Generative Diffusion Models Talk (26): Identity-Based Distillation (Part 2)

Jianlin Su

November 22, 2024

Continuing our series on diffusion models. In "Generative Diffusion Models Talk (25): Identity-Based Distillation (Part 1)", we introduced SiD (Score identity Distillation), a distillation scheme for diffusion models that requires neither real data nor sampling from a teacher model. Its form is similar to a GAN, but it possesses better training stability.

The core of SiD is to construct a better loss function for the student model through identity transformations. This point is pioneering, yet it leaves some questions. For instance, SiD's identity transformation of the loss function is incomplete; what would happen if it were fully transformed? How can we theoretically explain the necessity of the λ introduced by SiD? The paper "Flow Generator Matching" (FGM), released last month, successfully explained the choice of $\lambda = 0.5$ from a more fundamental gradient perspective. Inspired by FGM, I have further discovered an explanation for $\lambda = 1$.

Next, we will detail these theoretical advancements in SiD.

1 Review of Ideas

According to the previous article, we know that the idea behind SiD's distillation is that "similar distributions will result in similar denoising models trained on them." Expressed in formulas:

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

$$\text{Student Diffusion Model:}\quad \boldsymbol{\psi}^* = \mathop{\text{argmin}}_{\boldsymbol{\psi}} \mathbb{E}_{\boldsymbol{z},\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\left\|\boldsymbol{\epsilon}_{\boldsymbol{\psi}}(\boldsymbol{x}_t^{(g)},t)-\boldsymbol{\varepsilon}\right\|^2\right] \tag{2}$$

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

There are many notations here, so let's explain them one by one. The first loss function is the training objective of the diffusion model we want to distill, where $\boldsymbol{x}_t = \bar{\alpha}_t\boldsymbol{x}_0 + \bar{\beta}_t\boldsymbol{\varepsilon}$ represents the noisy sample, $\bar{\alpha}_t, \bar{\beta}_t$ are the noise schedule, and $\boldsymbol{x}_0$ is the training sample. The second loss function is a diffusion model trained using data generated by the student model, where $\boldsymbol{x}_t^{(g)} = \bar{\alpha}_t\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z}) + \bar{\beta}_t\boldsymbol{\varepsilon}$, and $\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z})$ represents the sample generated by the student model, also denoted as $\boldsymbol{x}_0^{(g)}$. The third loss function attempts to train the student generative model (generator) by narrowing the gap between the diffusion models trained on real data and student data.

The teacher model can be pre-trained, and the training of the two student models only requires the teacher model itself, without needing the data used to train the teacher. Thus, as a distillation method, SiD is data-free. The two student models are trained alternately, similar to a GAN, to gradually improve the generation quality of the generator. Based on the literature I have read, this training idea first appeared in the paper "Learning Generative Models using

Denoising Density Estimators", and we also covered it in "From Denoising Autoencoders to Generative Models".

However, although it looks sound, in practice, the alternating training of Eq. (2) and Eq. (3) is very prone to collapse, to the point where it almost yields no results. This is due to two gaps between theory and practice:

1. Theoretically, one should first find the optimal solution for Eq. (2) before optimizing Eq. (3). In practice, due to training costs, we optimize Eq. (3) before it reaches the optimum.

2. Theoretically, $\boldsymbol{\psi}^*$ changes with $\boldsymbol{\theta}$, meaning it should be written as $\boldsymbol{\psi}^*(\boldsymbol{\theta})$. Thus, when optimizing Eq. (3), there should be an additional term for the gradient of $\boldsymbol{\psi}^*(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. In practice, we treat $\boldsymbol{\psi}^*$ as a constant when optimizing Eq. (3).

The first problem is manageable because as training progresses, $\boldsymbol{\psi}$ can gradually approach the theoretical optimum $\boldsymbol{\psi}^*$. However, the second problem is very difficult and fundamental; it can be said that the training instability of GANs also owes much to this issue. The core contribution of SiD and FGM is precisely the attempt to solve this second problem.

2 Identity Transformation

SiD's idea is to reduce the dependence of the generator loss function (3) on $\boldsymbol{\psi}^*$ through identity transformations, thereby weakening the second problem. This idea is indeed pioneering, and many subsequent works have been built around SiD, including FGM, which we will introduce below.

The core of the identity transformation is the following identity:

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

Simply put, $\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t,t)$ can be replaced by $\boldsymbol{\varepsilon}$. Here $\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t,t)$ is the theoretical optimal solution to Eq. (1), and $\boldsymbol{f}(\boldsymbol{x}_t,t)$ is any vector function that depends only on $\boldsymbol{x}_t$ and t. Note that "depending only on $\boldsymbol{x}_t$ and t" is a necessary condition for the identity to hold. Once $\boldsymbol{f}$ is mixed with independent $\boldsymbol{x}_0$ or $\boldsymbol{\varepsilon}$, the identity may no longer hold. Therefore, one must carefully check this before applying the identity.

We provided a proof of this identity in the previous article, but in hindsight, that proof seemed a bit roundabout. Here is a more direct proof:

Proof: Rewrite the objective (1) equivalently as:

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

According to $\mathbb{E}[\boldsymbol{x}] = \mathop{\text{argmin}}_{\boldsymbol{\mu}} \mathbb{E}_{\boldsymbol{x}}\left[\|\boldsymbol{\mu} - \boldsymbol{x}\|^2\right]$ (if unfamiliar, one can prove this by taking the derivative), we can conclude that the theoretical optimal solution to the above equation is:

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

Therefore:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}_0\sim\tilde{p}(\boldsymbol{x}_0),\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\langle \boldsymbol{f}(\boldsymbol{x}_t,t), \boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t,t)\rangle\right] &= \mathbb{E}_{\boldsymbol{x}_t\sim p(\boldsymbol{x}_t)}\left[\langle \boldsymbol{f}(\boldsymbol{x}_t,t), \boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t,t)\rangle\right] \\
&= \mathbb{E}_{\boldsymbol{x}_t\sim p(\boldsymbol{x}_t)}\left[\left\langle \boldsymbol{f}(\boldsymbol{x}_t,t), \mathbb{E}_{\boldsymbol{\varepsilon}\sim p(\boldsymbol{\varepsilon}|\boldsymbol{x}_t)}[\boldsymbol{\varepsilon}]\right\rangle\right] \\
&= \mathbb{E}_{\boldsymbol{x}_t\sim p(\boldsymbol{x}_t),\boldsymbol{\varepsilon}\sim p(\boldsymbol{\varepsilon}|\boldsymbol{x}_t)}\left[\langle \boldsymbol{f}(\boldsymbol{x}_t,t), \boldsymbol{\varepsilon}\rangle\right] \\
&= \mathbb{E}_{\boldsymbol{x}_0\sim\tilde{p}(\boldsymbol{x}_0),\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\langle \boldsymbol{f}(\boldsymbol{x}_t,t), \boldsymbol{\varepsilon}\rangle\right]
\end{aligned} \tag{7}$$

Q.E.D. The "essential path" of the proof is the first equality, which requires the condition that "$\boldsymbol{f}(\boldsymbol{x}_t, t)$ depends only on $\boldsymbol{x}_t$ and t."

The key to identity (4) is the optimality of $\boldsymbol{\epsilon}_{\varphi^*}(\boldsymbol{x}_t, t)$. Since the forms of objectives (1) and (2) are identical, the same conclusion applies to $\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t, t)$. Using this, we can transform (3) into:

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

The final form is the generator loss function $\mathcal{L}_2$ proposed by SiD. It is the key to SiD's successful training. We can understand it as pre-estimating the value of $\boldsymbol{\psi}^*$ through identity transformation while weakening the dependence on $\boldsymbol{\psi}^*$. Thus, using it as the loss function to train the generator yields better results than $\mathcal{L}_1$.

The remaining issues with SiD are:

1. The identity transformation of $\mathcal{L}_2$ is not thorough. Expanding $\mathcal{L}_2$ reveals a term $\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle \boldsymbol{\epsilon}_{\varphi^*}(\boldsymbol{x}_t^{(g)}, t), \boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)}, t)\rangle]$, where $\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)}, t)$ can also be replaced by $\boldsymbol{\varepsilon}$. The question then is: would the complete transformation, i.e., the following equation, be a better choice than $\mathcal{L}_2$?

$$\mathcal{L}_3 = \mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\|\boldsymbol{\epsilon}_{\varphi^*}\|^2 - 2\langle \boldsymbol{\epsilon}_{\varphi^*}(\boldsymbol{x}_t^{(g)}, t), \boldsymbol{\varepsilon}\rangle + \langle \boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)}, t), \boldsymbol{\varepsilon}\rangle\right] \tag{9}$$

2. In practice, SiD ultimately uses the loss $\mathcal{L}_2 - \lambda\mathcal{L}_1$ instead of $\mathcal{L}_2$ or $\mathcal{L}_1$, where $\lambda > 0$. Experiments found that the optimal value of λ is around 1, and for some tasks, $\lambda = 1.2$ even performs best. This is very confusing because $\mathcal{L}_1$ and $\mathcal{L}_2$ are theoretically equal, so $\lambda > 1$ seems to be optimizing $\mathcal{L}_1$ in reverse? Doesn't this contradict the starting point? Clearly, this urgently needs a theoretical explanation.

3 Facing the Gradient

To recap, the fundamental difficulty we face is: theoretically, $\boldsymbol{\psi}^*$ is a function of $\boldsymbol{\theta}$. Therefore, when calculating $\nabla_{\boldsymbol{\theta}}\mathcal{L}_1$ or $\nabla_{\boldsymbol{\theta}}\mathcal{L}_2$, we need a way to find $\nabla_{\boldsymbol{\theta}}\boldsymbol{\psi}^*$. However, in practice, we can at most obtain $\mathcal{L}_i^{(\text{sg})} \triangleq \mathcal{L}_i|_{\boldsymbol{\psi}^*\to\text{sg}[\boldsymbol{\psi}^*]}$, where sg stands for stop gradient, meaning we cannot obtain the gradient of $\boldsymbol{\psi}^*$ with respect to $\boldsymbol{\theta}$. Thus, regardless of $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$, their gradients in practice are biased.

This is where FGM comes in. Its idea is closer to the essence: losses $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ only focus on equality at the loss level, but for an optimizer, we need equality at the gradient level. Therefore, we need to find a new loss function $\mathcal{L}_4$ such that it satisfies:

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

That is, $\nabla_{\boldsymbol{\theta}}\mathcal{L}_4^{(\text{sg})} = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{1/2/3}$. Then, by using $\mathcal{L}_4$ as the loss function, we can achieve an unbiased optimization effect.

The derivation of FGM is also based on the identity (4), although its original derivation is somewhat tedious. For this article, we can start directly from $\mathcal{L}_3$, i.e., Eq. (9). The only term related to $\boldsymbol{\psi}^*$ is $\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\langle \boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)}, t), \boldsymbol{\varepsilon}\rangle]$. We calculate its gradient directly by applying "identity transformation then gradient" and "gradient then identity transformation" respectively to the operation $\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\|\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)}, t)\|^2]$ and comparing the results.

Identity transformation then gradient:

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

Gradient then identity transformation:

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\|\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t)\|^2] \\
=&\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\nabla_{\boldsymbol{\theta}}\|\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t)\|^2] = 2\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t)\rangle] \\
=&2\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\mathrm{sg}[\psi^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t)\rangle] + \underbrace{2\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\psi^*}(\mathrm{sg}[\boldsymbol{x}_t^{(g)}],t),\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t)\rangle]}_{\text{can apply Eq. (4)}} \\
=&2\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\mathrm{sg}[\psi^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t)\rangle] + 2\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\psi^*}(\mathrm{sg}[\boldsymbol{x}_t^{(g)}],t),\boldsymbol{\varepsilon}\rangle]
\end{aligned} \tag{12}$$

Note the third equality here: only the term $\boldsymbol{\epsilon}_{\psi^*}(\mathrm{sg}[\boldsymbol{x}_t^{(g)}],t)$ can have the identity (4) applied to it. This is because the $\boldsymbol{x}_t^{(g)}$ in $\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\mathrm{sg}[\psi^*]}(\boldsymbol{x}_t^{(g)},t)$ needs to be differentiated with respect to $\boldsymbol{\theta}$, and after differentiation, it is not necessarily a function of $\boldsymbol{x}_t^{(g)}$ alone, thus failing the condition for applying Eq. (4).

Now we have two results for $\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\|\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t)\|^2]$. Multiplying Eq. (11) by 2 and subtracting Eq. (12) gives:

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\langle\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle] = \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\|\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t)\|^2] = (11)\times 2 - (12) \\
=&2\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\mathrm{sg}[\psi^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle] - 2\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\langle\nabla_{\boldsymbol{\theta}}\boldsymbol{\epsilon}_{\mathrm{sg}[\psi^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t)\rangle] \\
=&2\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\langle\boldsymbol{\epsilon}_{\mathrm{sg}[\psi^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle] - \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\|\boldsymbol{\epsilon}_{\mathrm{sg}[\psi^*]}(\boldsymbol{x}_t^{(g)},t)\|^2] \\
=&\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[2\langle\boldsymbol{\epsilon}_{\mathrm{sg}[\psi^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle - \|\boldsymbol{\epsilon}_{\mathrm{sg}[\psi^*]}(\boldsymbol{x}_t^{(g)},t)\|^2]
\end{aligned} \tag{13}$$

Note the expression being differentiated at the end. All its $\boldsymbol{\psi}^*$ are marked with sg, indicating that we do not need to find its gradient with respect to $\boldsymbol{\theta}$. However, its gradient is equal to the exact gradient of $\mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\langle\boldsymbol{\epsilon}_{\psi^*}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle]$. By replacing the corresponding term in $\mathcal{L}_3$ with it, we obtain $\mathcal{L}_4$:

$$\mathcal{L}_4^{(\mathrm{sg})} = \mathbb{E}_{\boldsymbol{z},\varepsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\|\boldsymbol{\epsilon}_{\varphi^*}\|^2 - 2\langle\boldsymbol{\epsilon}_{\varphi^*}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle + 2\langle\boldsymbol{\epsilon}_{\mathrm{sg}[\psi^*]}(\boldsymbol{x}_t^{(g)},t),\boldsymbol{\varepsilon}\rangle - \|\boldsymbol{\epsilon}_{\mathrm{sg}[\psi^*]}(\boldsymbol{x}_t^{(g)},t)\|^2\right] \tag{14}$$

This is the final result of FGM. It only depends on $\mathrm{sg}[\boldsymbol{\psi}^*]$, but $\nabla_{\boldsymbol{\theta}}\mathcal{L}_4^{(\mathrm{sg})} = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{1/2/3}$ holds. Upon closer inspection, we find that $\mathcal{L}_4^{(\mathrm{sg})} = 2\mathcal{L}_2^{(\mathrm{sg})} - \mathcal{L}_1^{(\mathrm{sg})} = 2(\mathcal{L}_2^{(\mathrm{sg})} - 0.5\times\mathcal{L}_1^{(\mathrm{sg})})$. Thus, FGM essentially confirms SiD's choice of $\lambda = 0.5$ from a gradient perspective.

Incidentally, the original FGM paper describes the process within the ODE-based diffusion framework (flow matching). However, as mentioned in the previous article, neither SiD nor FGM actually utilizes the iterative generation process of the diffusion model; they only use the denoising model trained by the diffusion model. Therefore, whether it is the ODE, SDE, or DDPM framework is merely superficial; the denoising model is the essence. Thus, this article can introduce FGM using the notation from the previous SiD article.

4 Generalized Divergence

FGM has successfully derived the most fundamental gradient, but this only explains SiD's $\lambda = 0.5$. This means that if we want to explain the feasibility of other λ values, we must modify our starting point. To this end, let us return to the beginning and reflect on the generator's objective (3).

Readers familiar with diffusion models should know that the theoretical optimal solution for Eq. (1) can also be written as $\boldsymbol{\epsilon}_{\boldsymbol{\varphi}^*}(\boldsymbol{x}_t, t) = -\bar{\beta}_t \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$. Similarly, the optimal solution for Eq. (2) is $\boldsymbol{\epsilon}_{\boldsymbol{\psi}^*}(\boldsymbol{x}_t^{(g)}, t) = -\bar{\beta}_t \nabla_{\boldsymbol{x}_t^{(g)}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)})$. Here, $p(\boldsymbol{x}_t)$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)})$ are the distributions of the real data and the generator's data after adding noise, respectively. If you are unfamiliar with this result, you can refer to "Generative Diffusion Model Talk (V): SDE in General Framework" and "Generative Diffusion Model Talk (XVIII): Score Matching = Conditional Score Matching".

Substituting these two theoretical optimal solutions back into Eq. (3), we find that the generator is actually trying to minimize the Fisher divergence:

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

What we need to reflect on is the rationality and potential improvements of the Fisher divergence. As can be seen, $p_{\boldsymbol{\theta}}$ appears twice in the Fisher divergence. Now, I invite the reader to consider: **Which of these two occurrences of $p_{\boldsymbol{\theta}}$ is more important?**

The answer is the **second one**. To understand this fact, let's consider two cases: 1. Fix the first $p_{\boldsymbol{\theta}}$ and only Fix the second $p_{\boldsymbol{\theta}}$ and only optimize the first $p_{\boldsymbol{\theta}}$. What is the difference between their results? In the first case, there will likely be no change, meaning $p_{\boldsymbol{\theta}} = p$ can still be learned. In fact, since the Fisher divergence contains $\|\cdot\|^2$, the following more general conclusion is almost obviously true:

> As long as $r(\boldsymbol{x})$ is a distribution that is non-zero everywhere, $p(\boldsymbol{x}) = q(\boldsymbol{x})$ remains the theoretical optimal solution for the following generalized Fisher divergence:
>
> $$\mathcal{F}(p, q|r) = \int r(\boldsymbol{x}) \|\nabla_{\boldsymbol{x}} p(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} q(\boldsymbol{x})\|^2 d\boldsymbol{x} \tag{16}$$

To put it simply, the first occurrence of $p_{\boldsymbol{\theta}}$ is not important at all; it could be replaced by any other distribution, and the $\|\cdot\|^2$ term alone would ensure the two distributions are equal. However, the second case is different. If we fix the second $p_{\boldsymbol{\theta}}$ and only optimize the first $p_{\boldsymbol{\theta}}$, the theoretical optimal solution is:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)}) = \delta(\boldsymbol{x}_t^{(g)} - \boldsymbol{x}_t^*), \quad \boldsymbol{x}_t^* = \underset{\boldsymbol{x}_t^{(g)}}{\operatorname{argmin}} \left\|\nabla_{\boldsymbol{x}_t^{(g)}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(g)}) - \nabla_{\boldsymbol{x}_t^{(g)}} \log p(\boldsymbol{x}_t^{(g)})\right\|^2 \tag{17}$$

where δ is the Dirac delta distribution. This means the model only needs to generate the single sample that minimizes $\|\cdot\|^2$ to minimize the loss. This is, quite frankly, Mode Collapse! Therefore, the role of the first $p_{\boldsymbol{\theta}}$ in the Fisher divergence is not only secondary but potentially negative.

This inspires us: when using a gradient-based optimizer to train the model, it might be better to simply remove the gradient of the first $p_{\boldsymbol{\theta}}$. Thus, the following form of Fisher divergence is

a better choice:

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

In other words, $\mathcal{L}_5$ here is very likely to be a better starting point than $\mathcal{L}_1$. It is numerically equal to $\mathcal{L}_1$ but lacks a portion of the gradient:

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

Since $\nabla_{\boldsymbol{\theta}}\mathcal{L}_1$ has already been calculated by FGM as $\nabla_{\boldsymbol{\theta}}(2\mathcal{L}_2^{(\text{sg})} - \mathcal{L}_1^{(\text{sg})})$, using $\mathcal{L}_5$ as the starting point results in a practical loss function of $2\mathcal{L}_2^{(\text{sg})} - \mathcal{L}_1^{(\text{sg})} - \mathcal{L}_1^{(\text{sg})} = 2(\mathcal{L}_2^{(\text{sg})} - \mathcal{L}_1^{(\text{sg})})$. This explains the choice of $\lambda = 1$. As for choices where λ is slightly greater than 1, they are more extreme, essentially treating $-\mathcal{L}_1^{(\text{sg})}$ as an additional penalty term on top of $\mathcal{L}_5$ to further reduce the risk of mode collapse. Of course, since it is a pure penalty term, the weight should not be too large; according to SiD's experimental results, the training begins to collapse when $\lambda = 1.5$.

Incidentally, prior to FGM, the authors had another work "One-Step Diffusion Distillation through Score Implicit Matching", which also proposed a similar approach of changing the first $p_{\boldsymbol{\theta}}$ to $p_{\text{sg}[\theta]}$, but it did not explicitly discuss the rationality of this operation from the original form of Fisher divergence, making it slightly less complete.

5 Summary

This article introduced the subsequent theoretical developments of SiD (Score identity Distillation). The main content explains the λ parameter setting in SiD from a gradient perspective. The core part is the clever idea of accurately estimating SiD gradients discovered by FGM (Flow Generator Matching), which confirms the choice of $\lambda = 0.5$. On this basis, I extended the concept of Fisher divergence to explain the value of $\lambda = 1$.