30 Aug

生成扩散模型漫谈（九）：条件控制生成结果

By 苏剑林 | 2022-08-30 | 135853位读者 引用

前面的几篇文章都是比较偏理论的结果，这篇文章我们来讨论一个比较有实用价值的主题——条件控制生成。

作为生成模型，扩散模型跟VAE、GAN、flow等模型的发展史很相似，都是先出来了无条件生成，然后有条件生成就紧接而来。无条件生成往往是为了探索效果上限，而有条件生成则更多是应用层面的内容，因为它可以实现根据我们的意愿来控制输出结果。从DDPM至今，已经出来了很多条件扩散模型的工作，甚至可以说真正带火了扩散模型的就是条件扩散模型，比如脍炙人口的文生图模型DALL·E 2、Imagen。

在这篇文章中，我们对条件扩散模型的理论基础做个简单的学习和总结。

技术分析

从方法上来看，条件控制生成的方式分两种：事后修改（Classifier-Guidance）和事前训练（Classifier-Free）。

对于大多数人来说，一个SOTA级别的扩散模型训练成本太大了，而分类器（Classifier）的训练还能接受，所以就想着直接复用别人训练好的无条件扩散模型，用一个分类器来调整生成过程以实现控制生成，这就是事后修改的Classifier-Guidance方案；而对于“财大气粗”的Google、OpenAI等公司来说，它们不缺数据和算力，所以更倾向于往扩散模型的训练过程中就加入条件信号，达到更好的生成效果，这就是事前训练的Classifier-Free方案。

Classifier-Guidance方案最早出自《Diffusion Models Beat GANs on Image Synthesis》，最初就是用来实现按类生成的；后来《More Control for Free! Image Synthesis with Semantic Diffusion Guidance》推广了“Classifier”的概念，使得它也可以按图、按文来生成。Classifier-Guidance方案的训练成本比较低（熟悉NLP的读者可能还会想起与之很相似的PPLM模型），但是推断成本会高些，而且控制细节上通常没那么到位。

至于Classifier-Free方案，最早出自《Classifier-Free Diffusion Guidance》，后来的DALL·E 2、Imagen等吸引人眼球的模型基本上都是以它为基础做的，值得一提的是，该论文上个月才放到Arxiv上，但事实上去年已经中了NeurIPS 2021。应该说，Classifier-Free方案本身没什么理论上的技巧，它是条件扩散模型最朴素的方案，出现得晚只是因为重新训练扩散模型的成本较大吧，在数据和算力都比较充裕的前提下，Classifier-Free方案表现出了令人惊叹的细节控制能力。

条件输入

说白了，Classifier-Free方案就是训练成本大，本身“没什么技术含量”，所以接下来的主要篇幅都是Classifier-Guidance方案，而Classifier-Free方案则是在最后简单介绍一下。

经过前面一系列文章的分析，想必读者已经知道，生成扩散模型最关键的步骤就是生成过程$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$的构建，而对于以$\boldsymbol{y}$为输入条件的生成来说，无非就是将$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$换成$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y})$而已，也就是说生成过程中增加输入$\boldsymbol{y}$。为了重用已经训练好的无条件生成模型$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$，我们利用贝叶斯定理得

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{y}) = \frac{p(\boldsymbol{x}_{t-1})p(\boldsymbol{y}|\boldsymbol{x}_{t-1})}{p(\boldsymbol{y})} \tag{1}$$

在每一项上面补上条件$\boldsymbol{x}_t$，就得到

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y}) = \frac{p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)p(\boldsymbol{y}|\boldsymbol{x}_{t-1}, \boldsymbol{x}_t)}{p(\boldsymbol{y}|\boldsymbol{x}_t)} \tag{2}$$

注意，在前向过程中，$\boldsymbol{x}_t$是由$\boldsymbol{x}_{t-1}$加噪声得到的，噪声不会对分类有帮助，所以$\boldsymbol{x}_t$的加入对分类不会有任何收益，因此有$p(\boldsymbol{y}|\boldsymbol{x}_{t-1}, \boldsymbol{x}_t) = p(\boldsymbol{y}|\boldsymbol{x}_{t-1})$，从而

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y}) = \frac{p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)p(\boldsymbol{y}|\boldsymbol{x}_{t-1})}{p(\boldsymbol{y}|\boldsymbol{x}_t)} = p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)e^{\log p(\boldsymbol{y}|\boldsymbol{x}_{t-1}) - \log p(\boldsymbol{y}|\boldsymbol{x}_t)} \tag{3}$$

近似分布

对于已经看过《生成扩散模型漫谈（五）：一般框架之SDE篇》的读者，大概会觉得接下来的过程似曾相识。不过即便没读过也不要紧，下面我们依旧完整推导一下。

当T足够大时，$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$的方差足够小，也就是说只有$\boldsymbol{x}_t$与$\boldsymbol{x}_{t-1}$很接近时概率才会明显大于0。反过来也是成立的，即也只有$\boldsymbol{x}_t$与$\boldsymbol{x}_{t-1}$很接近时$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{y})$或$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{y})$才明显大于0，我们只需要重点考虑这个范围内的概率变化。为此，我们用泰勒展开：

$$\log p(\boldsymbol{y}|\boldsymbol{x}_{t-1}) - \log p(\boldsymbol{y}|\boldsymbol{x}_t) \approx (\boldsymbol{x}_{t-1} - \boldsymbol{x}_t)\cdot\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t) \tag{4}$$

严格来讲还有一项关于t的变化项，但是那一项跟$\boldsymbol{x}_{t-1}$无关，属于不影响$\boldsymbol{x}_{t-1}$概率的常数项，因此我们没有写出。假设原来有$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1};\boldsymbol{\mu}(\boldsymbol{x}_t),\sigma_t^2\boldsymbol{I})\propto e^{-\Vert\boldsymbol{x}_{t-1}-\boldsymbol{\mu}(\boldsymbol{x}_t)\Vert^2/2\sigma_t^2}$，那么此时近似地有

$$\begin{aligned}p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{y}) \propto&\, e^{-\Vert\boldsymbol{x}_{t-1}-\boldsymbol{\mu}(\boldsymbol{x}_t)\Vert^2/2\sigma_t^2 + (\boldsymbol{x}_{t-1} - \boldsymbol{x}_t)\cdot\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t)}\\ \propto&\, e^{-\Vert\boldsymbol{x}_{t-1}-\boldsymbol{\mu}(\boldsymbol{x}_t)-\sigma_t^2\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t))\Vert^2/2\sigma_t^2}\end{aligned} \tag{5}$$

从这个结果可以看出，$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{y})$近似于$\mathcal{N}(\boldsymbol{x}_{t-1};\boldsymbol{\mu}(\boldsymbol{x}_t)+\sigma_t^2\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t),\sigma_t^2\boldsymbol{I})$，所以只需要把生成过程的采样改为

$$\boldsymbol{x}_{t-1} = \boldsymbol{\mu}(\boldsymbol{x}_t) + \underbrace{\sigma_t^2\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t)}_{\text{新增项}} + \sigma_t\boldsymbol{\varepsilon},\quad \boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}) \tag{6}$$

这就是Classifier-Guidance方案的核心结果。值得注意的是，本文的推导结果跟原论文略有不同，原论文新增项是

$$\sigma_t^2\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t)\big|_{\boldsymbol{x}_t=\boldsymbol{\mu}(\boldsymbol{x}_t)} \tag{7}$$

也就是梯度项在$\boldsymbol{\mu}(\boldsymbol{x}_t)$处的结果而非$\boldsymbol{x}_t$处，而一般情况下$\boldsymbol{\mu}(\boldsymbol{x}_t)$的零阶近似正是$\boldsymbol{x}_t$，所以两者结果是差不多的。

梯度缩放

原论文（《Diffusion Models Beat GANs on Image Synthesis》）发现，往分类器的梯度中引入一个缩放参数γ，可以更好地调节生成效果：

$$\boldsymbol{x}_{t-1} = \boldsymbol{\mu}(\boldsymbol{x}_t) + \sigma_t^2 \gamma \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\boldsymbol{x}_t) + \sigma_t \boldsymbol{\varepsilon},\quad \boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}) \tag{8}$$

当$\gamma > 1$时，生成过程将使用更多的分类器信号，结果将会提高生成结果与输入信号$\boldsymbol{y}$的相关性，但是会相应地降低生成结果的多样性；反之，则会降低生成结果与输入信号之间的相关性，但增加了多样性。

怎么从理论上理解这个参数呢？原论文提出将它理解为通过幂操作来提高分布的聚焦程度，即定义

$$\tilde{p}(\boldsymbol{y}|\boldsymbol{x}_t) = \frac{p^{\gamma}(\boldsymbol{y}|\boldsymbol{x}_t)}{Z(\boldsymbol{x}_t)},\quad Z(\boldsymbol{x}_t) = \sum_{\boldsymbol{y}} p^{\gamma}(\boldsymbol{y}|\boldsymbol{x}_t) \tag{9}$$

随着γ的增加，$\tilde{p}(\boldsymbol{y}|\boldsymbol{x}_t)$的预测会越来越接近one hot分布，用它来代替$p(\boldsymbol{y}|\boldsymbol{x}_t)$作为分类器做Classifier-Guidance，生成过程会倾向于挑出分类置信度很高的样本。

然而，这个角度虽然能提供一定的参考价值，但其实不完全对，因为

$$\nabla_{\boldsymbol{x}_t} \log \tilde{p}(\boldsymbol{y}|\boldsymbol{x}_t) = \gamma\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \log Z(\boldsymbol{x}_t) \neq \gamma\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\boldsymbol{x}_t) \tag{10}$$

原论文错误地认为$Z(\boldsymbol{x}_t)$是一个常数，所以$\nabla_{\boldsymbol{x}_t} \log Z(\boldsymbol{x}_t) = 0$，但事实上$\gamma \neq 1$时，$Z(\boldsymbol{x}_t)$会显式地依赖于$\boldsymbol{x}_t$。笔者也继续思考了一下有没有什么补救方法，但很遗憾没什么结果，仿佛只能很勉强地认为$\gamma = 1$时（此时$Z(\boldsymbol{x}_t) = 1$）的梯度性质能近似地泛化到$\gamma \neq 1$的情形。

相似控制

事实上，理解$\gamma \neq 1$的最佳方案，就是放弃从贝叶斯定理的式(2)和式(3)来理解$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y})$，而是直接定义

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y}) = \frac{p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) e^{\gamma\cdot\text{sim}(\boldsymbol{x}_{t-1},\boldsymbol{y})}}{Z(\boldsymbol{x}_t,\boldsymbol{y})},\quad Z(\boldsymbol{x}_t,\boldsymbol{y}) = \sum_{\boldsymbol{x}_{t-1}} p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) e^{\gamma\cdot\text{sim}(\boldsymbol{x}_{t-1},}$$

其中$\text{sim}(\boldsymbol{x}_{t-1},\boldsymbol{y})$是生成结果$\boldsymbol{x}_{t-1}$与条件$\boldsymbol{y}$的某个相似或相关度量。在这个角度下，$\gamma$直接融于$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{y})$的定义中，直接控制结果与条件的相关性，当$\gamma$越大，模型会倾向于生成跟$\boldsymbol{y}$越相关的$\boldsymbol{x}_{t-1}$。

为了进一步得到可采样的近似结果，我们可以在$\boldsymbol{x}_{t-1}=\boldsymbol{x}_t$处（也可以在$\boldsymbol{x}_{t-1}=\boldsymbol{\mu}(\boldsymbol{x}_t)$，跟前面类似）展开

$$e^{\gamma\cdot\text{sim}(\boldsymbol{x}_{t-1},\boldsymbol{y})}\approx e^{\gamma\cdot\text{sim}(\boldsymbol{x}_t,\boldsymbol{y})+\gamma\cdot(\boldsymbol{x}_{t-1}-\boldsymbol{x}_t)\cdot\nabla_{\boldsymbol{x}_t}\text{sim}(\boldsymbol{x}_t,\boldsymbol{y})}\tag{12}$$

假设此近似程度已经足够，那么除去与$\boldsymbol{x}_{t-1}$无关的项，我们得到

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{y})\propto p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)e^{\gamma\cdot(\boldsymbol{x}_{t-1}-\boldsymbol{x}_t)\cdot\nabla_{\boldsymbol{x}_t}\text{sim}(\boldsymbol{x}_t,\boldsymbol{y})}\tag{13}$$

跟前面一样，代入$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)=\mathcal{N}(\boldsymbol{x}_{t-1};\boldsymbol{\mu}(\boldsymbol{x}_t),\sigma_t^2\boldsymbol{I})$，配方后得到

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{y})\approx\mathcal{N}(\boldsymbol{x}_{t-1};\boldsymbol{\mu}(\boldsymbol{x}_t)+\sigma_t^2\gamma\nabla_{\boldsymbol{x}_t}\text{sim}(\boldsymbol{x}_t,\boldsymbol{y}),\sigma_t^2\boldsymbol{I})\tag{14}$$

这样一来，我们就不需要纠结$p(\boldsymbol{y}|\boldsymbol{x}_t)$的概率意义，而是只需要直接定义度量函数$\text{sim}(\boldsymbol{x}_t,\boldsymbol{y})$，这里的$\boldsymbol{y}$也不再是仅限于“类别”，也可以是文本、图像等任意输入信号，通常的处理方式是用各自的编码器将其编码为特征向量，然后用cos相似度：

$$\text{sim}(\boldsymbol{x}_t,\boldsymbol{y})=\frac{E_1(\boldsymbol{x}_t)\cdot E_2(\boldsymbol{y})}{\Vert E_1(\boldsymbol{x}_t)\Vert\Vert E_2(\boldsymbol{y})\Vert}\tag{15}$$

要指出的是，中间过程的$\boldsymbol{x}_t$是带高斯噪声的，所以编码器E_1一般不能直接调用干净数据训练的编码器，而是要用加噪声后的数据对它进行微调才比较好。此外，如果做风格迁移的，通常则是用Gram矩阵距离而不是cos相似度，这些都看场景发挥了。以上便是论文《More Control for Free! Image Synthesis with Semantic Diffusion Guidance》的一系列结果，更多细节可以自行参考原论文。

连续情形

经过前面的推导，我们得到均值的修正项为$\sigma_t^2\gamma\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t)$或$\sigma_t^2\gamma\nabla_{\boldsymbol{x}_t}\text{sim}(\boldsymbol{x}_t,\boldsymbol{y})$，它们都有一个共同特点，就是$\sigma_t=0$时，修正项也等于0，修正

就失效了。

那么生成过程的σ_t可以等于0吗？肯定可以，比如《生成扩散模型漫谈（四）：DDIM = 高观点DDPM》介绍的DDIM，就是方差为0的生成过程，这种情况下应该怎样做控制生成呢？此时我们需要用到《生成扩散模型漫谈（六）：一般框架之ODE篇》介绍的基于SDE的一般结果了，在里边我们介绍到，对于前向SDE：

$$d\boldsymbol{x} = \boldsymbol{f}_t(\boldsymbol{x})dt + g_t d\boldsymbol{w} \tag{16}$$

对应的最一般的反向SDE为

$$d\boldsymbol{x} = \left(\boldsymbol{f}_t(\boldsymbol{x}) - \frac{1}{2}(g_t^2 + \sigma_t^2)\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})\right)dt + \sigma_t d\boldsymbol{w} \tag{17}$$

这里允许我们自由选择反向方差σ_t^2，DDPM、DDIM都可以认为是它的特例，其中$\sigma_t = 0$时就是一般化的DDIM。可以看到，反向SDE跟输入有关的就是$\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})$，如果要做条件生成，自然是要将它换成$\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x}|\boldsymbol{y})$，然后利用贝叶斯定理，有

$$\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x}|\boldsymbol{y}) = \nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{y}|\boldsymbol{x}) \tag{18}$$

在一般的参数化下有$\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x}) = -\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)}{\bar{\beta}_t}$，因此

$$\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x}|\boldsymbol{y}) = -\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)}{\bar{\beta}_t} + \nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{y}|\boldsymbol{x}) = -\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t) - \bar{\beta}_t\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{y}|}{\bar{\beta}_t}$$

这就意味着，不管生成方差是多少，我们只需要用$\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t) - \bar{\beta}_t\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{y}|\boldsymbol{x})$代替$\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)$就可以实现条件控制生成了。因此，在SDE的统一视角下，我们可以非常简单而直接地得到Classifier-Guidance方案的最一般结果。

无分类器

最后，我们来简单介绍一下Classifier-Free方案。其实很简单，它就是直接定义

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{y}) = \mathcal{N}(\boldsymbol{x}_{t-1};\boldsymbol{\mu}(\boldsymbol{x}_t,\boldsymbol{y}),\sigma_t^2\boldsymbol{I}) \tag{20}$$

沿用前面DDPM的几篇文章的结果，$\boldsymbol{\mu}(\boldsymbol{x}_t,\boldsymbol{y})$一般参数化为

$$\boldsymbol{\mu}(\boldsymbol{x}_t,\boldsymbol{y})=\frac{1}{\alpha_t}\left(\boldsymbol{x}_t-\frac{\beta_t^2}{\bar{\beta}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,\boldsymbol{y},t)\right) \tag{21}$$

训练的损失函数就是

$$\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{y}\sim\tilde{p}(\boldsymbol{x}_0,\boldsymbol{y}),\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\left\Vert\boldsymbol{\varepsilon}-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t\boldsymbol{x}_0+\bar{\beta}_t\boldsymbol{\varepsilon},\boldsymbol{y},t)\right\Vert^2\right] \tag{22}$$

它的优点是在训练过程中就引入了额外的输入$\boldsymbol{y}$，理论上输入信息越多越容易训练；它的缺点也是在训练过程中就引入了额外的输入$\boldsymbol{y}$，意味着每做一组信号控制，就要重新训练整个扩散模型。

特别地，Classifier-Free方案也模仿Classifier-Guidance方案加入了γ参数的缩放机制来平衡相关性与多样性。具体来说，式(8)的均值可以改写成：

$$\boldsymbol{\mu}(\boldsymbol{x}_t)+\sigma_t^2\gamma\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t)=\gamma\left[\boldsymbol{\mu}(\boldsymbol{x}_t)+\sigma_t^2\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t)\right]-(\gamma-1)\boldsymbol{\mu}(\boldsymbol{x}_t)$$

Classifier-Free方案相当于直接用直接用模型拟合了$\boldsymbol{\mu}(\boldsymbol{x}_t)+\sigma_t^2\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t)$，那么类比上式，我们也可以在Classifier-Free方案中引入$w=\gamma-1$参数，用

$$\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,\boldsymbol{y},t)=(1+w)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,\boldsymbol{y},t)-w\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t) \tag{24}$$

代替$\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,\boldsymbol{y},t)$来做生成。那无条件的$\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)$怎么来呢？我们可以新引入一个特定的输入$\boldsymbol{\phi}$，它对应的目标图像为全体图像，加到了模型的训练中，这样我们就可以认为$\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)=\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,\boldsymbol{\phi},t)$了。

文章小结

本文简单介绍了建立条件扩散模型的相关理论结果，主要包含事后修改（Classifier-Guidance）和事前训练（Classifier-Free）两种方案。其中，前者不需要重新训练扩散模型，可以低成本实现简单的控制；后者需要重新训练扩散模型，成本较大，但可以实现比较精细的控制。

如果您需要引用本文，请参考：

苏剑林. (Aug. 30, 2022). 《生成扩散模型漫谈（九）：条件控制生成结果 》[Blog post]. Retrieved from https://spaces.ac.cn/archives/9257

```
@online{kexuefm-9257,
        title={生成扩散模型漫谈（九）：条件控制生成结果},
        author={苏剑林},
        year={2022},
        month={Aug},
        url={\url{https://spaces.ac.cn/archives/9257}},
}
```