# Lec 05 Building An Image Generator

之前我们已经学习了非条件生成模型

- **Problem**：从$p_{\text{data}}$中采样
- **Train**：利用条件流匹配，即

$$\mathcal{L}_{CFM}(\theta) = \boldsymbol{E}_{t\sim \text{Unif}, z\sim p_{\text{data}}, x\sim p_t(\cdot|z)} \left[ \left\| u_t^\theta(x) - u_t^{\text{target}}(x|z) \right\|^2 \right]$$

- **Sample**：模拟对应的ODE（或者SDE），即

$$dX_t = u_t^\theta(X_t)dt, \quad X_0 \sim p_{\text{init}}$$

# Guidance

到目前为止，我们所考虑的生成模型都是无条件的，例如，一个图像模型只是生成某个图像。然而，任务不仅仅是生成任意对象，而是要根据一些额外的信息来生成对象。例如，一个以文本提示$y$为输入的生成模型，生成一个基于文本提示$y$的图片$x$。那么对于一个固定的文本提示词$y$，我们就需要从$p_{\text{data}}(x|y)$这个分布中进行采样。我们假设$y$是空间$\mathcal{Y}$中的一个向量，当$y$对应某些离散的类别时，$\mathcal{Y}$也是离散的空间。用最经典的MNIST为例子，则$\mathcal{Y} = \{0, 1, \dots, 9\}$。

**Guided Generation: What Changes?**

| Unguided | | Guided | |
|---|---|---|---|
| Marginal probability path | $p_t(x)$ | **Guided** marginal probability path | $p_t(x\|y)$ |
| Marginal vector field | $u_t^{\text{target}}(x)$ | **Guided** marginal vector field | $u_t^{\text{target}}(x\|y)$ |
| Marginal score | $\nabla \log p_t(x)$ | **Guided** marginal score | $\nabla \log p_t(x\|y)$ |
| Model | $u_t^\theta(x)$ | **Guided** model | $u_t^\theta(x\|y)$ |
| CFM Objective | $\mathcal{L}_{\text{CFM}}(\theta)$ | **Guided** CFM Objective | ??? |

我们可以定义一个Guided Diffusion Model，其包含一个Guided Vector Field $u_t^\theta(\cdot|y)$、一个与时间相关的扩散系数$\sigma_t$，即

$$u^\theta : R^d \times \mathcal{Y} \times [0,1] \to R^d, \quad (x, y, t) \mapsto u_t^\theta(x|y)$$

$$\sigma_t : [0,1] \to [0, \infty], \quad t \mapsto \sigma_t$$

## Guidance for Flow Model

假如提示词$y$是固定的，那么问题就从条件生成模型恢复成非条件生成模型，我们的数据分布变成了$p_{\text{data}}(\cdot|y)$，因此我们可以构建条件流匹配目标，即

$$\mathcal{L}_{CFM}^{\text{guided}}(\theta; y) = \boldsymbol{E}_{z\sim p_{\text{data}}(\cdot|y), x\sim p_t(\cdot|z)} \left[ \left\| u_t^\theta(x|y) - u_t^{\text{target}}(x|z) \right\|^2 \right]$$

由于提示词$y$与条件概率路径$p_t(\cdot|z)$以及条件向量场$u_t^{\text{target}}(x|z)$无关，我们可以对$y$在$\mathcal{Y}$内的所有取值求期望值，对时间$t \in [0,1)$也是同理，因此我们就可以构建Guided Conditional Flow Matching目标，即

$$\mathcal{L}_{CFM}^{\text{guided}}(\theta) = \boldsymbol{E}_{(z,y)\sim p_{\text{data}}(z,y),t\sim\text{Unif}[0,1),x\sim p_t(\cdot|z)}\left[\left\|u_t^\theta(x|y) - u_t^{\text{target}}(x|z)\right\|^2\right]$$

**生成样本过程**：

**Algorithm 7** Guided Sampling Procedure

**Require:** A trained guided vector field $u_t^\theta(x|y)$.

1: Select a prompt $y \in \mathcal{Y}$, such as "a cat baking a cake".
2: Initialize $X_0 \sim p_{\text{init}}$.
3: Simulate $\mathrm{d}X_t = u_t^\theta(X_t|y)\mathrm{d}t$ from $t = 0$ to $t = 1$.

## Classifier-Free Guidance

上述的模型虽然理论上是正确的，但是实操的时候发现生成的图像不会完全符合提示词$y$。人们发现，当人为增强引导变量 y 的作用时，感知质量会得到提升。这一见解被提炼成一种称为无分类器引导(Classifier-Free Guidance)的技术，该技术在最先进的扩散模型中得到了广泛应用，接下来我们将对此进行讨论。为简单起见，这里我们将重点关注高斯概率路径的情况。

已知高斯条件概率路径为：

$$p_t(\cdot|z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d)$$

其中$\alpha_t$和$\beta_t$都是可微、单调的，并且满足$\alpha_0 = \beta_1 = 0$和$\alpha_1 = \beta_0 = 1$。通过引导分数函数$\nabla \log p_t(x|y)$，我们可以重写引导向量场$u_t^{\text{target}}(x|y)$，即

$$u_t^{\text{target}}(x|y) = a_t x + b_t \nabla \log p_t(x|y)$$

$$(a_t, b_t) = \left(\frac{\dot{\alpha}_t}{\alpha_t}, \frac{\dot{\alpha}_t\beta_t^2 - \dot{\beta}_t\beta_t\alpha_t}{\alpha_t}\right)$$

由于梯度是对于$x$计算梯度，所以$\nabla p_t(y) = 0$，因此：

$$\nabla \log p_t(x|y) = \nabla \log \frac{p_t(x)p_t(y|x)}{p_t(y)} = \nabla \log p_t(x) + \nabla p_t(y|x)$$

代入原式，我们可以得到：

$$u_t^{\text{target}}(x|y) = a_t x + b_t\left(\nabla \log p_t(x) + \nabla p_t(y|x)\right) = u_t^{\text{target}}(x) + b_t \nabla \log p_t(y|x)$$

这是原本的模型，引导向量场$u_t^{\text{target}}(x|y)$是非引导向量场$u_t^{\text{target}}(x)$与引导分数$\nabla \log p_t(y|x)$之和，为了使得图像$x$更加符合提示词$y$，我们可以放大$\nabla \log p_t(y|x)$这一项，也就是将第二项乘以一个大于1的系数$\omega$，得到：

$$\tilde{u}_t(x|y) = u_t^{\text{target}}(x) + \omega b_t \nabla \log p_t(y|x), \quad \omega > 1$$

其中$\omega > 1$被称为Guidance Scale。在乘以系数之后，我们可以再将式子化简：

$$\begin{aligned}
\tilde{u}_t(x|y) &= u_t^{\text{target}}(x) + \omega b_t \nabla \log p_t(y|x) \\
&= u_t^{\text{target}}(x) + \omega b_t\left(\nabla \log p_t(x|y) - \nabla \log p_t(x)\right) \\
&= u_t^{\text{target}}(x) - (\omega a_t x + \omega b_t \nabla \log p_t(x)) + (\omega a_t x + \omega b_t \nabla \log p_t(x|y)) \\
&= (1-\omega)u_t^{\text{target}}(x) + \omega u_t^{\text{target}}(x|y)
\end{aligned}$$

实际上我们可以把$u_t^{\text{target}}(x)$视为$u_t^{\text{target}}(x|\emptyset)$，即当$y = \emptyset$时$x$的条件向量场。因此我们就可以训练一个单独的模型$u_t^\theta(x|y)$，其中$y \in \{\mathcal{Y}, \emptyset\}$，则条件流匹配损失函数可以写成：

$$\mathcal{L}_{CFM}^{CFG}(\theta) = \boldsymbol{E}_{(z,y)\sim p_{\text{data}}(z,y),t\sim\text{Unif}[0,1),x\sim p_t(\cdot|z),\text{replace } y=\emptyset \text{ with prob. } \eta}\left[\left\|u_t^\theta(x|y) - u_t^{\text{target}}(x|z)\right\|^2\right]$$

**生成样本过程**：

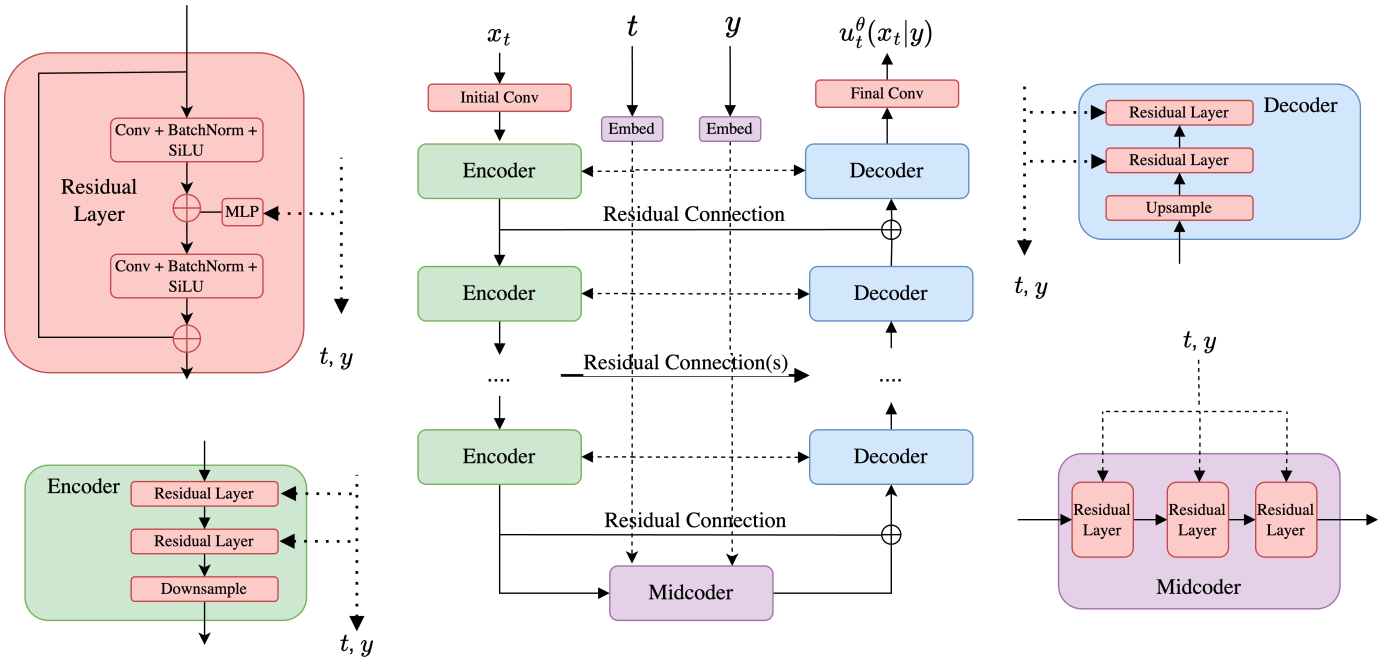
**Algorithm 8** Classifier-Free Guidance Sampling Procedure

**Require:** A trained guided vector field $u_t^\theta(x|y)$.

1: Select a prompt $y \in \mathcal{Y}$, or take $y = \varnothing$ for unguided sampling.
2: Select a **guidance scale** $w > 1$.
3: Initialize $X_0 \sim p_{\text{init}}$.
4: Simulate $\mathrm{d}X_t = \left[(1-w)u_t^\theta(X_t|\varnothing) + wu_t^\theta(X_t|y)\right]\mathrm{d}t$ from $t = 0$ to $t = 1$.

# Architectural Considerations for Image Generation

## U-Nets

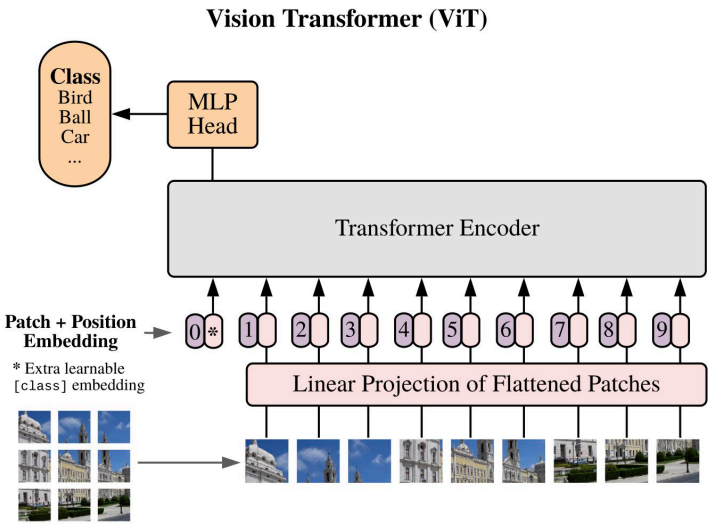


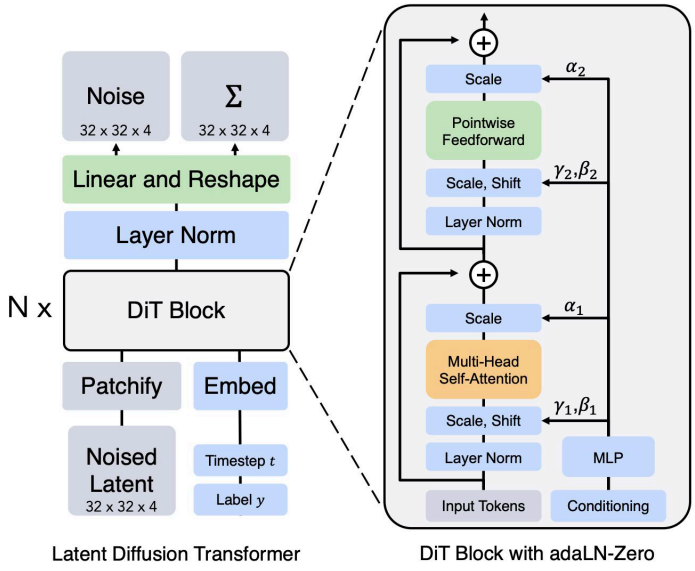

U-Net 架构是一种特定类型的卷积神经网络。最初是为图像分割而设计的，其关键特征在于其输入和输出都具有图像的形状（可能通道数不同）。由于对于特定的$y, t$，其输入和输出图片的形状相同，所以它特别时候用来参数化向量场$x \mapsto u_t^\theta(x|y)$。因此，U-Net 在扩散模型的开发中得到了广泛应用。U-Net 由一系列编码器$\mathcal{E}_i$，以及对应的解码器$\mathcal{D}_i$组成，中间存在一个潜在的处理模块，我们将其称为Midcoder.

请注意，随着输入通过编码器，其表示中的通道数量会增加，而图像的高度和宽度则会减小。编码器和解码器通常都由一系列卷积层组成（层与层之间有激活函数、池化操作等）。编码器和解码器通常通过残差连接相连接。然而，上述描述中的某些设计选择可能与实际中的各种实现方式有所不同。特别是，我们在上文中选择了纯卷积架构，而通常在编码器和解码器中也会加入注意力层。U-Net 因其编码器和解码器形成的类似“U”形而得名。

## Diffusion Transformers

U-Nets 的一种替代方案是扩散转换器（DiTs），它摒弃了卷积，纯粹使用注意力机制。扩散转换器基于视觉转换器（ViTs），其核心思想基本上是将图像分割成多个部分，对每个部分进行嵌入，然后在这些部分之间进行注意力处理。




Latent Diffusion Transformer

DiT Block with adaLN-Zero