22 Nov 生成扩散模型漫谈（二十六）：基于恒等式的蒸馏（下）

By 苏剑林 | 2024-11-22 | 6243位读者 引用

继续回到我们的扩散系列。在《生成扩散模型漫谈（二十五）：基于恒等式的蒸馏（上）》中，我们介绍了SiD（Score identity Distillation），这是一种不需要真实数据、也不需要从教师模型采样的扩散模型蒸馏方案，其形式类似GAN，但有着比GAN更好的训练稳定性。

SiD的核心是通过恒等变换来为学生模型构建更好的损失函数，这一点是开创性的，同时也遗留了一些问题。比如，SiD对损失函数的恒等变换是不完全的，如果完全变换会如何？如何从理论上解释SiD引入的λ的必要性？上个月放出的《Flow Generator Matching》（简称FGM）成功从更本质的梯度角度解释了$\lambda = 0.5$的选择，而受到FGM启发，笔者则进一步发现了$\lambda = 1$的一种解释。

接下来我们将详细介绍SiD的上述理论进展。

思想回顾

根据上一篇文章的介绍，我们知道SiD实现蒸馏的思想是“相近的分布，它们训练出来的去噪模型也是相近的”，用公式表示就是

$$\text{教师扩散模型：}\quad \boldsymbol{\varphi}^* = \mathop{\text{argmin}}_{\boldsymbol{\varphi}} \mathbb{E}_{\boldsymbol{x}_0\sim\tilde{p}(\boldsymbol{x}_0),\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\left\Vert\boldsymbol{\epsilon}_{\boldsymbol{\varphi}}(\boldsymbol{x}_t,t) - \boldsymbol{\varepsilon}\right\Vert^2\right] \tag{1}$$

$$\text{学生扩散模型：}\quad \boldsymbol{\psi}^* = \mathop{\text{argmin}}_{\boldsymbol{\psi}} \mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\left\Vert\boldsymbol{\epsilon}_{\boldsymbol{\psi}}(\boldsymbol{x}_t^{(g)},t) - \boldsymbol{\varepsilon}\right\Vert^2\right] \tag{2}$$

$$\text{学生生成模型：}\quad \boldsymbol{\theta}^* = \mathop{\text{argmin}}_{\boldsymbol{\theta}} \underbrace{\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\left\Vert\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t^{(g)},t) - \boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t)\right\Vert^2\right]}_{\mathcal{L}_1} \tag{3}$$

这里记号比较多，我们逐一解释。第一个损失函数就是我们要蒸馏的扩散模型的训练目标，其中$\boldsymbol{x}_t = \bar{\alpha}_t\boldsymbol{x}_0 + \bar{\beta}_t\boldsymbol{\varepsilon}$代表加噪样本，$\bar{\alpha}_t, \bar{\beta}_t$是noise schedule，$\boldsymbol{x}_0$是训练样本；第二个损失函数则是用学生模型生成的数据来训练的扩散模型，其中

$\boldsymbol{x}_t^{(g)} = \bar{\alpha}_t \boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z}) + \bar{\beta}_t \boldsymbol{\varepsilon}$，这里的$\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z})$代表学生模型的生成样本，也记为$\boldsymbol{x}_0^{(g)}$；第三个损失函数，则是试图通过拉近真实数据和学生数据所训练的扩散模型的差距，来训练学生生成模型（生成器）。

这里的教师模型是可以提前训练好的，而两个学生模型的训练只需要教师模型本身，并不需要用到训练教师模型的数据，所以作为一种蒸馏方式来看SiD是data-free的；两个学生模型则是类似GAN那样的交替训练，逐步提高生成器的生成质量。就笔者所阅读过的文献来看，这种训练思想最早出自论文《Learning Generative Models using Denoising Density Estimators》，我们在《从去噪自编码器到生成模型》也有过相关介绍。

然而，尽管看上去没什么毛病，但实际情况是式(2)和式(3)的交替训练非常容易崩溃，以至于几乎不能出效果。这是因为理论和实践上的两个gap：

> 1、理论上要求先求出式(2)的最优解，然后才去优化式(3)，但实际上从训练成本考虑，我们并没有将它训练到最优就去优化式(3)了；
>
> 2、理论上$\boldsymbol{\psi}^*$随$\boldsymbol{\theta}$而变，即应该写成$\boldsymbol{\psi}^*(\boldsymbol{\theta})$，从而在优化式(3)时应该多出一项$\boldsymbol{\psi}^*(\boldsymbol{\theta})$对$\boldsymbol{\theta}$的梯度，但实际上在优化式(3)时我们都只当$\boldsymbol{\psi}^*$是常数。

第1个问题其实还好，因为随着训练的推进$\boldsymbol{\psi}$总能慢慢逼近理论最优的$\boldsymbol{\psi}^*$，但第2个问题非常困难且本质，可以说GAN的训练不稳定性同样也有这个问题的“功劳”。而SiD和FGM的核心贡献，正是试图解决第2个问题。

恒等变换

SiD的想法是通过恒等变换来减少生成器损失函数(3)对$\boldsymbol{\psi}^*$的依赖，从而弱化第2个问题。这一想法确实是开创性的，后面已经有不少工作围绕着SiD展开，包括下面要介绍的FGM也算是其中之一。

恒等变换的核心，是如下恒等式：

$$\mathbb{E}_{\boldsymbol{x}_0\sim\tilde{p}(\boldsymbol{x}_0),\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\langle\boldsymbol{f}(\boldsymbol{x}_t,t),\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t,t)\rangle\right] = \mathbb{E}_{\boldsymbol{x}_0\sim\tilde{p}(\boldsymbol{x}_0),\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\langle\boldsymbol{f}(\boldsymbol{x}_t,t),\boldsymbol{\varepsilon}\rangle\right] \tag{4}$$

简单来说就是$\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t,t)$可以替换成$\boldsymbol{\varepsilon}$。这里的$\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t,t)$是式$(1)$的理论最优解，而$\boldsymbol{f}(\boldsymbol{x}_t,t)$是任意只依赖于$\boldsymbol{x}_t$和$t$的向量函数。注意“只依赖于$\boldsymbol{x}_t$和$t$”是恒等式成立的必要条件，一旦$\boldsymbol{f}$掺杂了独立的$\boldsymbol{x}_0$或$\boldsymbol{\varepsilon}$，那么恒等式就未必成立了，所以应用该恒等式之前需要仔细检查这一点。

上一篇文章我们已经给出了该恒等式的证明，不过现在看来那个证明显得有点迂回，这里给出一个更直接点的证明：

证明： 将目标(1)等价地改写成

$$\boldsymbol{\varphi}^* = \mathop{\text{argmin}}_{\boldsymbol{\varphi}}\mathbb{E}_{\boldsymbol{x}_t\sim p(\boldsymbol{x}_t)}\left[\mathbb{E}_{\boldsymbol{\varepsilon}\sim p(\boldsymbol{\varepsilon}|\boldsymbol{x}_t)}\left[\Vert\boldsymbol{\epsilon}_{\boldsymbol{\varphi}}(\boldsymbol{x}_t,t) - \boldsymbol{\varepsilon}\Vert^2\right]\right] \tag{5}$$

根据$\mathbb{E}[\boldsymbol{x}] = \mathop{\text{argmin}}_{\boldsymbol{\mu}}\mathbb{E}_{\boldsymbol{x}}\left[\Vert\boldsymbol{\mu} - \boldsymbol{x}\Vert^2\right]$（不熟悉可以求导证一下），我们可以得出上式的理论最优解是

$$\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t,t) = \mathbb{E}_{\boldsymbol{\varepsilon}\sim p(\boldsymbol{\varepsilon}|\boldsymbol{x}_t)}[\boldsymbol{\varepsilon}] \tag{6}$$

所以

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}_0\sim\tilde{p}(\boldsymbol{x}_0),\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\langle\boldsymbol{f}(\boldsymbol{x}_t,t),\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t,t)\rangle\right] =&\, \mathbb{E}_{\boldsymbol{x}_t\sim p(\boldsymbol{x}_t)}\left[\langle\boldsymbol{f}(\boldsymbol{x}_t,t),\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t,t)\rangle\right] \\
=&\, \mathbb{E}_{\boldsymbol{x}_t\sim p(\boldsymbol{x}_t)}\left[\left\langle\boldsymbol{f}(\boldsymbol{x}_t,t),\mathbb{E}_{\boldsymbol{\varepsilon}\sim p(\boldsymbol{\varepsilon}|\boldsymbol{x}_t)}[\boldsymbol{\varepsilon}]\right\rangle\right] \\
=&\, \mathbb{E}_{\boldsymbol{x}_t\sim p(\boldsymbol{x}_t),\boldsymbol{\varepsilon}\sim p(\boldsymbol{\varepsilon}|\boldsymbol{x}_t)}\left[\langle\boldsymbol{f}(\boldsymbol{x}_t,t),\boldsymbol{\varepsilon}\rangle\right] \\
=&\, \mathbb{E}_{\boldsymbol{x}_0\sim\tilde{p}(\boldsymbol{x}_0),\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\langle\boldsymbol{f}(\boldsymbol{x}_t,t),\boldsymbol{\varepsilon}\rangle\right]
\end{aligned} \quad ($$

证毕。证明过程的“必经之路”是第一个等号，这需要用到“$\boldsymbol{f}(\boldsymbol{x}_t,t)$只依赖于$\boldsymbol{x}_t$和$t$”这个条件。

恒等式(4)的关键是$\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t,t)$的最优性，而目标$(1)$和$(2)$形式是一样的，所以同样的结论也适用于$\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t,t)$，利用它我们就可以将$(3)$变换成

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\left\Vert\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t^{(g)},t) - \boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t)\right\Vert^2\right] \\
=&\,\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\left\langle\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t^{(g)},t) - \boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t), \boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t^{(g)},t) - \underbrace{\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t)}_{\text{可以替换为}\boldsymbol{\varepsilon}}\right\rangle\right] \\
=&\,\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\left\langle\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t^{(g)},t) - \boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t), \boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t^{(g)},t) - \boldsymbol{\varepsilon}\right\rangle\right] \triangleq \mathcal{L}_2
\end{aligned} \tag{8}$$

最后的形式就是SiD所提的生成器损失函数$\mathcal{L}_2$，它是SiD成功训练的关键，我们可以理解为它通过恒等变换提前预估了$\boldsymbol{\psi}^*$的值，同时弱化了对$\boldsymbol{\psi}^*$的依赖，从而以它为损失函数训练生成器比$\mathcal{L}_1$有着更好的效果。

SiD的遗留问题是：

> 1、$\mathcal{L}_2$的恒等变换并不彻底，将$\mathcal{L}_2$展开会发现里边还有一项$\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t^{(g)},t), \boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t)\rangle]$，这一项的$\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t)$同样可以替换为$\boldsymbol{\varepsilon}$，那么问题就是完整的变换即下式会是一个比$\mathcal{L}_2$更好的选择吗？
>
> $$\mathcal{L}_3 = \mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\Vert\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}\Vert^2 - 2\langle\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle + \langle\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t)\boldsymbol{\varepsilon}\rangle\right] \tag{9}$$
>
> 2、实际上SiD最终用的损失不是$\mathcal{L}_2$也不是$\mathcal{L}_1$，而是$\mathcal{L}_2 - \lambda\mathcal{L}_1$，其中$\lambda > 0$，并且实验发现$\lambda$的最优值在$1$附近，某些任务甚至在$\lambda = 1.2$表现最好，这是非常让人困惑的，因为$\mathcal{L}_1,\mathcal{L}_2$是理论上相等的，所以$\lambda > 1$似乎在反向优化$\mathcal{L}_1$？这不就跟出发点相反了？显然这迫切需要一个理论解释。

直面梯度

再来回顾一下，我们面临的根本困难是：理论上$\boldsymbol{\psi}^*$是$\boldsymbol{\theta}$的函数，所以我们在求$\nabla_{\boldsymbol{\theta}}\mathcal{L}_1$或$\nabla_{\boldsymbol{\theta}}\mathcal{L}_2$时，需要想办法求$\nabla_{\boldsymbol{\theta}}\boldsymbol{\psi}^*$，但实践中我们至多可以得到$\mathcal{L}_i^{(\text{sg})} \triangleq \mathcal{L}_i|_{\boldsymbol{\psi}^*\to\text{sg}[\boldsymbol{\psi}^*]}$，其中sg是stop gradient的意思，即无法获取$\boldsymbol{\psi}^*$关于$\boldsymbol{\theta}$的梯度，所以不论$\mathcal{L}_1,\mathcal{L}_2,\mathcal{L}_3$，它们在实践中的梯度都是有偏的。

这时候就轮到FGM登场了，它的想法更贴近本质：损失$\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$都只关注到了损失层面的相等性，但对于优化器来说我们需要的是梯度层面的相等，所以我们需要想办法找一个新的损失函数$\mathcal{L}_4$，使得它满足

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_4(\boldsymbol{\theta}, \text{sg}[\boldsymbol{\psi}^*]) = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{1/2/3}(\boldsymbol{\theta}, \boldsymbol{\psi}^*) \tag{10}$$

即$\nabla_{\boldsymbol{\theta}}\mathcal{L}_4^{(\text{sg})} = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{1/2/3}$，那么以$\mathcal{L}_4$为损失函数时，就可以实现无偏的优化效果了。

FGM的推导同样基于恒等式(4)，不过它的原始推导有点繁琐，对于本文来说可以直接从$\mathcal{L}_3$即式(9)出发，它跟$\boldsymbol{\psi}^*$相关的项就只剩下$\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t), \boldsymbol{\varepsilon}\rangle]$，我们直接把它的梯度算出来，方法将“先恒等变换后求梯度”和“先求梯度后恒等变换”分别应用于$\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\Vert\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t)\Vert^2]$操作一遍，对比它们的结果。

先恒等变换后求梯度：

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\Vert\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t)\Vert^2] \\
=&\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t), \boldsymbol{\varepsilon}\rangle] = \mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t), \boldsymbol{\varepsilon}\rangle] \\
=&\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)}, t), \boldsymbol{\varepsilon}\rangle] + \mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\text{sg}[\boldsymbol{x}_t^{(g)}], t), \boldsymbol{\varepsilon}\rangle]
\end{aligned} \tag{11}$$

先求梯度后恒等变换：

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\Vert\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t)\Vert^2] \\
=&\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\nabla_{\boldsymbol{\theta}}\Vert\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t)\Vert^2] = 2\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t), \boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t)\rangle] \\
=&2\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)}, t), \boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t)\rangle] + \underbrace{2\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\text{sg}[\boldsymbol{x}_t^{(g)}], t), \boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t}_{\text{可以应用式}(4)} \\
=&2\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)}, t), \boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t)\rangle] + 2\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\text{sg}[\boldsymbol{x}_t^{(g)}], t), \boldsymbol{\varepsilon}\rangle]
\end{aligned}$$

这里要注意第三个等号，只有$\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\text{sg}[\boldsymbol{x}_t^{(g)}], t)$这一项才可以应用恒等式(4)，因为$\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)}, t)$的$\boldsymbol{x}_t^{(g)}$要对$\boldsymbol{\theta}$求梯度，求完梯度后就不一定是$\boldsymbol{x}_t^{(g)}$的函数了，所以不满足应用式(4)的条件。

现在对于$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\Vert\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t)\Vert^2]$我们有两个结果，将式$(11)$乘以2然后减去式$(12)$得到

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle] = \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\Vert\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t)\Vert^2] = (11)\times 2 - \\
=&\, 2\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle] - 2\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)} \\
=&\, 2\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle] - \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\Vert\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)},t)\Vert^2] \\
=&\, \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[2\langle\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle - \Vert\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)},t)\Vert^2]
\end{aligned}$$

留意最后被求梯度的式子，它所有的$\boldsymbol{\psi}^*$都被加上了sg，说明我们不用设法求它关于$\boldsymbol{\theta}$的梯度了，但它的梯度等于$\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle]$的准确梯度，所以用它来替换掉$\mathcal{L}_3$的对应项，我们就得到了$\mathcal{L}_4$：

$$\mathcal{L}_4^{(\text{sg})} = \mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\Big[\Vert\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}\Vert^2 - 2\langle\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle + 2\langle\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle - \Vert\boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)}$$

这就是FGM的最终结果，它只依赖于$\text{sg}[\boldsymbol{\psi}^*]$，但成立$\nabla_{\boldsymbol{\theta}}\mathcal{L}_4^{(\text{sg})} = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{1/2/3}$。再仔细观察一下，就会发现成立$\mathcal{L}_4^{(\text{sg})} = 2\mathcal{L}_2^{(\text{sg})} - \mathcal{L}_1^{(\text{sg})} = 2(\mathcal{L}_2^{(\text{sg})} - 0.5\times\mathcal{L}_1^{(\text{sg})})$，所以FGM相当于从梯度角度肯定了SiD的$\lambda = 0.5$的选择。

顺便说一下，FGM原论文的描述是在ODE式扩散框架（flow matching）内进行的，但正如笔者在上一篇文章所说，不管是SiD还是FGM，它实际并没有用到扩散模型的迭代生成过程，而是只用到了扩散模型所训练的去噪模型，所以不管是ODE、SDE还是DDPM框架都只是表象，它的去噪模型才是本质，所以本文可以接着上一篇SiD的记号来介绍FGM。

广义散度

FGM已经成功地求出了最本质的梯度，但这只能解释SiD的$\lambda = 0.5$，这意味着如果我们需要解释其他λ值的可行性，就必须修改出发点了。为此，我们回到原点，反思一下生成器的目标(3)。

熟悉扩散模型的读者应该都知道，式(1)的理论最优解还可以写成$\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t, t) = -\bar{\beta}_t\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t)$，同理式(2)的最优解则是$\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t) = -\bar{\beta}_t\nabla_{\boldsymbol{x}_t^{(g)}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)})$，这里的$p(\boldsymbol{x}_t)$、$p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)})$分别是真实数据、生成器数据加噪的分布，如果不了解这个结果，可以参考《生成扩散模型漫谈（五）：一般框架之SDE篇》、《生成扩散模型漫谈（十八）：得分匹配 = 条件得分匹配》等介绍。

将这两个理论最优解代回式(3)，我们会发现生成器实际上在试图最小化Fisher散度：

$$\begin{aligned}\mathcal{F}(p, p_{\boldsymbol{\theta}}) =&\, \mathbb{E}_{\boldsymbol{z},\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\Vert\nabla_{\boldsymbol{x}_t^{(g)}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)}) - \nabla_{\boldsymbol{x}_t^{(g)}}\log p(\boldsymbol{x}_t^{(g)})\Vert^2\right] \\ =&\, \int p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)})\left\Vert\nabla_{\boldsymbol{x}_t^{(g)}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)}) - \nabla_{\boldsymbol{x}_t^{(g)}}\log p(\boldsymbol{x}_t^{(g)})\right\Vert^2 d\boldsymbol{x}_t^{(g)}\end{aligned}\tag{15}$$

我们要反思的事情，就是Fisher散度的合理性和改进点。可以看到，Fisher散度里边$p_{\boldsymbol{\theta}}$出现了两次，现在我们来请读者思考一个问题：**这两处$p_{\boldsymbol{\theta}}$中哪一处更重要呢?**

答案是**第二处**。为了理解这个事实，我们不妨考虑两种情况：<u>1、固定第一处$p_{\boldsymbol{\theta}}$，只优化第二处$p_{\boldsymbol{\theta}}$；2、固定第二处$p_{\boldsymbol{\theta}}$，只优化第一处$p_{\boldsymbol{\theta}}$。</u>它们的结果有什么区别呢？第一种情况大概率不会有什么变化，即依然能学到$p_{\boldsymbol{\theta}} = p$，事实上由于Fisher散度带有$\Vert\Vert^2$，所以下面更一般的结论几乎是显然成立的：

> 只要$r(\boldsymbol{x})$是一个处处不为零的分布，那么$p(\boldsymbol{x}) = q(\boldsymbol{x})$依然是如下广义Fisher散度的理论最优解：
>
> $$\mathcal{F}(p, q|r) = \int r(\boldsymbol{x})\Vert\nabla_{\boldsymbol{x}} p(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} q(\boldsymbol{x})\Vert^2 d\boldsymbol{x}\tag{16}$$

说简单点，就是第一处$p_{\boldsymbol{\theta}}$根本不重要，换成其他分布都行，单靠$\Vert\Vert^2$就能保证两个分布相等。但第二种情况就不一样了，固定第二处$p_{\boldsymbol{\theta}}$只优化第一处$p_{\boldsymbol{\theta}}$的理论最优解是

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)}) = \delta(\boldsymbol{x}_t^{(g)} - \boldsymbol{x}_t^*),\quad \boldsymbol{x}_t^* = \mathop{\text{argmin}}_{\boldsymbol{x}_t^{(g)}}\left\Vert\nabla_{\boldsymbol{x}_t^{(g)}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)}) - \nabla_{\boldsymbol{x}_t^{(g)}}\log p(\boldsymbol{x}_t^{(g)})\right\Vert$$

其中δ是狄拉克delta分布，即模型只需要生成让$\Vert\Vert^2$最小的那个样本，就可以让损失最

小，这说白了就是模式坍缩（Mode Collapse）！所以，Fisher散度中的第一处$p_{\boldsymbol{\theta}}$的作用不单单是次要的，甚至还可能是负面的。

这启发我们，当我们使用基于梯度的优化器来训练模型时，第一处$p_{\boldsymbol{\theta}}$的梯度干脆不要还会更好，即下述形式的Fisher散度是一个更好的选择

$$\begin{aligned}\mathcal{F}^{+}(p, p_{\boldsymbol{\theta}}) =&\, \int p_{\text{sg}[\boldsymbol{\theta}]}(\boldsymbol{x}_t^{(g)})\left\Vert\nabla_{\boldsymbol{x}_t^{(g)}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)}) - \nabla_{\boldsymbol{x}_t^{(g)}}\log p(\boldsymbol{x}_t^{(g)})\right\Vert^2 d\boldsymbol{x}_t^{(g)} \\ =&\, \mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\Vert\nabla_{\boldsymbol{x}_t^{(g)}}\log p_{\boldsymbol{\theta}}(\text{sg}[\boldsymbol{x}_t^{(g)}]) - \nabla_{\boldsymbol{x}_t^{(g)}}\log p(\text{sg}[\boldsymbol{x}_t^{(g)}])\Vert^2\right] \\ \propto&\, \underbrace{\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\Vert\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\text{sg}[\boldsymbol{x}_t^{(g)}],t) - \boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\text{sg}[\boldsymbol{x}_t^{(g)}],t)\Vert^2\right]}_{\mathcal{L}_5}\end{aligned} \quad (18)$$

也就是说，这里的$\mathcal{L}_5$极有可能会是一个比$\mathcal{L}_1$更好的出发点，它数值上跟$\mathcal{L}_1$是相等的，但少了一部分梯度：

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_5 = \nabla_{\boldsymbol{\theta}}\mathcal{L}_1 - \nabla_{\boldsymbol{\theta}}\underbrace{\mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\Vert\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t^{(g)},t) - \boldsymbol{\epsilon}_{\text{sg}[\boldsymbol{\psi}^*]}(\boldsymbol{x}_t^{(g)},t)\Vert^2\right]}_{\text{刚好是}\mathcal{L}_1^{(\text{sg})}} \quad (19)$$

其中$\nabla_{\boldsymbol{\theta}}\mathcal{L}_1$已经由FGM算出来了，它等于$\nabla_{\boldsymbol{\theta}}(2\mathcal{L}_2^{(\text{sg})} - \mathcal{L}_1^{(\text{sg})})$，因此以$\mathcal{L}_5$为出发点，我们实践中的损失函数是$2\mathcal{L}_2^{(\text{sg})} - \mathcal{L}_1^{(\text{sg})} - \mathcal{L}_1^{(\text{sg})} = 2(\mathcal{L}_2^{(\text{sg})} - \mathcal{L}_1^{(\text{sg})})$，这就解释了$\lambda=1$的选择。至于$\lambda$稍大于1的选择，则更为极端一些，它相当于在$\mathcal{L}_5$的基础上将$-\mathcal{L}_1^{(\text{sg})}$作为额外的惩罚项，进一步降低模式坍缩的风险，当然这里真就是单纯的惩罚项，所以权重就不能太大了，根据SiD的实验结果，$\lambda=1.5$的时候已经开始训崩了。

顺便说一下，FGM之前作者还有个作品《One-Step Diffusion Distillation through Score Implicit Matching》，里边也提出了类似的对第一处$p_{\boldsymbol{\theta}}$改为$p_{\text{sg}[\boldsymbol{\theta}]}$的做法，但没有明确地从Fisher散度的原始形式出发讨论该操作的合理性，稍欠完整。

文章小结

本文介绍了SiD（Score identity Distillation）的后续理论进展，主要内容是从梯度视角解释了SiD中的λ参数设置，核心部分是由FGM（Flow Generator Matching）发现的准

确估计SiD梯度的巧妙思路，这肯定了$\lambda=0.5$的选择，在此基础上，笔者拓展了Fisher散度的概念，从而解释了$\lambda=1$的取值。