

扩散模型 DDPM.

一、《一个视频看懂DDPM原理推导》——Nik-Li

人降噪过程每一步学的是一个分布(而非确切的 X_{t-1}) \Rightarrow 更好随机性

但 unet 的输出最后是个噪音 \leftarrow 重参数化 噪音

零：理论基础。

1. 正态分布(高斯分布), $x \sim N(\mu, \sigma^2)$

(1) 扩散过程的终点 x_T 是一个标准正态分布。

(2) 多维高斯分布

$$x \sim N(\mu, \Sigma)$$

· x 是图片展平后的向量

· μ : 均值向量

· Σ : 协方差矩阵. $\Rightarrow \sigma^2 I$ (每个像素噪声独立)

(3) DDPM用到的“三大高斯性质”:

① 重参数化 (Reparameterization)

· 前向加噪核心: 让随机采样可导。

· 从 $N(\mu, \sigma^2)$ 中采样一个 x :

1° 从 $N(0, 1)$ 中采样一个随机噪声 ϵ .

2° 线性变换得 x : $x = \mu + \sigma \epsilon$

· DDPM中:

$$x_t \sim N(\sqrt{1-\beta_t} x_{t-1}, \beta_t I)$$

$$x_t = \sqrt{1-\beta_t} x_{t-1} + \sqrt{\beta_t} \cdot \epsilon \quad \begin{matrix} \leftarrow noise \\ \text{转换成预测} \end{matrix}$$

噪声 ϵ .

② 高斯分布的可加性.

VAE 变分自编码器

(VAE, DDPM)

在生成模型论文中的符号:

- Q: 人为设定的、已知的真实分布
· 破坏者 · 完全已知 $q(x_t | x_{t-1})$: 在前一时刻的基础上加噪
· Encoder

- P: 神经网络试图学习的、模拟的分布.

- 重建者 · 可学习的, p_θ (θ 为权重)
 $p_\theta(x_{t-1} | x_t)$: 神经网络猜上一时刻的分布.
· Decoder.

• 两个独立高斯分布的随机变量相加，结果依然高斯。

$$X \sim N(\mu_1, \sigma_1^2), Y \sim N(\mu_2, \sigma_2^2) \Rightarrow X+Y \sim N(\mu_1+\mu_2, \sigma_1^2+\sigma_2^2)$$

• 在DDPM中：

让扩散中的加噪一步到位 $\Rightarrow q(x_t|x_0)$ 的分布。

③ 贝叶斯公式与高斯乘积

$$N(x; \mu_1, \sigma_1^2) \cdot N(x; \mu_2, \sigma_2^2) \propto N(x; \mu_{\text{new}}, \sigma_{\text{new}}^2)$$

• 在DDPM中： $q(x_t|x_{t-1}) \cdot q(x_{t-1}|x_0)$ ，具体情况见下节。

2. 贝叶斯公式

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

(1) 先验概率 (Prior, $P(A)$)

• 根据以往的经验 $\rightarrow q(x_{t-1}|x_0)$

(2) 似然 (Likelihood, $P(B|A)$)

• 假设 A 已发生的情况下，出现 B 的概率 $\rightarrow q(x_t|x_{t-1})$

(3) 后验 (Posterior, $P(A|B)$) (从果推因)

• 最想求的值。看到证据 B 后，需要修正原来的判断，得到事件 A 发生的概率。

逆向去噪过程：已知：全是噪声的 x_t (证据 B)

想求：这图上一时刻 x_{t-1} (A) 的概率

$$q(x_{t-1}|x_t)$$

(4) 贝叶斯在DDPM。

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

后验概率公式

• 假设训练集已知原因 x_0 :

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

• 反侧：三项全部已知且高斯

$q(x_{t-1}|x_t, x_0)$ 的均值与方差可以单独解析

4. 联合分布 (Joint Distribution)

(1) 描述了多个随机变量同时取特定值的概率。

(1) 离散 (Discrete)

$$P_{XY}(x,y) \triangleq P(X=x, Y=y)$$

性质：(1) 归一性

$$\sum_{x \in X} \sum_{y \in Y} P(x=x, y=y) = 1$$

- X, Y : 随机变量 (不确定的函数/状态) $P(x \leq X \leq x+dx, y \leq Y \leq y+dy) \approx f_{XY}(x,y) dx dy$
- x, y : 该变量取到的具体数值.

(2) 连续 (Continuous)

使用联合概率密度函数 (PDF):

$$f_{XY}(x,y) \rightarrow p(x,y)$$

对连续度量，单点概率 $P(x=x, y=y) = 0$
因此我们讨论的是落在一个小区域内的概率。

归一性:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{XY}(x,y) dx dy = 1.$$

5. 边缘分布 (Marginal Distribution)

(1) 对联合分布中“无关变量”进行求和(离散)/求积分(连续)。

(2) 离散

$$P_X(x) = \sum_{y \in Y} P_{XY}(x,y)$$

(3) 连续

$$f_X(x) = \int_{-\infty}^{\infty} f_{XY}(x,y) dy$$

↓DDPM

$$p(x) = \int p(x,y) dy.$$

6. 重要法则

(1) 加法法则 (边缘化)

→ 谓变量

$$P(x) = \int p(x,y) dy.$$

7. 期望 (expectation view)

· 把积分写成期望

$$P(x) = E_{Z \sim p(z)} [p(x|z)]$$

· $p(z)$: z 服从的分布

· DDPM: 因象 x 的分布,本质上是无数条可能噪声路径的生成概率的平均值。

边缘概率 权重: z 出现的先验概率

(2) 乘法

$$P(x,y) = P(y|x) \cdot P(x) = P(x|y) \cdot P(y)$$

(3) 全概率公式 → 描述隐变量模型的生成过程.

$$P(x) = \int p(x|z) \cdot p(z) dz$$

物理意义: x 发生的概率 = 在所有可能的隐状态 z 下生成 x 的概率的加权平均.

→ 将神经网络的任务 → 学习 $q(x_{t+1} | x_t, x_0)$ 的均值 → 预测噪声 ϵ .

3. 最大化似然 (Log likelihood) + 变分推断 (Variational Inference)

(1) 定义: 真实数据 $X = \{x_1, x_2, \dots, x_n\}$, 模型是一个概率分布 $P_\theta(x)$. 似然函数就是模型生成这些真实数据的联合概率: $L(\theta) = P_\theta(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P_\theta(x_i)$
目标是找一个 θ , 让 $L(\theta)$ 最大.

(2) Why Log?

- 连乘 → 连加: 防止 Underflow
- 方便求导.

$$\Rightarrow \theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^n \log P_\theta(x_i)$$

(3) 最小化 Loss \Leftrightarrow 最小化负的 Likelihood.

$$Loss = - \sum_{i=1}^n \log P_\theta(x_i)$$

(4) DDPGM 中的最大似然.

1° 由于预测数据 ~ 高斯分布 $\rightarrow P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

$$\Rightarrow -\log P(x) = \underbrace{\frac{1}{2\sigma^2}(x-\mu)^2}_{MSE} + \underbrace{\log(\sqrt{2\pi\sigma^2})}_{\text{常数项}}$$

∴ 在高斯分布前提下: Max Log-Likelihood \Leftrightarrow MSE.

2° 由于解析的 $P_\theta(x_0)$ 无法计算 $\rightarrow \int \underbrace{P_\theta(x_0, x_1, \dots, x_T)}_{\text{无数条 } x_0 \rightarrow x_1 \text{ 的路径}} dx_{1:T} = P(x_0)$ (边缘概率)

∴ 要取其下界 (Evidence Lower Bound) \mathbb{E}_{BLBO}

$$\therefore ELBO \approx \sum_{\text{后验}} D_{KL}(\text{后验} || \text{模型}) \approx \sum \text{MSE}$$

高斯分布下, $D_{KL} \propto \|\mu_1 - \mu_2\|^2 \rightarrow$ 均值的均方误差.

★ 分布差异 \Leftrightarrow 由噪声决定

4. KL 散度 (KL Divergence)

· 衡量 2 个概率分布之间的差异. $D_{KL}(A || B) = 0 \Leftrightarrow A \text{ 和 } B \text{ 完全一致.}$
· 值越大, 表示差距越大.

推导:

$$X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{1-\alpha_t} \varepsilon_{t-1}$$

$$= \sqrt{\alpha_t} (\underbrace{\sqrt{\alpha_{t-1}} X_{t-2} + \sqrt{1-\alpha_{t-1}} \varepsilon_{t-2}}_{\text{①}}) + \sqrt{1-\alpha_t} \varepsilon_{t-1}$$

$$= \sqrt{\alpha_t \alpha_{t-1}} X_{t-2} + \underbrace{\sqrt{\alpha_t(1-\alpha_{t-1})} \varepsilon_{t-2} + \sqrt{1-\alpha_t} \varepsilon_{t-1}}_{\text{②}}$$

noise



$$\sqrt{\sqrt{\alpha_t^2 + (1-\alpha_t)^2}}$$

$$= \sqrt{1 - \alpha_t \alpha_{t-1}}$$

$$= \sqrt{\alpha_t \alpha_{t-1}} X_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}} \varepsilon$$

$$\Rightarrow \sqrt{\alpha_t} X_0 + \sqrt{1 - \alpha_t} \cdot \varepsilon$$

$$\underbrace{(\ln \alpha_t)^2}_{\text{无偏性}} + \underbrace{(1 - \alpha_t)^2}_{\text{无偏性}} = \text{exp}(\ln \alpha_t)$$

6. 白噪声 \Leftrightarrow 独立同分布 ε 且 $E[\varepsilon] = 0$, $E[\varepsilon^2] = 1$

$$\text{cov}[\varepsilon] = E[\varepsilon \varepsilon^T] = E[\varepsilon \varepsilon^T] - E[\varepsilon] E[\varepsilon^T] = 0$$

(Gaussian noise) 是「某时点上」

$$\text{EWA}_3 = (0.333 \times 0.333 \times 0.333) \times 0.857$$

的「未来」 \rightarrow 未来是不可见的

→ 未来是不可见的

过去是已知的

现在是已知的

一、前向过程

1. 单步加噪:

$$x_t = \sqrt{1-\beta_t} x_{t-1} + \sqrt{\beta_t} \varepsilon_t$$

(重参数化)

- β_t : 方差调度参数 (Variance Schedule), 很小接近于0.

- $\varepsilon_t \sim N(0, I)$ 采样的随机噪声.

2. 任意步加噪 ($x_0 \rightarrow x_t$)

- 记: $\alpha_t = 1 - \beta_t$

$$\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$$

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

i.e. $q(x_t | x_0) \sim N(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I)$

- 物理含义: $\sqrt{\bar{\alpha}_t} x_0$: 信号项. 随着 $t \uparrow$, 原图信息↓

- $\sqrt{1 - \bar{\alpha}_t} \varepsilon$: 噪声项. $t \uparrow$, 权重 $\rightarrow 1$, 图像变为纯噪声.

- 当 x_T 中 I 会很大时, $x_T \approx \varepsilon$ 意味着什么?

Ans. 扩散过程最终将任意数据分布收敛到标准正态分布, 保证了我们可以用同样标准正态分布的噪声来开始生成过程.

二、逆向过程

$$\frac{P(x_t | x_{t-1}) \cdot q(x_{t-1})}{P(x_t)} \text{ 不好求.}$$

- 思路: 直接算 $P(x_{t-1} | x_t)$ 很难, 但可以算 $q(x_{t-1} | x_t, x_0)$ 的均值

i.e. $q(x_{t-1} | x_t, x_0) \sim N(x_{t-1}; \tilde{\mu}_t, \tilde{\beta}_t I)$

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_{t-1}}{1 - \bar{\alpha}_t} x_0 \quad \leftarrow \text{无法推理}$$

$$\Downarrow \text{任意步加噪反解 } x_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \varepsilon}{\sqrt{\bar{\alpha}_t}}$$

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

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

没有未知数!

- x_t : 输入给网络的当前图像 (已知)

- $\bar{\alpha}_t, \beta_t, \beta_t$: 人为设定超参 (已知)

- ε : 导致 x_0 变成 x_t 的那个噪声. (未知)

\Rightarrow 不需要预测 x_{t-1} , 也不需要预测原图 x_0 .
 训练网络 $E_\theta(x_t, t) \rightarrow$ 预测小噪点
 $\epsilon_\theta \rightarrow \hat{\mu}_t$ 然 $x_{t-1} \rightarrow \dots \rightarrow x_0$
 $\Downarrow \hat{\mu}_t(x_0, x_t) + \sqrt{\beta_t} \epsilon \sim N(0, 1)$

该噪声对于第7个
 逆向步来说是分别预测
 出来的(每步单独预测),
 但合义时从后到前的
 噪声.

2. 关于方差 β_t

- 神经网络在设定方差时直接固定.
- 物理含义: σ : 这步去噪有多大随机性/模糊度.
 μ : 图像长什么样.
- IDDPM (OpenAI) 发现: 预测小学习方差, 生成的图片效果不明显, 但 log-likelihood (对数似然) 会显著提升, 且允许用更少的参数生成图片.

三. 常见问题.

1. 一下取多大的时候, x_t 满足 $N(0, 1)$? Ans. 取大些.

2. Why $\beta_t = 1 - 2t$? i.e. $\mu^2 + \sigma^2 = 1$

Ans. 保证均值 $\rightarrow 0$ 时, 方差 $\rightarrow 1$.

3. 能否跳步?

Ans. $P(x_{t-1}|x_t, x_0)$ 是基于马尔可夫性质建立的, 所以要在逆向过程跳步的话, 需要去马尔可夫化 \rightarrow DDIM.
 (极是一种采样方式).

4. 为什么不用 UNet 去做 $x_T \xrightarrow{\text{预测}} x_{T-1}$?

Ans. 直接去预测 x_{T-1} 类似退化为风格任务. 此时每一个 x_T 都对应一个 x_0 , 即每一步都 deterministic. 取 2 个一样的 x_T 会得到 2 个一样的 x_0 .

但去做 $P(x_{t-1}|x_t)$ 多样性更高. $x_T \xrightarrow{x_0} x_0'$
 (因为涉及采样)

四. Loss Function.

$$L(\theta) = E_{t, x_0, \varepsilon} [\|\varepsilon - \varepsilon_\theta(\sqrt{a_t}x_0 + \sqrt{1-a_t}\varepsilon, t)\|^2]$$

$$Loss = \|\varepsilon - \varepsilon_\theta(x_t, t)\|^2$$

- 上述是简化后的 Loss：将所有 θ 的权重都设为 1，但简化后生成的图片质量反而更好。
- 由下界推导，我们要优化的是两个高斯分布（后验 q 和模型 P ）之间的 KL 散度。

$$L_{vib} = E \left[\sum_{j=1}^T \frac{\beta_t^2}{2\sigma_t^2 a_t(1-\bar{a}_t)} \|\varepsilon - \varepsilon_\theta(x_t, t)\|^2 \right]$$

- Why 简化更好？

Ans. • 加权 Loss 会更多关注 t 很小的步骤（纠结于 imperceptible 的细节），通常是图像几乎完全丢弃的微小噪声。
• 不加权相当增加了 t 较大时的权重，迫使模型更好地学习图像的整体结构和内容。

五. DDPM 中的 UNet 结构。

1. DDPM 中的改动。

- 同一个 UNet 要处理 $t=1 \sim 1000$ 所有情况 \rightarrow 需知道现在第几步。
- 方法：正弦位置编码 (Sinusoidal Positional Embeddings)。
 - 输入：标量 t
 - 编码：正弦/余弦 \rightarrow 高维向量
 - 注入：送入 UNet 的每个残差块中；

$$\text{Scale \& shift : } \text{Feature}_{\text{new}} = \text{Feature}_{\text{old}} \times (1 + \text{Scale}_t) + \text{Shift}_t$$

DDIM

零、引入

1. $X_t = \sqrt{\sigma_t} X_{t-1} + \sqrt{1-\sigma_t} \varepsilon \sim N(0, 1)$ 中, why $\sigma_t \rightarrow 1$ 且 $\sigma_t < 1$?

Ans. 确相邻的加噪过程, 添加的噪音的方差很小, 从而保证加噪前都是正态分布.

2. why $\bar{\sigma}_T \rightarrow 0$?

Ans. $X_t = \sqrt{\sigma_T} X_0 + \sqrt{1-\sigma_T} \varepsilon \approx \varepsilon \sim N(0, 1)$, 这样我们才能从标准正态分布取一个随机噪音并基于此开始降噪.

$\Rightarrow T$ 必须要大. \Rightarrow 通过减小 T 来加速模型 is impossible.

3. 是否可以跳步?

预测目标(噪音的分布): $P(X_{t-1}|X_t, X_0) \leftarrow$ 由马尔可夫性及得
 \Rightarrow 不能跳步.

一、DDIM 的去马尔可夫化.

0. 思想: 寻找一个新的 Non-Markov 的分布 $P(x_s|x_k, x_0)$, 其中 $s < k-1$. (跳步)

* \Rightarrow 设一个不循^连Markov 的采样方程 (但有限制 $P(x_t|x_0)$ 遵循训练过程的公式), 通过待定系数经求出对应的采样方程.

$$P(x_s|x_k, x_0) \sim N\left(\sqrt{\bar{\sigma}_s} x_0 + \frac{\sqrt{1-\bar{\sigma}_s} \bar{\sigma}^2}{\sqrt{1-\bar{\sigma}_k}} (x_k - \sqrt{\bar{\sigma}_k} x_0), \bar{\sigma}^2 I\right)$$

$(s \leq k-1)$

1. 由于训练过程我们没有用 $P(x_k|x_s, x_0)$ ^① 而是用的 $P(x_t|x_0)$ ^②, 所以即使①变化了, 只要②没变, 则模型依然可以用.

二、标准差的选取.

$$\bar{\sigma} = \eta \cdot \sigma_{\text{DDPM}} \Rightarrow \eta = 1 \Rightarrow \text{DDPM}$$

$\eta = 0 \Rightarrow \text{DDIM}$ (效果最好, 但以牺牲多样性为代价)
 (图的质量)

三、思考.

1. 多样性只源自最开始随机噪声 x_T 的初始化, i.e. $t=1 \rightarrow t=0$
这个路径是固定的, 类似于 GAN 中的 Latent space, 因此该过程
可以被用作图像编码 → 迁移学习.

2. 只是改了采样方法, 并没有修改训练方法. 我们降噪过程的
目标: $\underbrace{P(x_{t-1} | x_t, x_0)}_{\text{DDPM}} \rightarrow \underbrace{P(x_s | x_k, \cancel{x_0})}_{\text{DDIM}}$, 为什么 DDPM 直接使
用 DDIM 的采样方法依然有效?

Ans. 由于 UNet 预测的是噪声 ϵ ($\epsilon \rightarrow x_0 \rightarrow P_{\text{DDPM}}$), 所以即使
分布变了, 我们依然用到 x_0 , 此时 ϵ 依旧有效. 模型找的
是 噪声 ϵ 和当前 x_T 之间的关系, 因此依然有效.

3. 当 DDPM 中 $\sigma = 0$ 时, 效果很差, 而
DDIM 中, $\sigma = 0$ 时, 效果最好. why?

Ans. 在 DDIM 中, 我们假设 $P(x_s | x_k, x_0) \sim N(kx_0 + mx_k, \sigma^2)$ 自由
随机变量, 通过解二元一次方程组, 得 k 和 m (即用 σ 表示),
经过实验发现 $\sigma = 0$ 效果最好.

DDPM 中, σ 通过贝叶斯公式, 其中三个已知的概率分布, 推导出来的, 其本身就是正确的.

① 而 DDIM 中的 σ 是不确定的, 当 DDPM 中 $\sigma = 0$, 其采样过程
变成 deterministic, 即向着均值采样, 此过程忽略了方差
的影响导致期望误差累积大 (T 很大).

4. 训练过程依赖 $P(x_t | x_0)$, 则贝叶斯公式相关的两项不能变,
只能变 $P(x_k | x_s, x_0)$

Stable Diffusion.

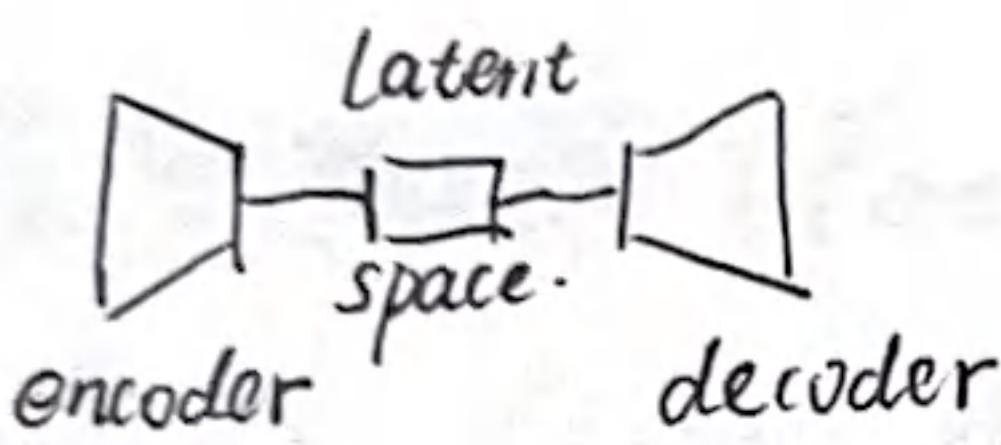
1. 使用了 CLIP (Contrastive Language-Image Pre-training) 中的 Text Encoder

2. Variational AutoEncoder.

(1) 用于压缩前/逆向过程的维度以提高计算效率
↓
latent diffusion.

→ 将学习 data distribution \rightarrow latent representation of the data.

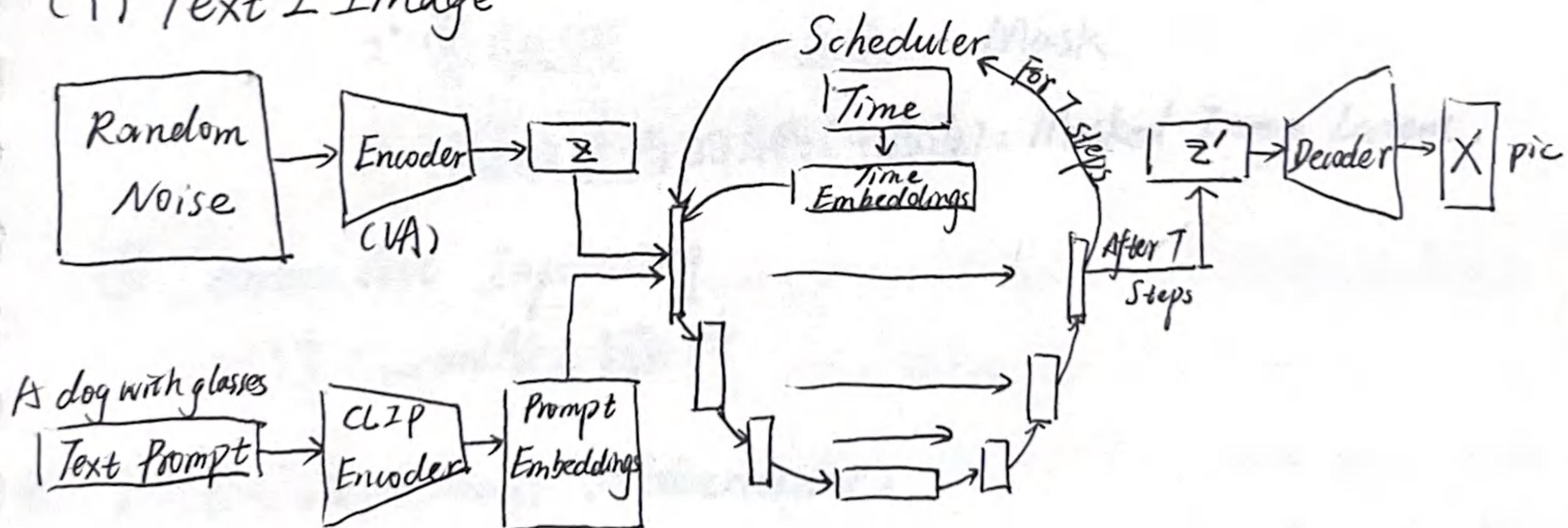
(2) 不压缩, 同时也学习了 latent space.



↑ represents the parameters of a multivariate distribution (多元分布)
(Gaussian) 学的是 μ 和 σ .

3. Architecture.

(1) Text 2 Image



(2) Image 2 Image

将 Random Noise 换成真实图片; encoder 得到的是 Add noise to Latent.

- 添加的噪声少, 生成的自由度广
- 少的噪声 means 不能从根本上改变图片.
- 噪声多少 \Rightarrow 对原图有多大注意力

(3) In-Painting. (fool the model) \rightarrow 引入约束

① 推理时干预 (Inference-time Strategy)

· 代表作: RePaint

· 思想: 不重新训练模型, 在去噪过程作弊.

$$X_{t-1}^{\text{final}} = \underbrace{\text{Mask} \cdot X_{t-1}^{\text{pred}}}_{\begin{array}{l} \text{被遮住的部分} \\ \text{使用模型推断结果} \end{array}} + \underbrace{(1 - \text{Mask}) \cdot X_{t-1}^{\text{known}}}_{\begin{array}{l} \text{保留区域: 不是直接将原图 } X_0 \text{ 贴上} \\ \text{而是前向到 } t-1 \text{ 强度的噪声.} \end{array}}$$

· 问题: 边缘不连贯.

\rightarrow 生成 X_{t-1} 后, 强行加噪回 X_t , 反复横跳.

计算量大

② 模型微调 (Fine-tuning Strategy)

· 代表作: Stable Diffusion Inpainting Model.

· 思想: 将输入通道 4 \rightarrow 9 \Rightarrow UNet.

1° 当前噪声图 (4通道): Z_t

2° 豪版图 (13通道): Mask

3° 被豪版遮挡的原图 (4通道): Masked Image Latent.

③ ControlNet Inpainting

训练 ControlNet 拓件.

4. VAE (Variational Autoencoder)

(1) · 原理: Not learning how to compress the data but learning a latent space which are the parameters of a multivariate Gaussian distribution.

\rightarrow Actually, the VAE is trained to learning the μ and σ .

\rightarrow Then we will sample the distribution.

log6.

5. 核心参数

① strength: 重绘程度

- 决定了生成图和原图有多像 图片还未完全变成噪声.
- 代码逻辑: 起始噪声水平 (从中间步开始降噪)

② do-cfg = True: 启用分类器引导.

- 是否叫懂人话

• 代码逻辑: 如果为True, 模步的每步预测会同时计算2个结果:

- 1° 有提示词预测 (output-cond)
- 2° 无 ~ (output-uncond): 完全猜测, 根据次向prompt 或室内名去预测.

3° 将2种 output 结合

- 如果不开启, 生成虽然自然, 但不会理会prompt.

③ cfg-scale: 引导系数

- 代码逻辑: 最终结果 = 无引导结果 + 系数 (有引导 - 无引导)
- < 2: 不理会 prompt
- > 15: 强行匹配每一个词.

6. 代码小思考

(1) 调度策略: 原DDPM用的 Linear Schedule, SD中用的无开方做插值再平方回去的 Scaled Linear Schedule, why?

Ans. DDPM是在像素空间训练的, SD是在Latent space上训练的, 对于方差和分布更敏感. 采用SLS, 让 β 在初期增长得慢一些, 适合 Latent Space 的信噪比变化规律.

(2) 项目中采用的是Pespaced DDPM, 而非DDIM?

Ans. 因为每步噪声采样的 variance $\neq 0$, 如果为0的话才是DDIM.

(3) VAE中添加的噪声和 diffusion 前后加的噪声的区别?

Ans. VAE把图片压缩后, 得到的是一个概率分布 (μ, σ) ,

需要从中取出一个具体的 Latent 向量 z 为后面的流程用，
因此需要重参数化采样： $\text{Latent} = \text{Mean} + \text{Variance} \times \frac{\text{Noise}}{\sigma}$
没有这个噪音，VAE 无法训练退化为普通的编码器。

$$(x_0, x_1, \dots, x_n) \sim \mathcal{N}(\mu, \Sigma) \quad \frac{(x_0 - \mu)^2 + (x_1 - \mu)^2 + \dots + (x_n - \mu)^2}{n} = \frac{(x - \mu)^T (x - \mu)}{n}$$

$$3 \cdot \frac{(x_0 - \mu)^2 + (x_1 - \mu)^2 + (x_2 - \mu)^2}{3} = 3\sigma^2 + (\mu - \bar{x})^2 = \bar{x}^T \bar{x}$$

$$3 \cdot \frac{(x_0 - \mu)^2 + (x_1 - \mu)^2 + (x_2 - \mu)^2}{3} =$$

$$(\bar{x}^T \bar{x}) \cdot 3 = \bar{x}^T \bar{x} \cdot 3$$

$$(x_0 - \mu)^2 + (x_1 - \mu)^2 + (x_2 - \mu)^2 = \bar{x}^T \bar{x} \cdot 3 = \bar{x}^T \bar{x} \cdot 3$$

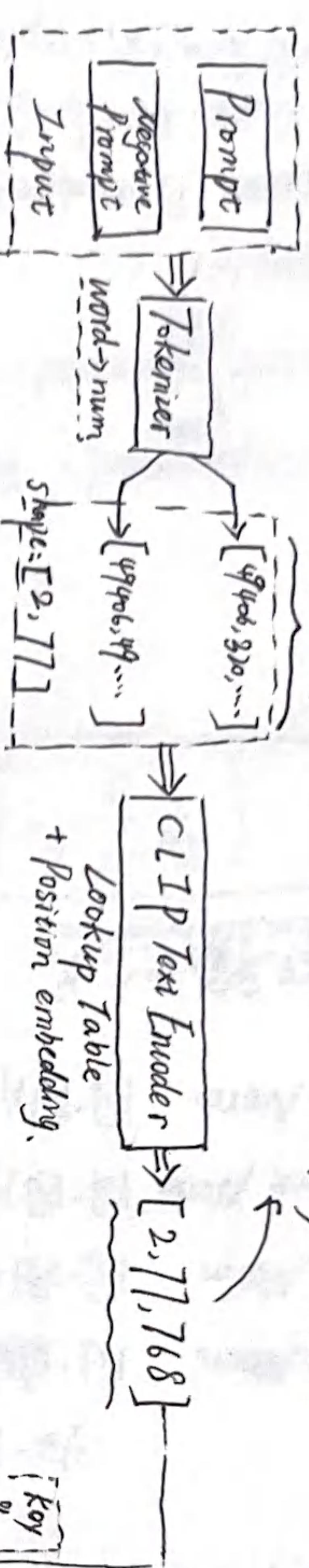
$$(x_0 - \bar{x})^2 + (x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 = \frac{\bar{x}^T \bar{x} - 3\bar{x}^T \bar{x} + 3\bar{x}^T \bar{x}}{3} = \frac{\bar{x}^T \bar{x}}{3}$$

即方差的前项

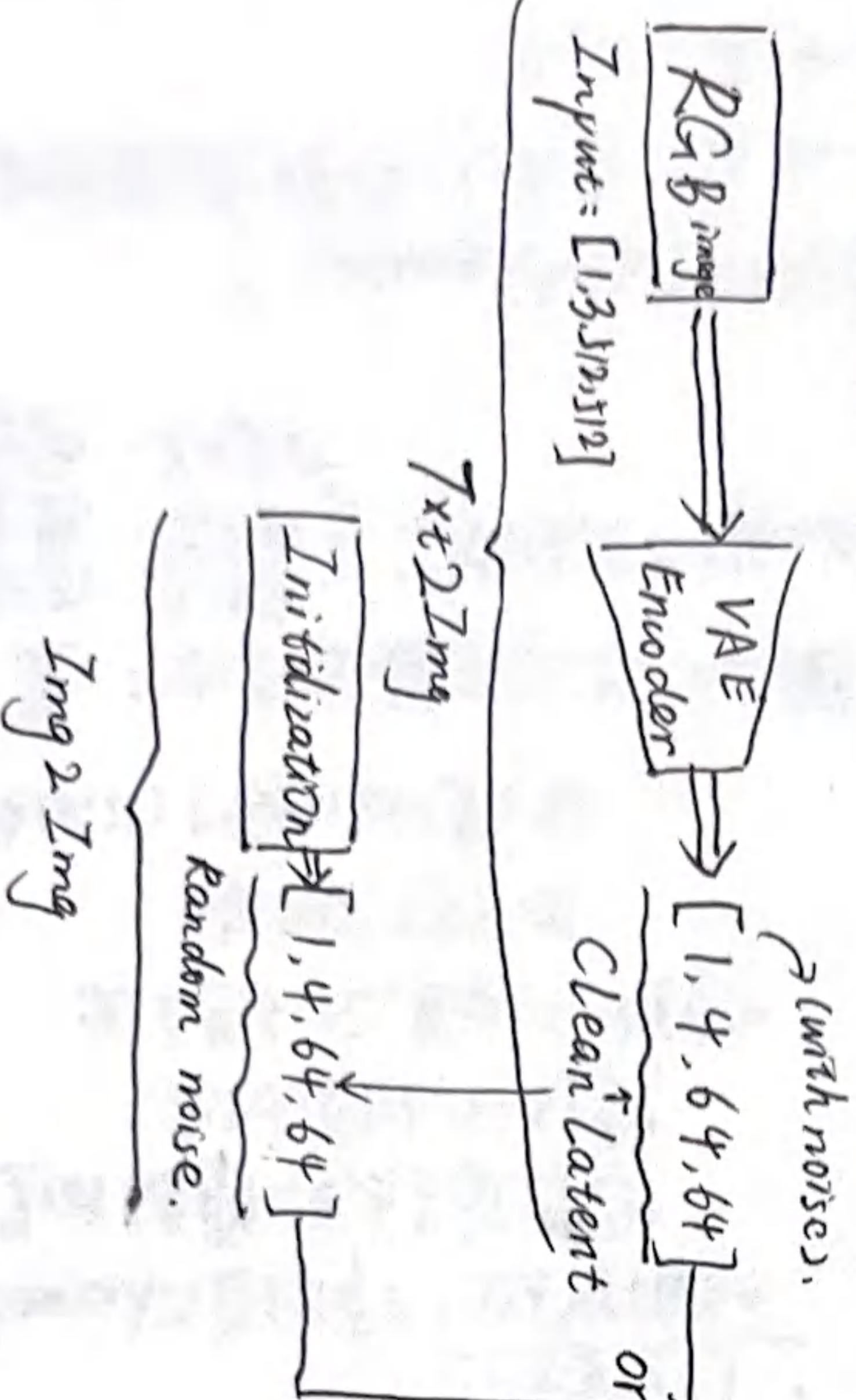
Step 1: Conditioning

length = 77

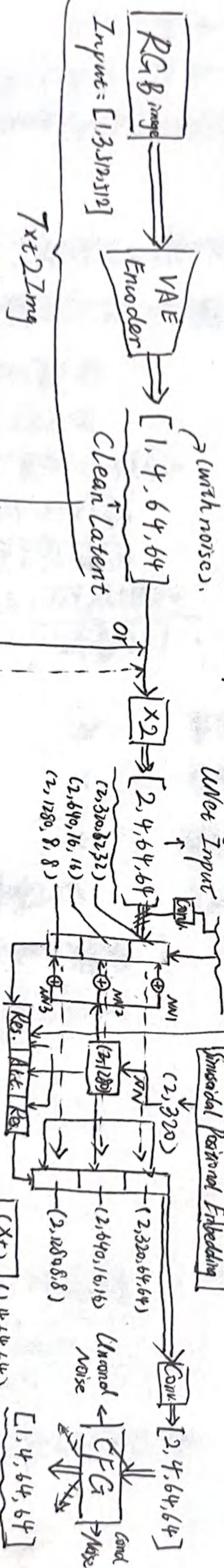
Cross-Attention: key, value.



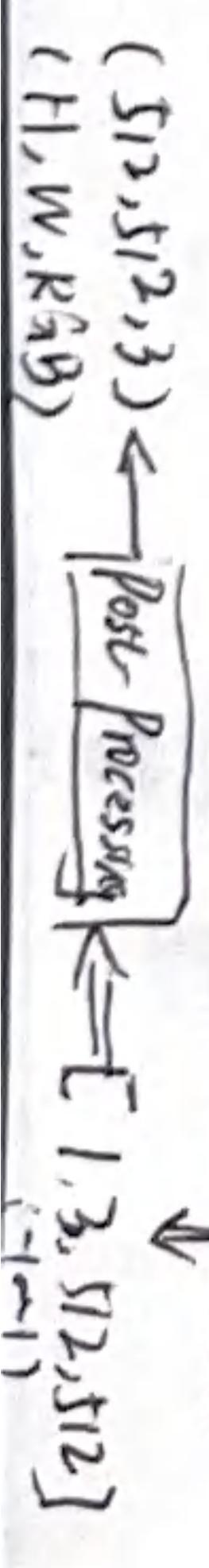
Step 2: Latent Preparation



Step 3: Denoising



Step 4: Decoding



代码知识：

1. SiLu



$$SiLU(x) = x \cdot G(x) = \frac{x}{1+e^{-x}}$$

优势：①〇点可导

②负值区梯度有传递，

防止神经元死亡。

③隐式正则化

2. Normalization

$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} * \gamma + \beta$$

(不改变形状)

标准化 → 射影变换 (γ, β 可学习参数)

[图]

	归一化范围	依赖 batch-size	应用场景	比喻
Batch Norm	同一通道，所有照片	✓ (劣势)	CNN	全校单杆测温
Layer Norm	同一图片，所有通道	✗	NLP	个人综合总分
Instance Norm	同一图片，单一通道	✗	风格迁移GAN	个人单科
Group Norm	同一图片，一组通道	✗	VAE, Stable Diffusion	个人理/文科

3. 卷积输出公式

$$H_{out} = \left\lceil \frac{H_{in} + P - K}{S} \right\rceil + 1$$

• “1x1 卷积”

物理含义：作用在 Channel 维度上的全连接层。
把每个像素点的特征向量做一次矩阵乘法，让特征间相互融合。

作用：① 通道间信息交流

② 升维与降维

③ 增加非线性 (后面加激励函数)。

4. • nn.Sequential：只能一种输入 x ，然后把输出传给下一层。

• SwitchSequential：处理 UNet 每层不同的需求
↓ 包装

• ModuleList : 文件夹

卷积层：图像 x

残差层：图像 $x + \text{时间} t$

注意力层：图像 $x + \text{文本 context}$

▷ Why UNet 用 ModuleList 包裹许多 SwitchSequential?

Ans. UNet 的编码器(Downloader)每经过一层都需保存当前特征图，以便后面解码器(Uploader)时进行 concat.

ControlNet 沈文帝读

一、Introduction

1. 拟解决问题：数据少，算力大

二、Related Works

1. Diffusion

DDPM → DDIM → Latent Diffusion Model.
score-based ↓
解算算力

2. Fine-tuning

(1) Hyper Network：在预训练模型后再加几层神经网络。

(2) zero convolution

3. Text-to-Image Diffusion

(1) CLIP：基于对比学习的文本和图像的多模态模型。
(对图像打标签、对文本编码)

(2) Disco Diffusion

4. Control of Pretrained Diffusion Model

(1) img2img (Stable Diffusion)：color-level detail variations
(2) inpainting：对图中某个区域进行修改。

5. Img2Img

(1) Taming Transformer.

(像素对齐关系)

三、Method.

1. 设计哲学

遗忘 × 很难做到强
↑ 宽泛泛制

以前控制扩散模型用微调/HyperNetwork，所以采用保护与扩展的思想。（保护Stable Diffusion的权重）。

2. ControlNet 网络搭建

(1) 令 SD 的 U-Net 里的一个神经网络层记作 F , 参数为 Θ .

① 输入是 x , 输出是 y .

$$y = F(x; \Theta)$$

边缘图 C

② 一个分支是将 Θ 锁定 $\rightarrow \Theta_{lock}$

另一分支复制相同结构 $\rightarrow \Theta_{trainable}$. 为了可以接收 额外输入, 该分支的输入为 $x + \mathcal{Z}(c)$. \mathcal{Z} 是简单的特征提取器.

③ Zero Convolution.

(2) 不能直接将 trainable copy 的输出加回去.

引入零卷积层 $\mathcal{Z} \rightarrow \mathcal{Z}(\cdot; \Theta_{zero})$
其中 W, B 初始化为 0.

$$y_c = F(x; \Theta_{lock}) + \mathcal{Z}(F(x + \mathcal{Z}(c); \Theta_{trainable}); \Theta_{zero})$$

(2) Why Zero Convolution?

在训练的第一步, 零卷积的输出是零 $\Rightarrow y_c = f(x; \Theta_{lock}) + 0 = y$
which means, 在训练开始时, ControlNet 对模型的影响为 0, 等价于原本的 SD \rightarrow 保证了训练的稳定性.

逐步调整 Θ_{zero} 和 $\Theta_{trainable}$, 逐步注入控制信息.

(3) 复制了什么?

只复制了 Encoder 和 Middle Block (共 13 个模块), 并没有复制 Decoder. \leftarrow 为了省参数

注入点: 加到 SD 的 decoder 的每一层上.

ControlNet 的每一层输出, 会作为残差加到相应的 SD 层的输出上, 然后一起进入下一层的 SD-Decoder 中.

3. 训练策略.

(1) 数据构建：(原图 x_0 , Prompt, Condition Map(c))
↑
自动生成

(2) Loss

$$L = E_{x_0, t, c, \epsilon} [\|\epsilon - \Sigma_0(x_t, t, c)\|^2]$$

· SD权重锁死，梯度只会回传到 copy 层和零卷积层。

(3) 突然收敛/顿悟 (Grokking) 现象。

(4) 空文本训练 (classifier-Free Guidance Support)

· 为了让 ControlNet 在推理时能够调节 Prompt 对生成结果产生影响，训练时必须采用 Dropout 策略：50% 将 Text Prompt 替换为 “ ” 字符串。

→ 迫使不仅仅依赖文本，学会从 Condition Map 寻找线索。

(5) Zero convolution 中的 backpropagation.

① 前向： $y = wx + b$ $\xrightarrow[\text{Step}]{\text{first}} 0 \cdot x + 0 = 0 \Rightarrow$ 传给 SD 的是 0。

② 反向：需要更新权重 w. (根据 L)

$$\text{Chain Law: } \frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial w} = \frac{\partial L}{\partial y} \cdot x$$

⇒ 只要 输入不是 0，并且模型有误差，w 就会获得 Δ，从而更新 ControlNet 的

(4) ControlNet 的头部有个卷积网络，用来做降维和匹配通道数与 U-Net 输入一致。

卷积与全连接的联系

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & \cdots & \cdots & \cdots & \cdots \\ w_{31} & & & & \\ w_{41} & & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

→ 计算 y_1 需用到 $x_1 \sim x_5$
的所有输入信息。

• $[a, b]$ 的卷积核：

$$y_1 = ax_1 + bx_2$$

$$y_2 = ax_2 + bx_3$$

⋮

$$y_4 = ax_4 + bx_5$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} a & b & 0 & & \\ a & b & 0 & & \\ 0 & a & b & 0 & \\ & a & b & 0 & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

稀疏连接、局部感受野 \Leftarrow Attention 改进
参数共享、平移不变性 (对特征位置不敏感)

Conclusion: 卷积是一种被限制了自由度, 但换来了极高效率和泛化能力的全连接层。