21 Sep

生成扩散模型漫谈（十一）：统一扩散模型（应用篇）

By 苏剑林 | 2022-09-21 | 43224位读者 引用

在《生成扩散模型漫谈（十）：统一扩散模型（理论篇）》中，笔者自称构建了一个统一的模型框架（Unified Diffusion Model，UDM），它允许更一般的扩散方式和数据类型。那么UDM框架究竟能否实现如期目的呢？本文通过一些具体例子来演示其一般性。

框架回顾

首先，UDM通过选择噪声分布$q(\boldsymbol{\varepsilon})$和变换$\boldsymbol{\mathcal{F}}$来构建前向过程

$$\boldsymbol{x}_t = \boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon}),\quad \boldsymbol{\varepsilon}\sim q(\boldsymbol{\varepsilon}) \tag{1}$$

然后，通过如下的分解来实现反向过程$\boldsymbol{x}_{t-1}\sim p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$的采样

$$\hat{\boldsymbol{x}}_0\sim p(\boldsymbol{x}_0|\boldsymbol{x}_t)\quad \&\quad \boldsymbol{x}_{t-1}\sim p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0=\hat{\boldsymbol{x}}_0) \tag{2}$$

其中$p(\boldsymbol{x}_0|\boldsymbol{x}_t)$就是用$\boldsymbol{x}_t$预估$\boldsymbol{x}_0$的概率，一般用简单分布$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$来近似建模，训练目标基本上就是$-\log q(\boldsymbol{x}_0|\boldsymbol{x}_t)$或其简单变体。当$\boldsymbol{x}_0$是连续型数据时，$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$一般就取条件正态分布；当$\boldsymbol{x}_0$是离散型数据时，$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$可以选择自回归模型或者非自回归模型。

至于$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)$的最基准的选择就是

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) = p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)\quad\Leftrightarrow\quad \boldsymbol{x}_{t-1} = \boldsymbol{\mathcal{F}}_{t-1}(\boldsymbol{x}_0,\boldsymbol{\varepsilon}) \tag{3}$$

从这个基准出发，在不同的条件下可以得到不同的优化结果。当$\boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon})$关于$\boldsymbol{\varepsilon}$是可逆的，那么可以解出$\boldsymbol{\varepsilon}=\boldsymbol{\mathcal{F}}_t^{-1}(\boldsymbol{x}_0,\boldsymbol{x}_t)$，然后得到更好的确定性采样方式

$$\boldsymbol{x}_{t-1} = \boldsymbol{\mathcal{F}}_{t-1}(\boldsymbol{x}_0,\boldsymbol{\mathcal{F}}_t^{-1}(\boldsymbol{x}_0,\boldsymbol{x}_t)) \tag{4}$$

更进一步，如果$q(\boldsymbol{\varepsilon})$是标准正态分布，那么可以得到

$$\boldsymbol{x}_{t-1} = \boldsymbol{\mathcal{F}}_{t-1}\left(\boldsymbol{x}_0, \sqrt{1-\tilde{\sigma}_t^2}\boldsymbol{\mathcal{F}}_t^{-1}(\boldsymbol{x}_0,\boldsymbol{x}_t) + \tilde{\sigma}_t\boldsymbol{\varepsilon}\right) \tag{5}$$

热之扩散

现在这一节中，我们证明“热扩散模型”是UDM的一个特例，这里的热扩散（Hot Diffusion）指的是前面介绍的DDPM、DDIM等主流的扩散模型，这个称呼出自下面的“冷扩散”论文中。

主流扩散模型处理的是连续型数据，以加性正态噪声来构建前向过程：

$$\boldsymbol{x}_t = \bar{\alpha}_t\boldsymbol{x}_0 + \bar{\beta}_t\boldsymbol{\varepsilon},\quad \boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}) \tag{6}$$

$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$的选择就是正态分布$\mathcal{N}(\boldsymbol{x}_0;\bar{\boldsymbol{\mu}}(\boldsymbol{x}_t),\bar{\sigma}_t^2\boldsymbol{I})$，一般不将$\bar{\sigma}_t$作为训练参数，所以略去常数项后就有

$$-\log q(\boldsymbol{x}_0|\boldsymbol{x}_t) = \frac{1}{2\bar{\sigma}_t^2}\Vert\boldsymbol{x}_0 - \bar{\boldsymbol{\mu}}(\boldsymbol{x}_t)\Vert^2 \tag{7}$$

进一步引入参数化$\bar{\boldsymbol{\mu}}(\boldsymbol{x}_t) = \frac{1}{\bar{\alpha}_t}\left(\boldsymbol{x}_t - \bar{\beta}_t\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)\right)$并结合$\boldsymbol{x}_t = \bar{\alpha}_t\boldsymbol{x}_0 + \bar{\beta}_t\boldsymbol{\varepsilon}$得到

$$-\log q(\boldsymbol{x}_0|\boldsymbol{x}_t) = \frac{\bar{\beta}_t^2}{2\bar{\sigma}_t^2\bar{\alpha}_t^2}\left\Vert\boldsymbol{\varepsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t\boldsymbol{x}_0 + \bar{\beta}_t\boldsymbol{\varepsilon},t)\right\Vert^2 \tag{8}$$

实验显示略去前面的系数后效果更好，所以最终训练目标一般是$\Vert\boldsymbol{\varepsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t\boldsymbol{x}_0 + \bar{\beta}_t\boldsymbol{\varepsilon},t)\Vert^2$。至于采样过程中$\bar{\sigma}_t$的选择，可以参考《生成扩散模型漫谈（七）：最优扩散方差估计（上）》、《生成扩散模型漫谈（八）：最优扩散方差估计（下）》来进行。

最后，关于$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)$我们有

$$\begin{aligned}\boldsymbol{x}_{t-1} =&\, \bar{\alpha}_{t-1}\boldsymbol{x}_0 + \bar{\beta}_{t-1}\boldsymbol{\varepsilon} \\ \sim&\, \bar{\alpha}_{t-1}\boldsymbol{x}_0 + \sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2}\boldsymbol{\varepsilon}_1 + \sigma_t\boldsymbol{\varepsilon}_2\end{aligned},\quad \boldsymbol{\varepsilon},\boldsymbol{\varepsilon}_1,\boldsymbol{\varepsilon}_2\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}) \tag{9}$$

从$\boldsymbol{x}_t = \bar{\alpha}_t\boldsymbol{x}_0 + \bar{\beta}_t\boldsymbol{\varepsilon}$解得$\boldsymbol{\varepsilon} = (\boldsymbol{x}_t - \bar{\alpha}_t\boldsymbol{x}_0)/\bar{\beta}_t$，替换掉$\boldsymbol{\varepsilon}_1$，最终可以得到一般的$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)$为

$$\boldsymbol{x}_{t-1} = \bar{\alpha}_{t-1}\boldsymbol{x}_0 + \sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2}\frac{\boldsymbol{x}_t - \bar{\alpha}_t\boldsymbol{x}_0}{\bar{\beta}_t} + \sigma_t\boldsymbol{\varepsilon},\quad \boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}) \tag{10}$$

而$\hat{\boldsymbol{x}}_0\sim p(\boldsymbol{x}_0|\boldsymbol{x}_t)$意味着

$$\hat{\boldsymbol{x}}_0 = \bar{\boldsymbol{\mu}}(\boldsymbol{x}_t) + \bar{\sigma}_t\boldsymbol{\varepsilon} = \frac{1}{\bar{\alpha}_t}\left(\boldsymbol{x}_t - \bar{\beta}_t\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right) + \bar{\sigma}_t\boldsymbol{\varepsilon} \tag{11}$$

上面两式结合，就是最一般的主流扩散模型框架的反向过程，其中DDPM取了$\bar{\sigma}_t = 0, \sigma_t = \frac{\bar{\beta}_{t-1}\beta_t}{\bar{\beta}_t}$，DDIM则取了$\bar{\sigma}_t = 0, \sigma_t = 0$，而Analytical-DPM则重新估计了最优的非零的$\bar{\sigma}_t$。

冷之扩散

接下来，我们证明《Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise》所介绍的“冷扩散（Cold Diffusion）”也是UDM的一个特例。Cold Diffusion处理的也是连续型数据，从论文标题可以看出，它着重于使用任意（无噪声的）变换来构建前向过程，据笔者所知，这是第一篇尝试一般前向过程的论文，UDM在构建过程中，受到了它的颇多启发，在此对原作者表示感谢。

Cold Diffusion通过确定性的变换$\boldsymbol{x}_t = \boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0)$构建前向过程，为了方便后面的分析，我们引入更一般的前向过程

$$\boldsymbol{x}_t = \boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0) + \sigma\boldsymbol{\varepsilon},\quad \boldsymbol{\varepsilon}\sim q(\boldsymbol{\varepsilon}) \tag{12}$$

这里的变换$\boldsymbol{\mathcal{F}}$可以是对原始数据的任意破坏方式，对于图像来说有模糊、遮掩、池化等，如果需要确定性的变换，事后让$\sigma \to 0$即可。

接着，$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$的选择为$l_1$范数为度量的正态分布，即

$$q(\boldsymbol{x}_0|\boldsymbol{x}_t) = \frac{1}{Z(\tau)}\int e^{-\Vert\boldsymbol{x}_0 - \boldsymbol{\mathcal{G}}_t(\boldsymbol{x}_t)\Vert_1/\tau}d\boldsymbol{x}_0 \tag{13}$$

其中$Z(\tau)$是对应的归一化因子。取τ为固定值，那么除去常数项后有$-\log q(\boldsymbol{x}_0|\boldsymbol{x}_t) \propto \Vert\boldsymbol{x}_0 - \boldsymbol{\mathcal{G}}_t(\boldsymbol{x}_t)\Vert_1$，结合$\boldsymbol{x}_t = \boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0)$，得到训练目标为最小化

$$\Vert\boldsymbol{x}_0 - \boldsymbol{\mathcal{G}}_t(\boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0))\Vert_1 \tag{14}$$

在反向过程中，Cold Diffusion直接忽略了$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$的方差（即让$\tau \to 0$），这样得到$\hat{\boldsymbol{x}}_0 = \boldsymbol{\mathcal{G}}_t(\boldsymbol{x}_t)$。如果$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$直接取基准选择$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)$，即$\boldsymbol{x}_{t-1} = \boldsymbol{\mathcal{F}}_{t-1}(\boldsymbol{x}_0) + \sigma\boldsymbol{\varepsilon}$，那么代入$\hat{\boldsymbol{x}}_0$并取$\sigma \to 0$的极限后就得到

$$\hat{\boldsymbol{x}}_0 = \boldsymbol{\mathcal{G}}_t(\boldsymbol{x}_t),\quad \boldsymbol{x}_{t-1} = \boldsymbol{\mathcal{F}}_{t-1}(\hat{\boldsymbol{x}}_0) \tag{15}$$

这就是原论文的“Naive Sampling”。而如果从$\boldsymbol{x}_t = \boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0) + \sigma\boldsymbol{\varepsilon}$解出$\boldsymbol{\varepsilon} = (\boldsymbol{x}_t - \boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0))/\sigma$后，代入$\boldsymbol{x}_{t-1} = \boldsymbol{\mathcal{F}}_{t-1}(\boldsymbol{x}_0) + \sigma\boldsymbol{\varepsilon}$中就得到

$$\hat{\boldsymbol{x}}_0 = \boldsymbol{\mathcal{G}}_t(\boldsymbol{x}_t),\quad \boldsymbol{x}_{t-1} = \boldsymbol{x}_t + \boldsymbol{\mathcal{F}}_{t-1}(\hat{\boldsymbol{x}}_0) - \boldsymbol{\mathcal{F}}_t(\hat{\boldsymbol{x}}_0) \tag{16}$$

这就是原论文的“Improved Sampling”。

总的来说，Cold Diffusion首次成功实现了一般变换的前向过程的实现，但由于它过于强调“Without Noise”，所以它理论上有着无法弥补的缺陷。比如，对于$w\times w\times 3$的图片数据，Cold Diffusion在用模糊操作实现前向过程时，最终结果就相当于一个3维向量，而Cold Diffusion的反向过程也是确定性的，所以就是说Cold Diffusion通过一个确定性的变换，将$3w^2$维的图片变成了3维，然后又通过确定性的变换，将3维重建为$3w^2$维的图片，其中间过程必然有着严重的信息损失的，这必然会限制重建的清晰度，从而也限制了生成的清晰度。

要解决这个问题，就不能在前向或者反向过程中拒绝噪声的存在。因为噪声意味着不确定性，不确定性意味着“一对多”，“一对多”意味着允许“多对一”的前向过程，即允许信息损失的出现。事实上，Cold Diffusion本身就已经意识到3维的向量难以生成$3w^2$维的完整数据这个事实了，它在生成过程中，事实上还往这个3维向量加入了$3w^2$维的轻微随机噪声，实验显示这个操作提高了生成效果。而这个操作大致上就相当于$\sigma > 0$的前向过程了。

编辑模型

以上两个例子处理的都是连续型数据，而我们说过，UDM原则上不限定数据类型，这一节我们介绍一个离散型的例子，它显示基于编辑操作的文本生成模型，本质上也可以看成UDM的特例。

简单起见，我们考虑长度为l的定长句子生成，比如五言律诗、七言绝句等，变长句子不是不可以，而是细节上稍微复杂些。然后，我们将前向过程$\boldsymbol{x}_t = \boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0, \boldsymbol{\varepsilon})$定义为“随机替换”，即

> 随机选句子中的t个token随机替换为别的token

其中$t \leq l$时，当$t = l$时，此时$\boldsymbol{x}_t$就是l个完全随机组合的token。

此时$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$就是用随机替换后的序列来预测原序列的模型，用自回归/非自回归模型均可，损失函数用交叉熵。注意此时$\boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0, \boldsymbol{\varepsilon})$关于噪声必然是不可逆的（即给定$\boldsymbol{x}_0$和$\boldsymbol{x}_t$，从$\boldsymbol{x}_0$变到$\boldsymbol{x}_t$的方式不止有一种），因此我们只能用基准选择$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)$，这意味着生成过程是：

> 1、随机选l个token作为初始的$\boldsymbol{x}_l$；
>
> 2、从$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$预测$\hat{\boldsymbol{x}}_0$；
>
> 3、随机选$\hat{\boldsymbol{x}}_0$的$t-1$个token随机替换为别的token，作为$\boldsymbol{x}_{t-1}$；
>
> 4、反复执行2、3步，直到得出最终的$\boldsymbol{x}_0$。

但是，这样的算法效果不会很好，因为第2步的预估成果往往会被第3步的随机替换“毁”掉不少，有点“一夜回到解放前”的感觉，要想提高效果，就必须要用更好的采样方案，这要求$\boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon})$关于噪声可逆，也就是从给定的$\boldsymbol{x}_0$和$\boldsymbol{x}_t$可以看出变换方式是怎样的。为此，我们规定前向过程为：

> 随机选句子中的t个token随机替换为**不同**的token

它跟原来的区别是随机替换的过程中，原来的token必须替换为原来不一样的token，如果不做这个选择，则有可能采样到一样的token。做了这个限制后，我们可以直接对比$\boldsymbol{x}_0$和$\boldsymbol{x}_t$的差异，来看出修改了什么，从而将第3步的随机替换，换成由$\hat{\boldsymbol{x}}_0$到$\boldsymbol{x}_t$的替换变换：

> 1、随机选l个token作为初始的$\boldsymbol{x}_l$；
>
> 2、从$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$预测$\hat{\boldsymbol{x}}_0$，要求$\hat{\boldsymbol{x}}_0$与$\boldsymbol{x}_t$有$t$个不同token（用非自回归模型比较好实现）；
>
> 3、随机选$\boldsymbol{x}_t$中与$\hat{\boldsymbol{x}}_0$不同的token中的一个，替换为$\hat{\boldsymbol{x}}_0$对应位置的token，作为$\boldsymbol{x}_{t-1}$；
>
> 4、反复执行2、3步，直到得出最终的$\boldsymbol{x}_0$。

这样一来，每次的预测结果$\hat{\boldsymbol{x}}_0$的有效部分（$\hat{\boldsymbol{x}}_0$与$\boldsymbol{x}_t$相同的部分）都得以保留，并且$\boldsymbol{x}_{t-1}$与$\boldsymbol{x}_t$相比只修改了一个token，因此生成过程是渐进式的稳定生成。它跟普通的自回归模型区别则是去掉了从左往右的生成方向限制。

掩码模型

如果读者对上述模型还是很模糊，这里笔者再提供一个简单例子辅助理解。同样考虑长度为l的定长句子生成，我们将前向过程$\boldsymbol{x}_t=\boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon})$定义为“随机掩码”，即

> 随机选句子中的t个token随机替换为[MASK]

其中$t\leq l$时，当$t=l$时，此时$\boldsymbol{x}_t$就是l个[MASK]。

此时$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$就是用带[MASK]的序列来预测原序列的模型，用一般用类似BERT的MLM模型（非自回归模型）来实现，损失函数用交叉熵。基准的生成过程是
生成过程是：

> 1、以l个[MASK]作为初始的$\boldsymbol{x}_l$；
>
> 2、从$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$采样$\hat{\boldsymbol{x}}_0$；
>
> 3、随机选$\hat{\boldsymbol{x}}_0$的$t-1$个token随机替换为[MASK]，作为$\boldsymbol{x}_{t-1}$；
>
> 4、反复执行2、3步，直到得出最终的$\boldsymbol{x}_0$。

注意到，此时$\boldsymbol{\mathcal{F}}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon})$关于噪声是可逆的，即我们完全可以从给定的$\boldsymbol{x}_0$和$\boldsymbol{x}_t$可以看出变换方式是怎样的（即哪些token被替换为了[MASK]）。因此可以构造改进版生成过程

> 1、以l个[MASK]作为初始的$\boldsymbol{x}_l$；
>
> 2、从$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$采样$\hat{\boldsymbol{x}}_0$，注意只需采样那些原来是[MASK]的token，原来非[MASK]的不做改变；
>
> 3、从原来$\boldsymbol{x}_t$的t个[MASK]所在位置中随机选$t-1$个，将$\hat{\boldsymbol{x}}_0$的这些位置的token替换为[MASK]，作为$\boldsymbol{x}_{t-1}$；
>
> 4、反复执行2、3步，直到得出最终的$\boldsymbol{x}_0$。

当然，其实第2、3步可以合并为更直接的一步：

> 2 & 3、从$\boldsymbol{x}_t$的t个[MASK]所在位置中随机选1个，按$q(\boldsymbol{x}_0|\boldsymbol{x}_t)$对应位置的概率采样一个token替换上去，作为$\boldsymbol{x}_{t-1}$；

这跟基于MLM模型的Gibbs采样几乎一致了（参考《【搜出来的文本】·（三）基于BERT的文本采样》）。从“编辑模型”和“掩码模型”两个例子我们应该可以大致体会到，很多

“渐变式生成”的模型，都可以用UDM框架来重新表述。又或者反过来，我们能想到的任何渐变式生成方式，都可以尝试用UDM框架来构建其概率表述。

编码模型

前面我们所讨论的前向过程都是无训练参数的，也就是说都是事先设计好的流程，但这其实也并不是必要的。我们可以将DDPM的扩散过程一般化为

$$\boldsymbol{x}_t = \bar{\alpha}_t \boldsymbol{\mathcal{F}}(\boldsymbol{x}_0) + \bar{\beta}_t \boldsymbol{\varepsilon},\quad \boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{17}$$

其中$\boldsymbol{\mathcal{F}}(\boldsymbol{x}_0)$是对$\boldsymbol{x}_0$的编码模型，可以带训练参数。此时训练目标就是

$$-\log q(\boldsymbol{x}_0|\boldsymbol{x}_t) = -\log q(\boldsymbol{x}_0|\bar{\alpha}_t \boldsymbol{\mathcal{F}}(\boldsymbol{x}_0) + \bar{\beta}_t \boldsymbol{\varepsilon}) \tag{18}$$

只不过此时$\boldsymbol{\mathcal{F}}$也有训练参数。至于反向过程也是类似的，只不过最后采样到$\hat{\boldsymbol{x}}_0\sim q(\boldsymbol{x}_0|\boldsymbol{x}_1)$就直接返回$\hat{\boldsymbol{x}}_0$了。特别地，由于多了一个编码模型$\boldsymbol{\mathcal{F}}$，所以输入$\boldsymbol{x}_0$既可以是离散型数据，也可以是连续型数据，它提供了类似VAE的将数据分布编码到隐变量的正态分布的一种方法。

文章小结

本文主要应用上一篇文章所构建的统一扩散模型框架（Unified Diffusion Model，UDM）来推导几个具体的例子，包括主流的扩散模型、Cold Diffusion、文本编辑生成、编码模型等。

如果您需要引用本文，请参考：

苏剑林. (Sep. 21, 2022). 《生成扩散模型漫谈（十一）：统一扩散模型（应用篇）》[Blog post]. Retrieved from https://spaces.ac.cn/archives/9271

```
@online{kexuefm-9271,
    title={生成扩散模型漫谈（十一）：统一扩散模型（应用篇）},
    author={苏剑林},
    year={2022},
    month={Sep},
    url={\url{https://spaces.ac.cn/archives/9271}},
}
```