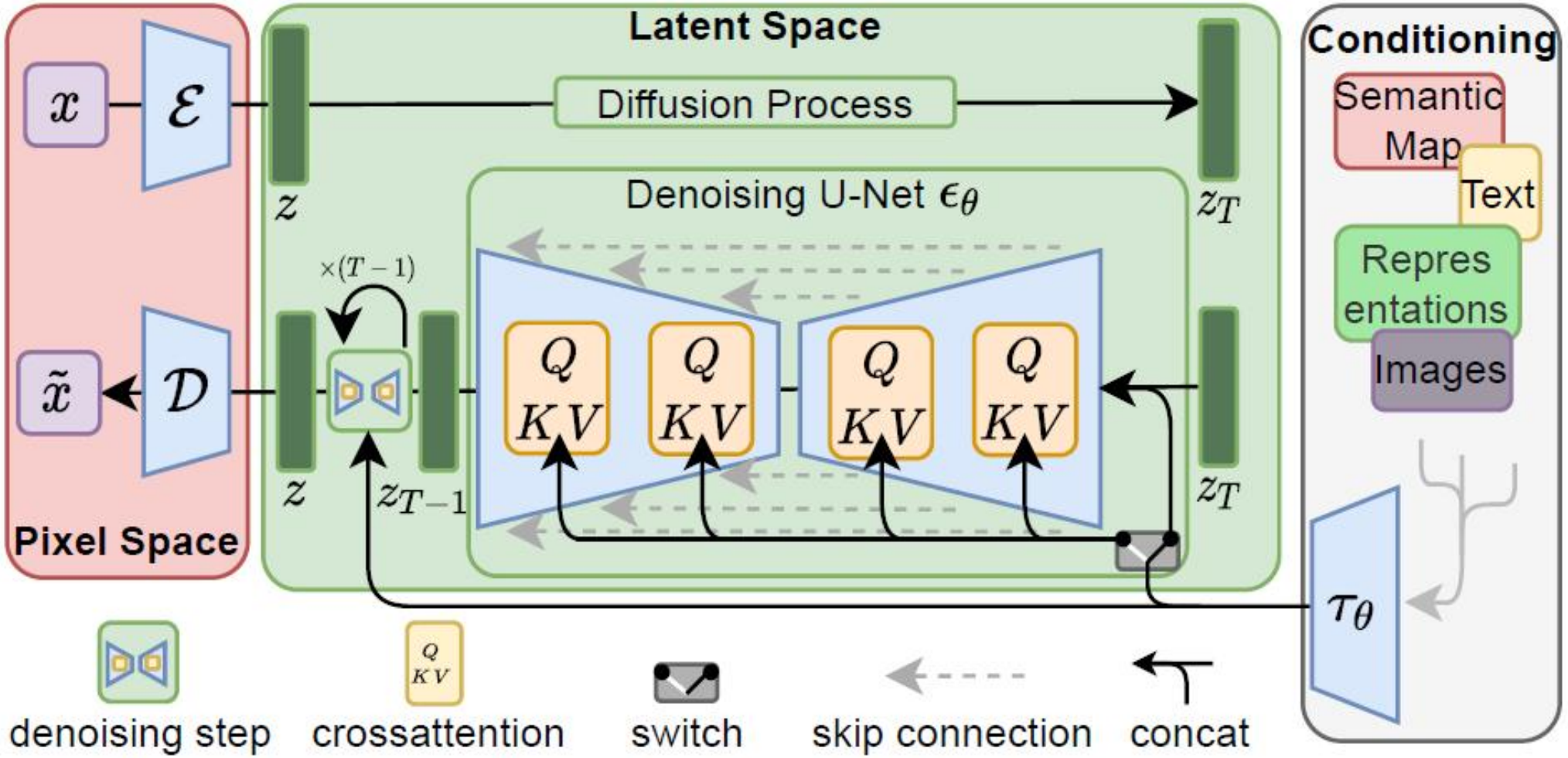


本地部署

https://github.com/AUTOMATIC1111/stable-diffusion-webui

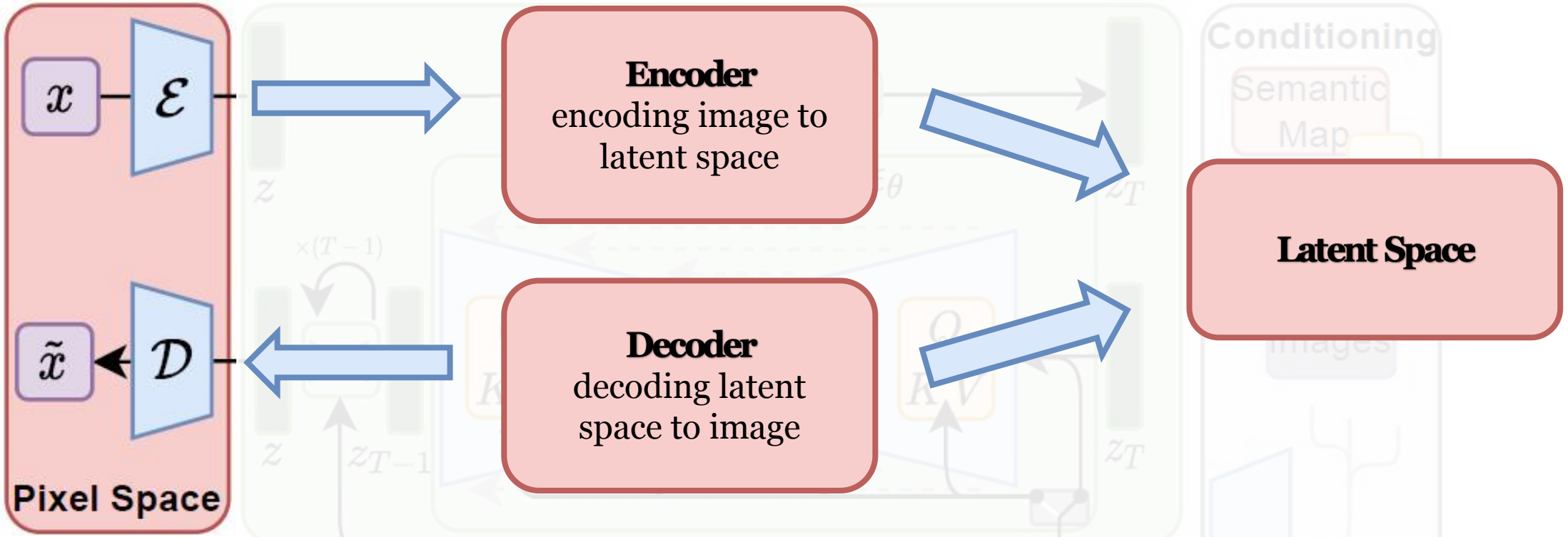
网站访问

Application 01 —— 图像生成 LDM



High-Resolution Image Synthesis with Latent Diffusion Models

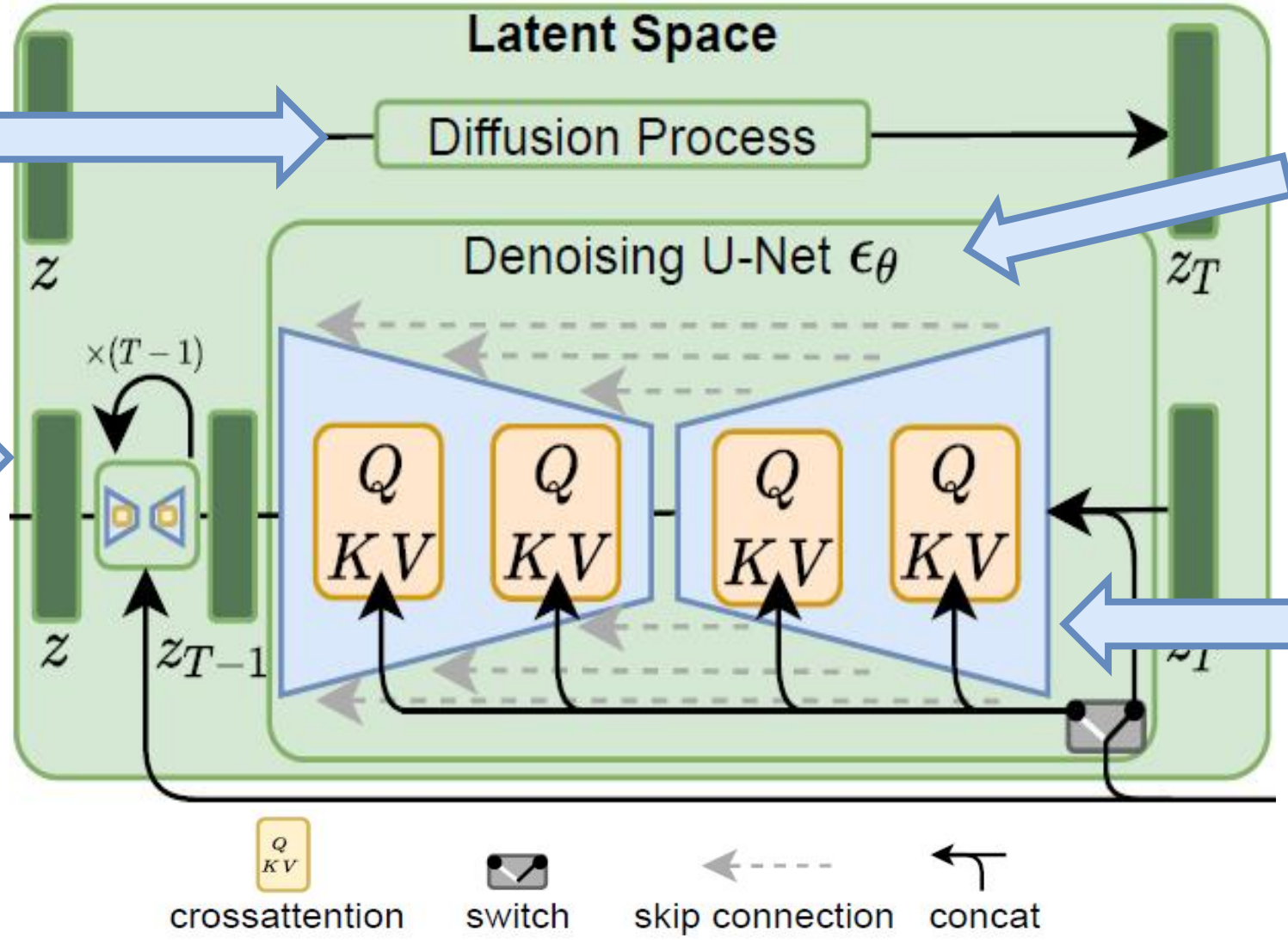
Application 01 —— 图像生成 LDM



Application 01 —— 图像生成 LDM

Diffusion Process
gradual denoising of data

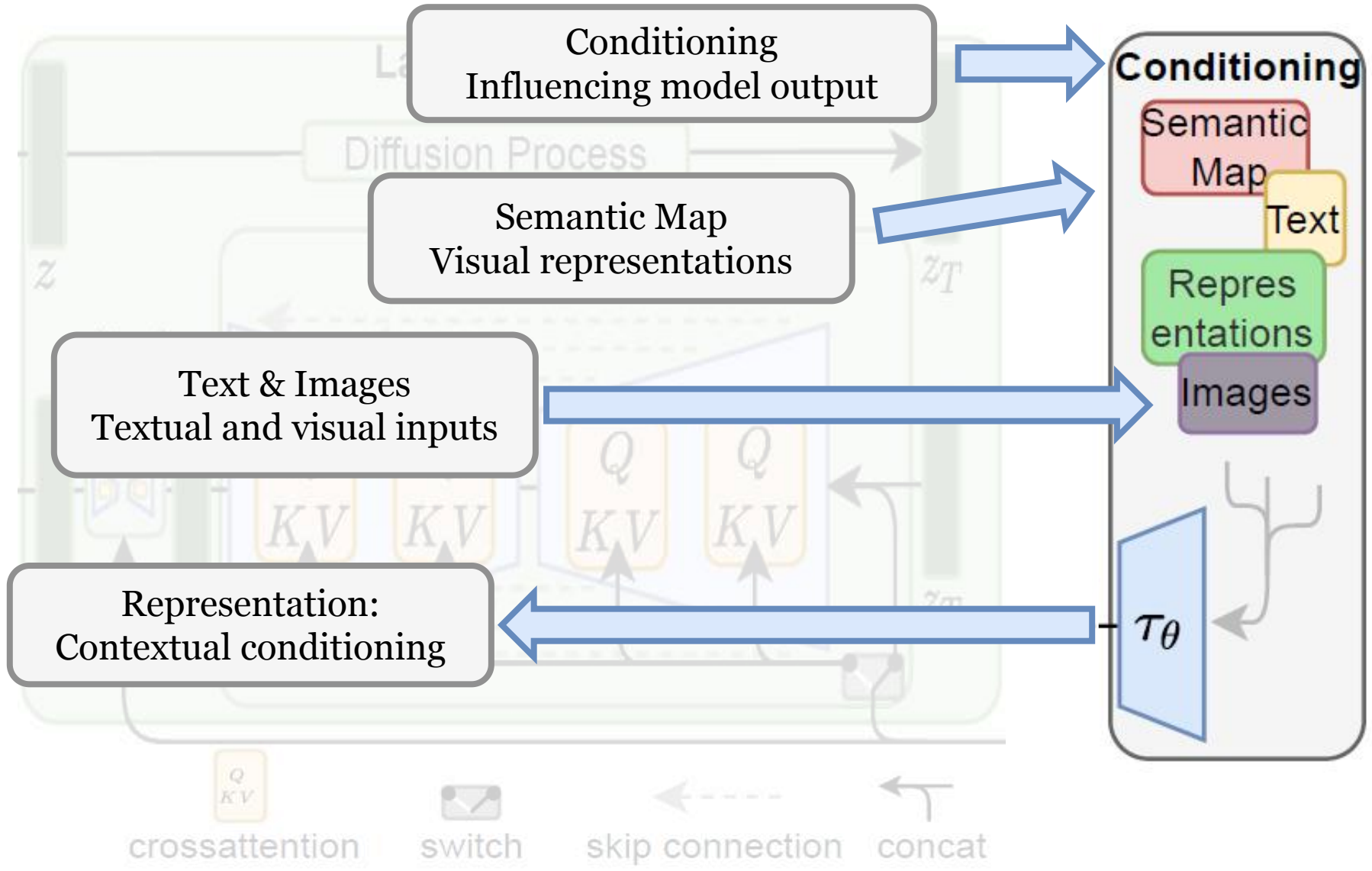
Latent Space
Representation space for diffusion



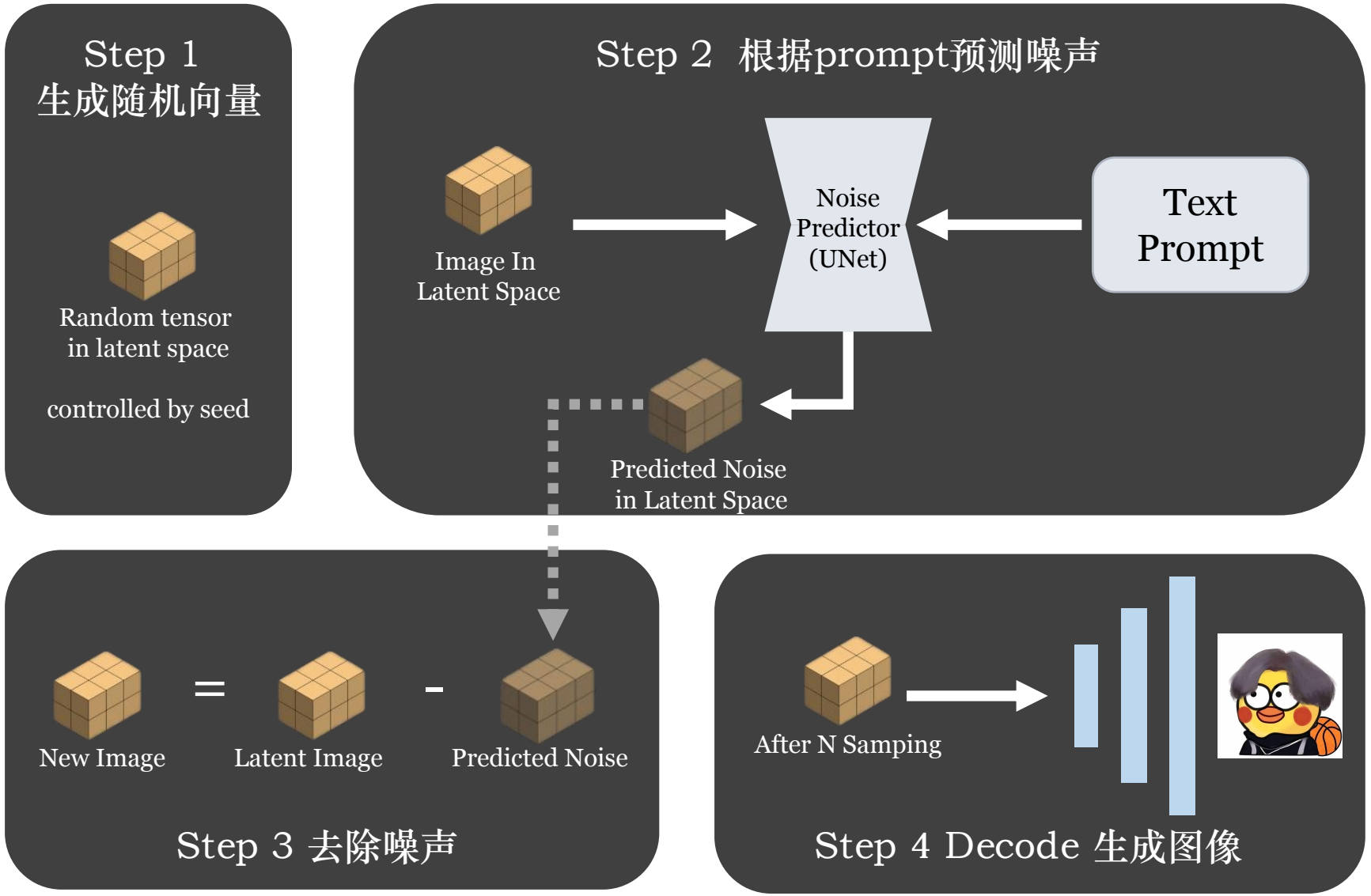
Denoising U-Net:
Denoising architecture

Attention Mechanism
Cross-attention with QKV

Application 01 —— 图像生成 LDM



Application 01 —— LDM 原理



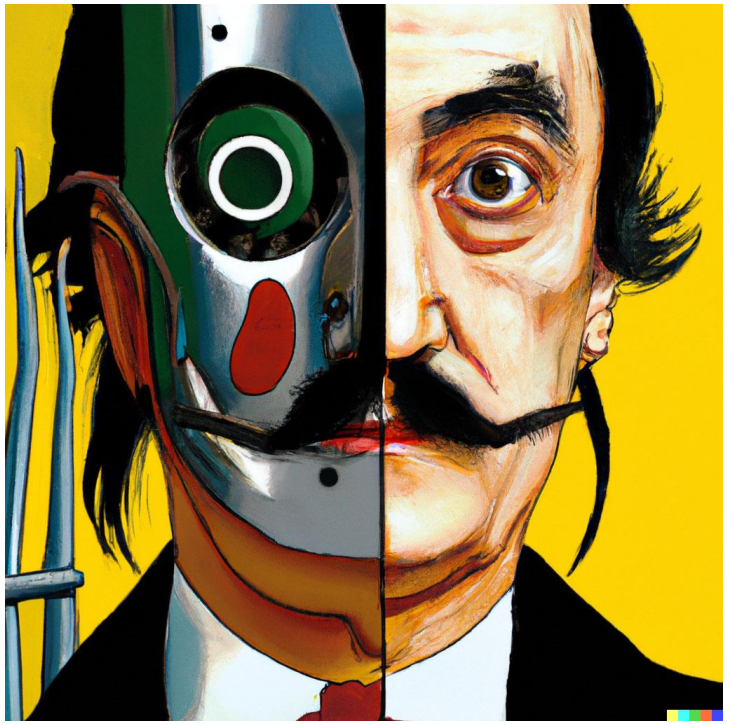
Application 01 —— 图像生成 DALLE2

Have a try! API supported

https://platform.openai.com/docs/overview

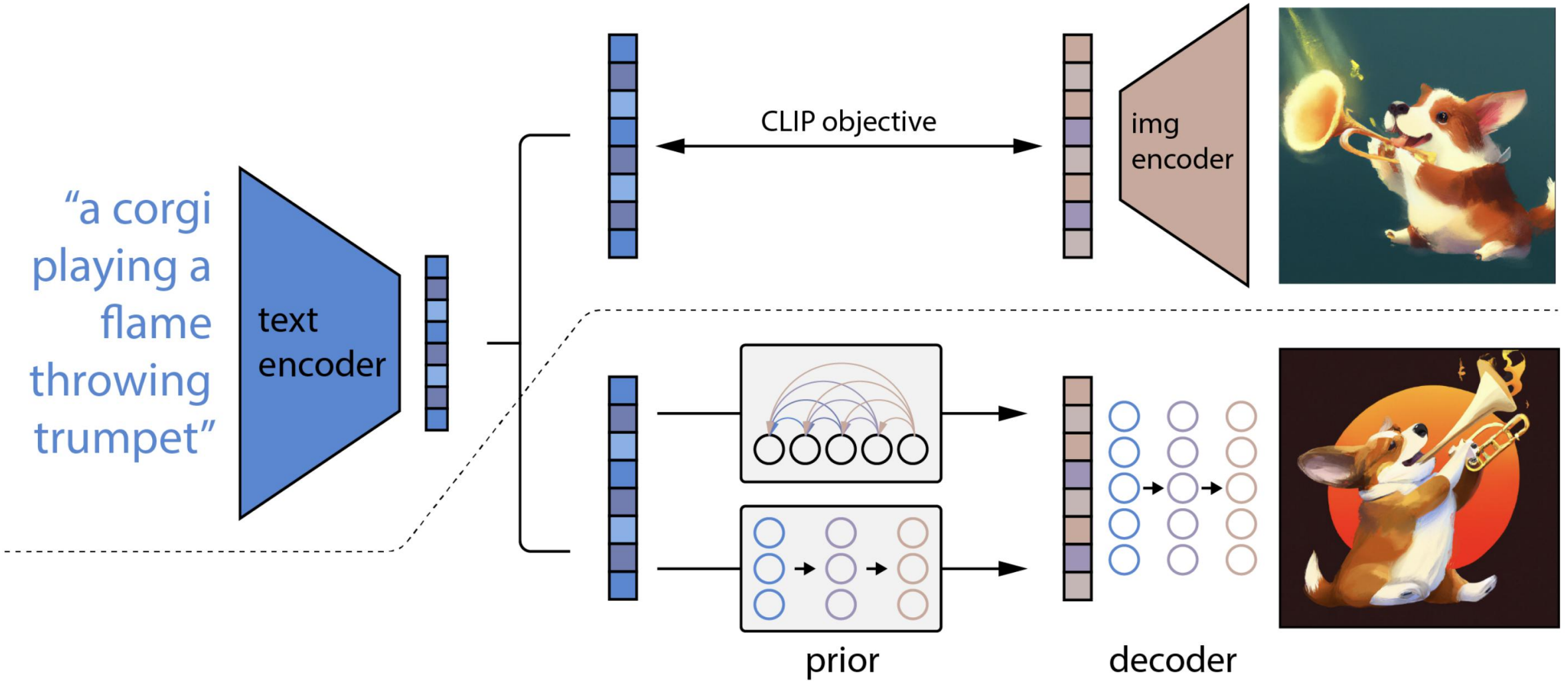
A photo of a white fur monster standing in a purple room

panda mad scientist mixing sparkling chemicals, artstation



vibrant portrait painting of Salvador Dalí with a robotic half face

Application 01 —— 图像生成 DALLE2



Hierarchical Text-Conditional Image Generation with CLIP Latents

Application 02 —— LLaDA

扩散模型也能玩转大语言模型？

Large Language Diffusion Models

What is LLaDA？ **L**arge **La**nguage **D**iffusion with m**A**sking

A text generation method different from the traditional **left-to-right** approach

Application 02 —— LLaDA

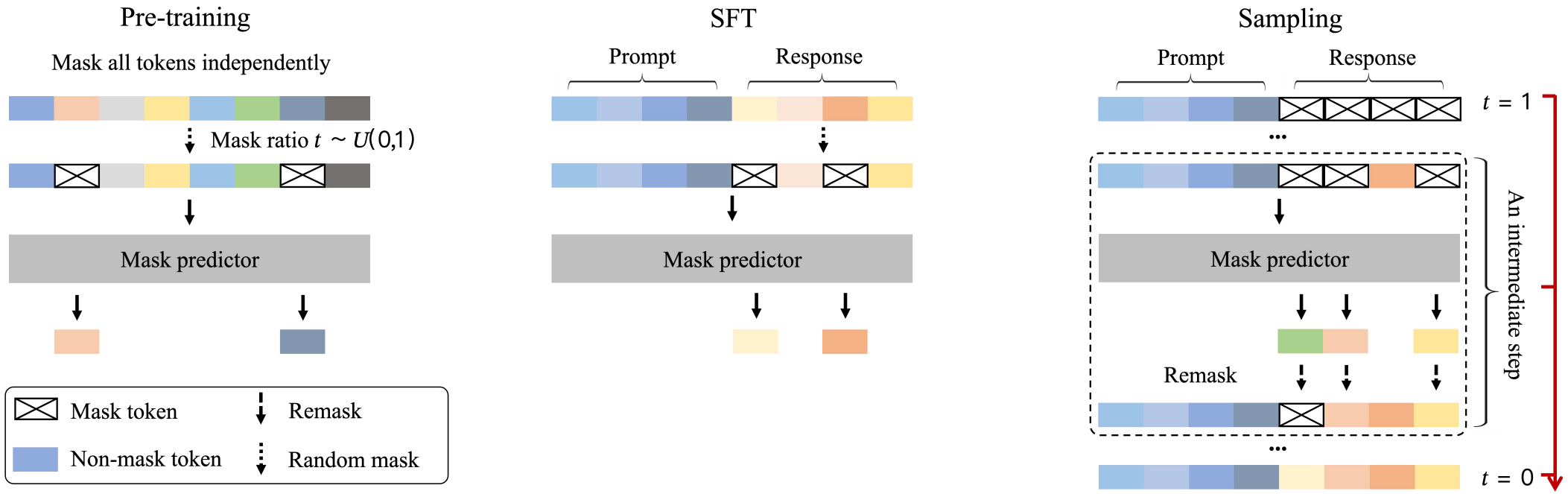
传统路径：Auto Regression

每次生成一个token，新生成的token会拼到序列末尾
每个token的生成依赖于之前所有已生成的tokens

Talk is Cheap， Show me the Code!

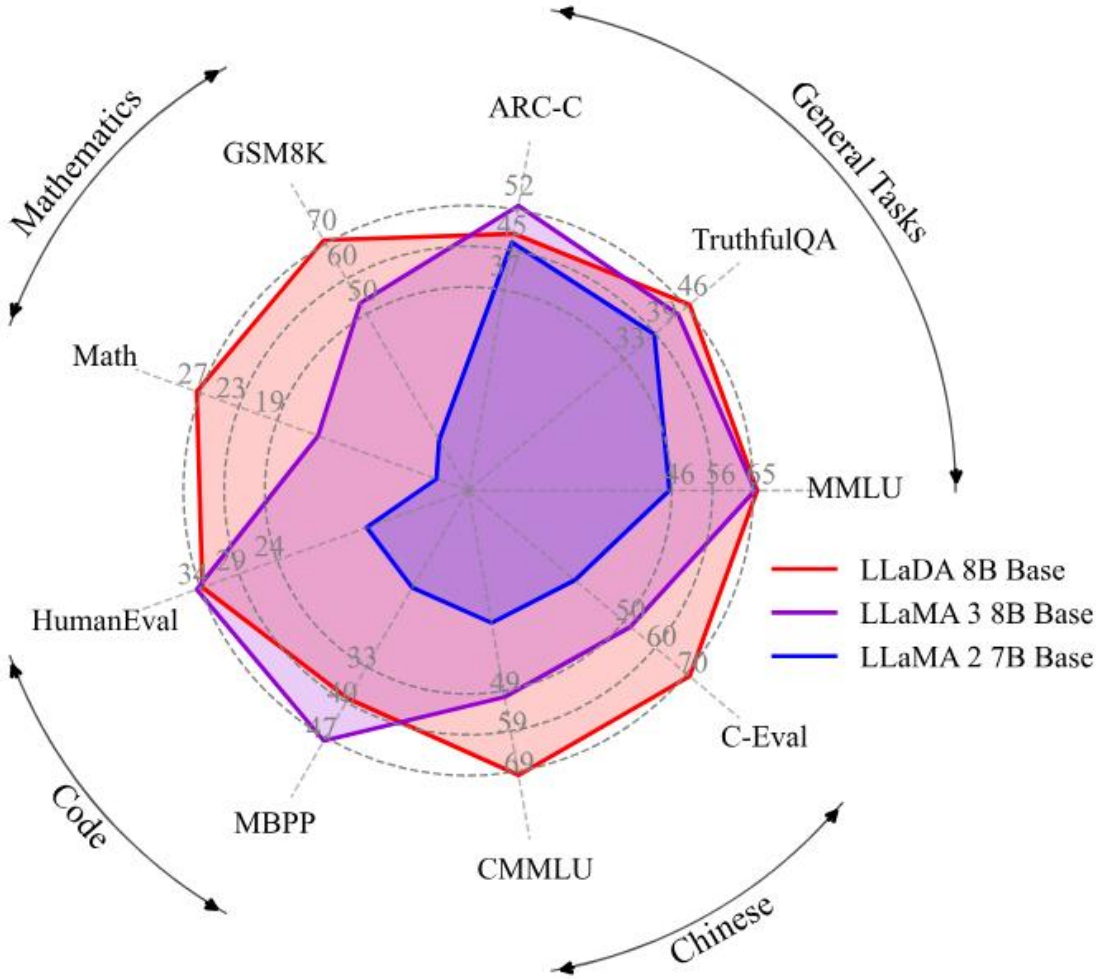
LLaDA

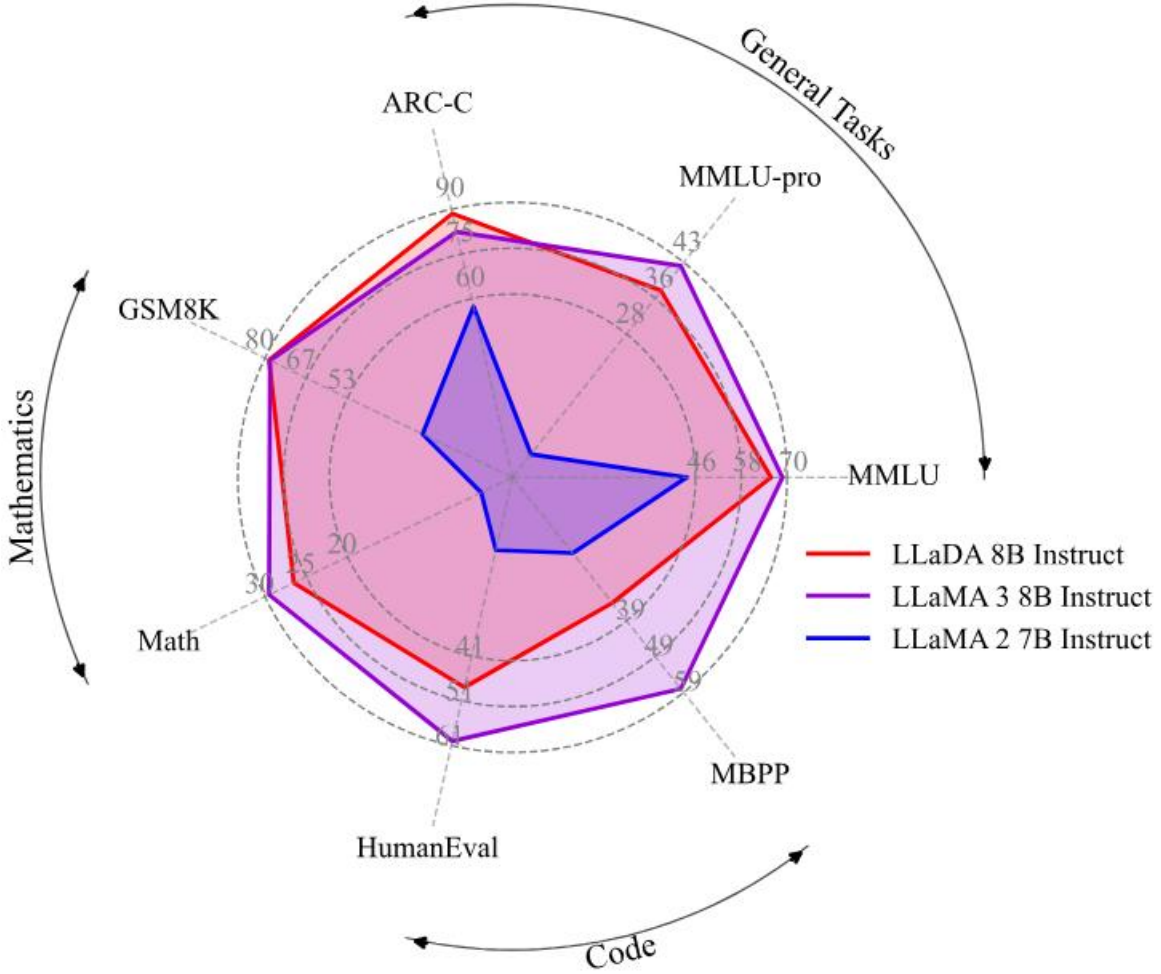
Mask的过程体现了diffusion加噪去噪的思想



Application 02 —— LLaDA

Performance





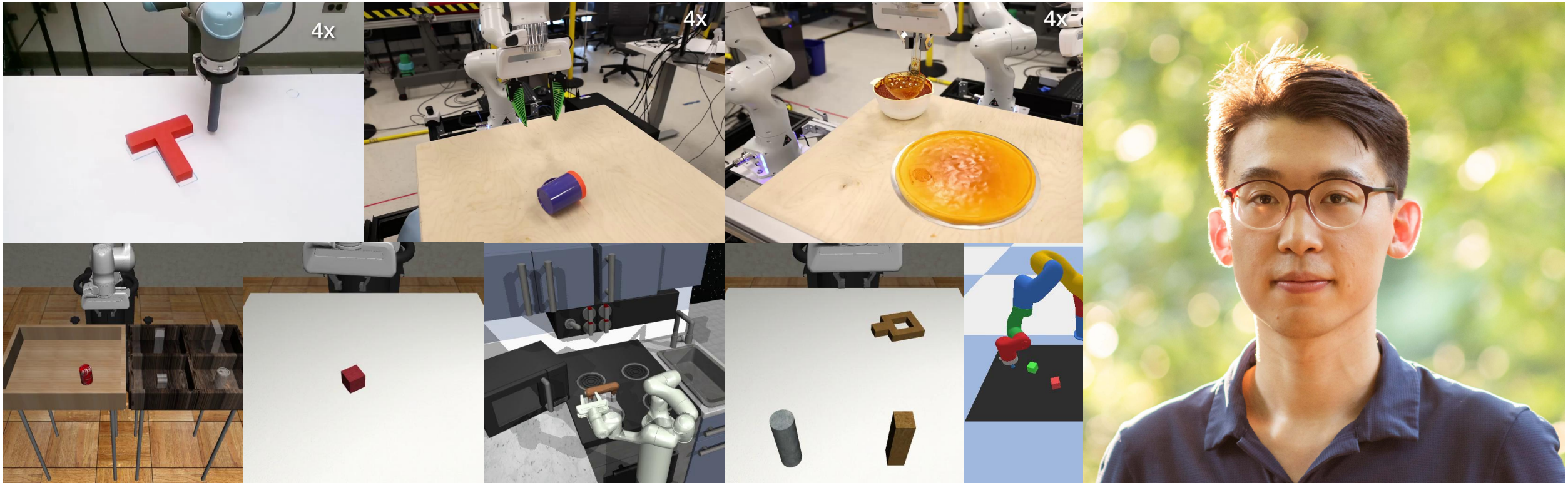
Application 03 —— Robotics

Diffusion Policy

COLUMBIA ENGINEERING

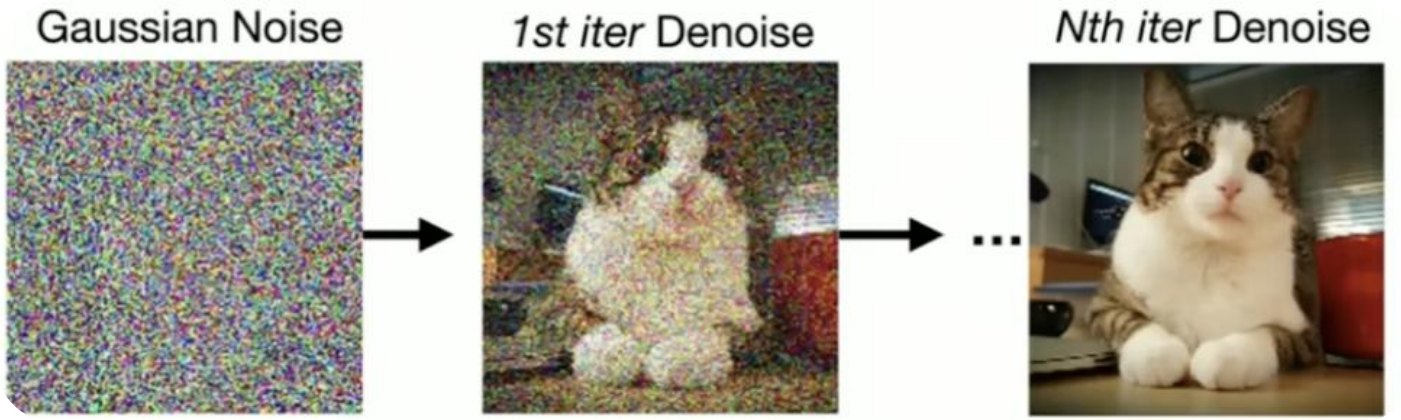
TOYOTA RESEARCH INSTITUTE

Massachusetts Institute of Technology

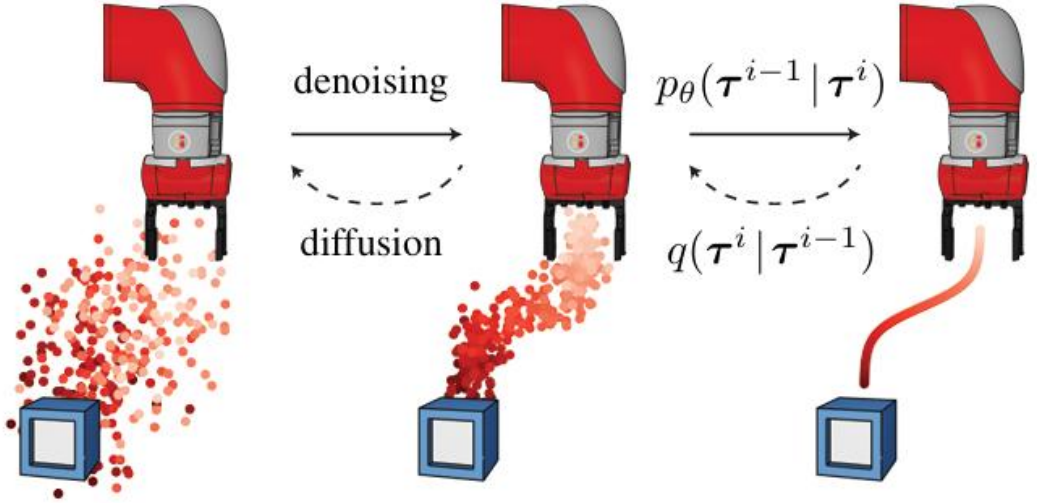


Application 03 —— Robotics

Image Diffusion



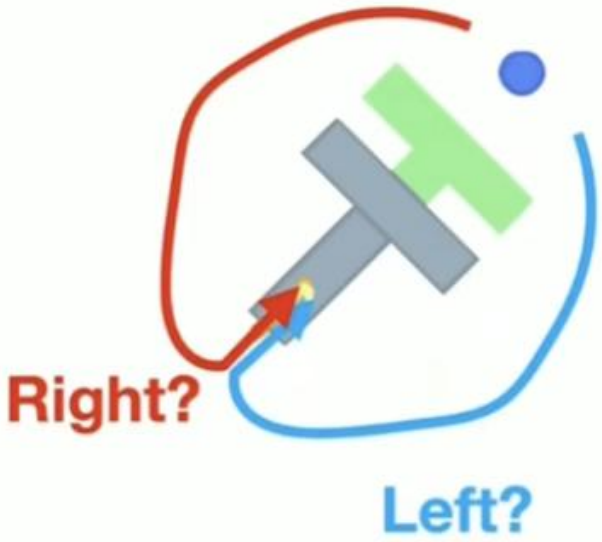
Action Diffusion

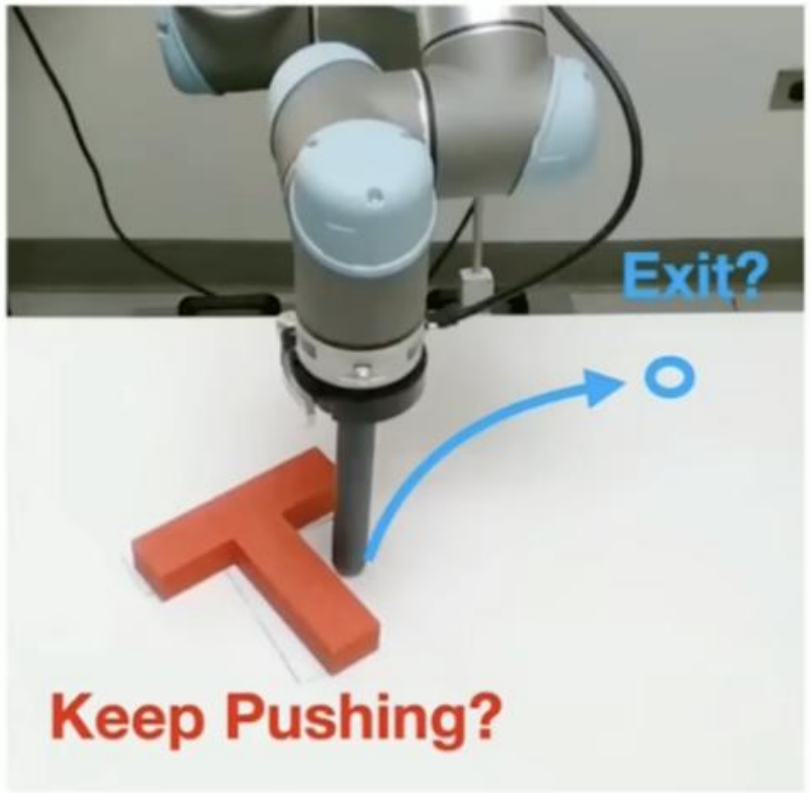


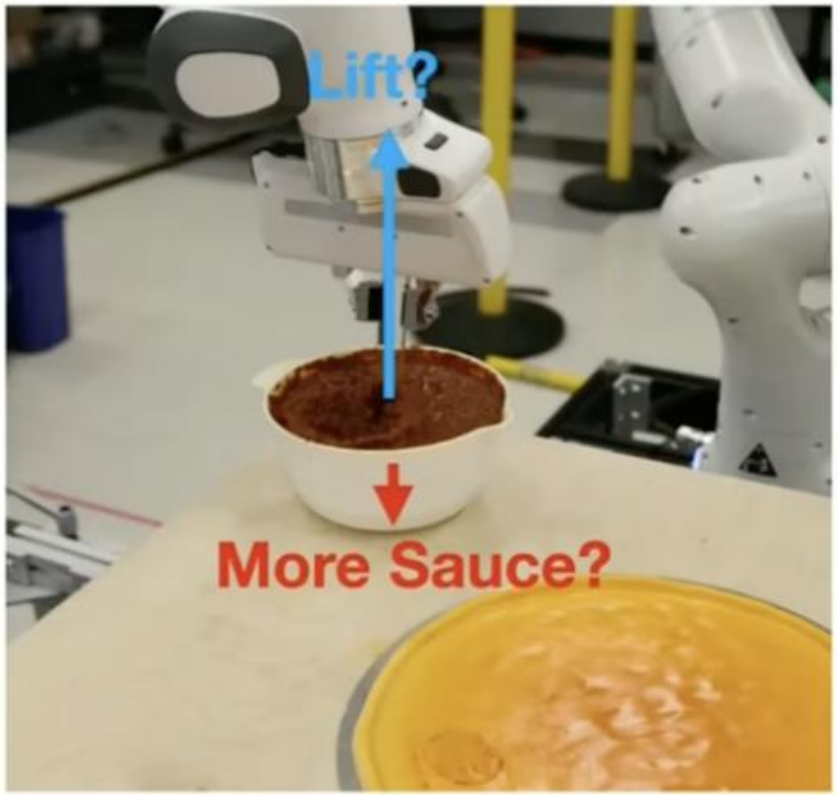
Application 03 —— Diffusion Policy

Action Multimodality

Multiple valid actions for the same observation

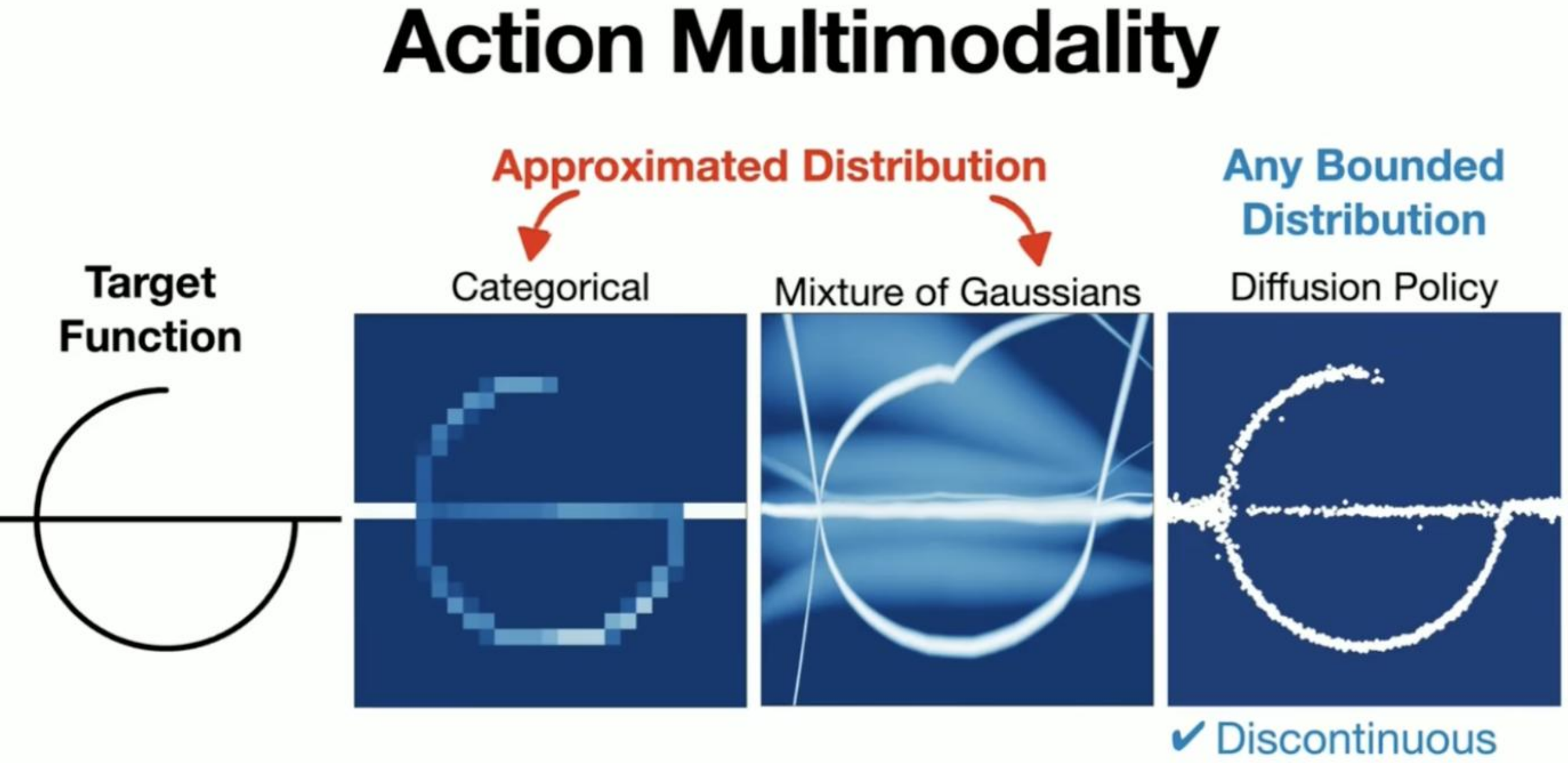






Surprisingly Common

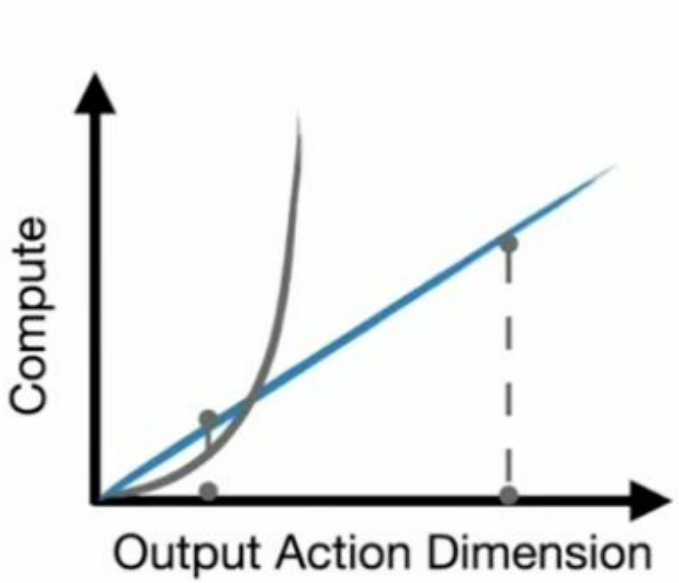
Application 03 —— Diffusion Policy

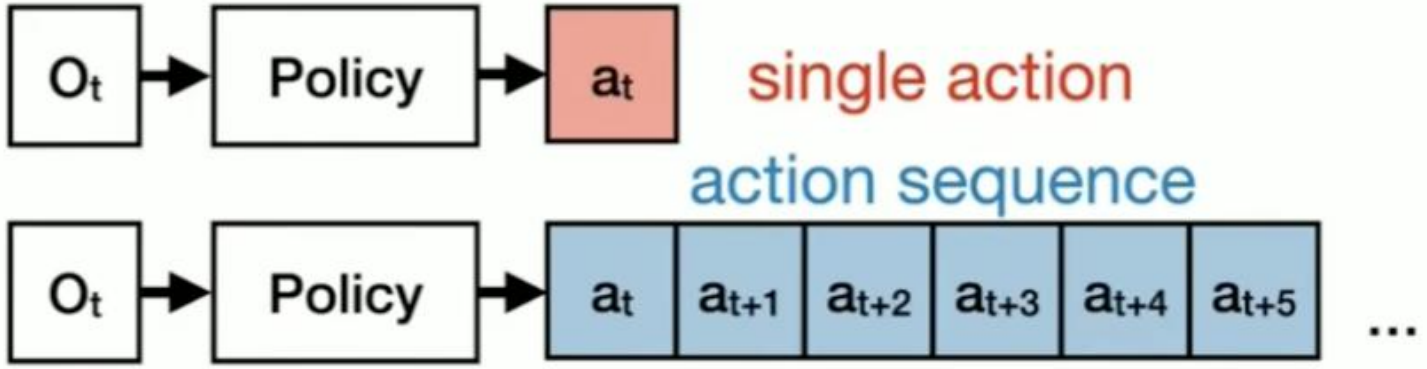


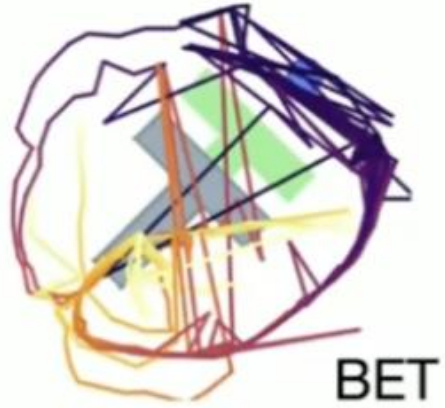
Application 03 —— Diffusion Policy

Action Space Scalability

Easily afford action sequence prediction







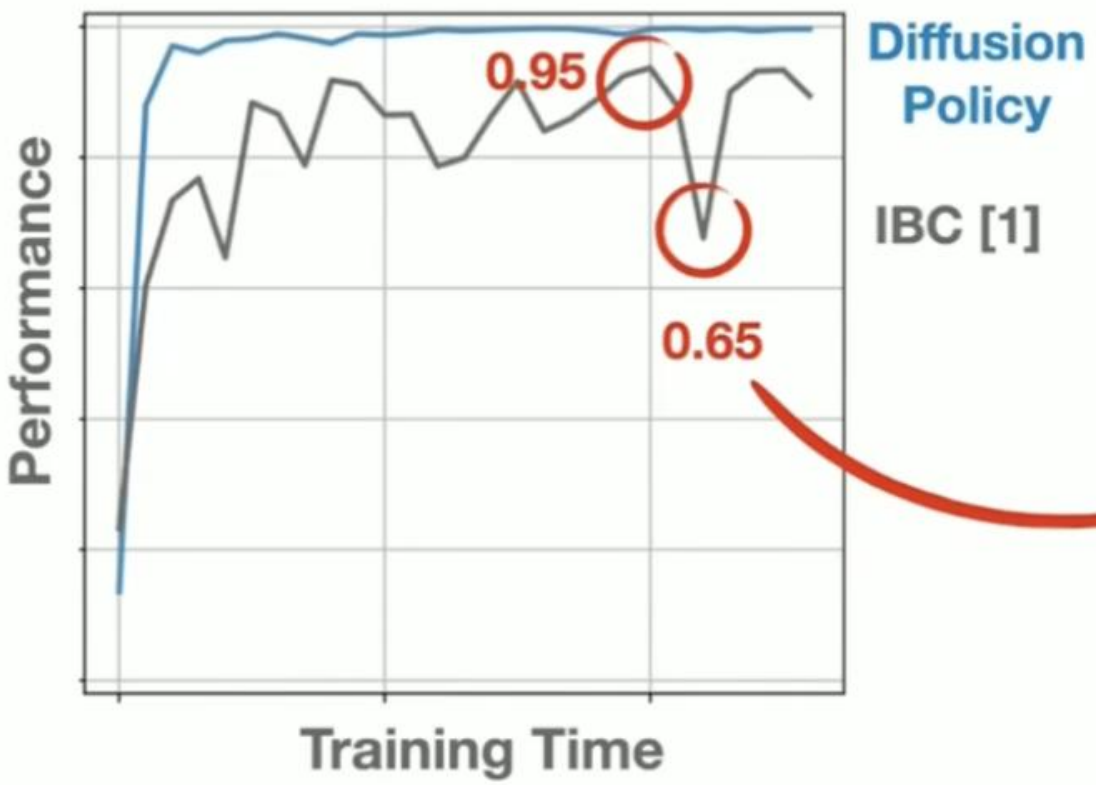
BET [2]

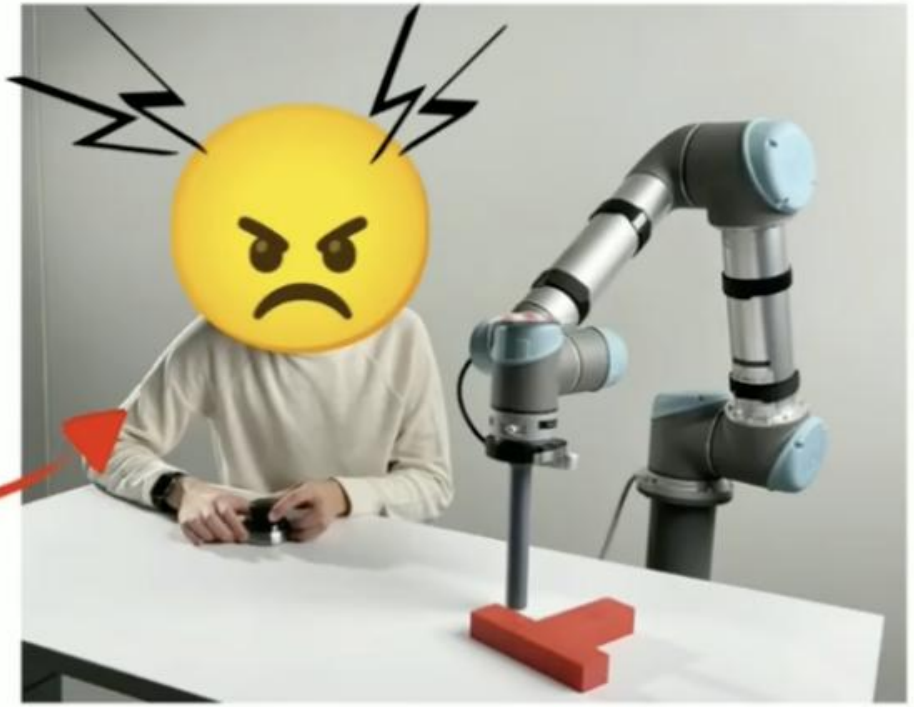
DiffusionPolicy

Application 03 —— Diffusion Policy

Training Stability

Compared to Energy Based Models





Checkpoint selection requires expensive realworld evaluation

Application 03 —— Diffusion Policy

Approach

Step1 训练阶段：学习“去噪”过程　经过多次迭代去噪，模型生成出符合当前观察条件的“干净”动作x0

$$x_{k-1} = \alpha(x_k - \gamma\epsilon_\theta(x_k, k) + N(0, \sigma^2 I))$$

去噪步长的缩放系数　噪声预测函数

$$L = MSE(\epsilon_k, \epsilon_\theta(x_0 + \epsilon_k, k))$$

噪声预测函数

Step2 得分匹配来优化动作的能量

$$\nabla_a \log p(a|o) = -\nabla_a E_\theta(a, o)$$

沿着降低能量的方向调整动作

Step3 InfoNCE损失：区分“好”动作与“坏”动作

$$L_{infoNCE} = -\log\left(\frac{e^{-E_\theta(o,a)}}{e^{-E_\theta(o,a)} + \sum_{j=1}^{N_{neg}} e^{-E_\theta(o,\tilde{a}_j)}}\right)$$

“好”动作

“坏”动作

Step4 推理阶段：实时生成动作序列

Reference

If you are interested in this topic, check these papers uploaded in DingTalk Group

Thanks