A Very Simple Introduction to Diffusion Models

Robert Gower

October 21, 2022

This is a short note at how arrive at the correct loss function used to train Diffusion models with the shortest explanation possible. I avoid entirely the VDM (Variational Diffusion Model) viewpoint. Instead, here I motivate the loss directly as trying to minimize the error of the model in reversing one step of the diffusion model. Both this direct approach, and the VDM approach are equally formal. Furthermore, both approaches arrive at the correct loss function upto a multiplicative constant in time. In this sense, both approaches are equally impractical. I also show a quick equivalence to score based modelling using only simple calculus. Thus without resorting to Tweedie's formula or finding the reverse of a Stochastic Differential Equation.

1 What Does it Do?

Consider the task of generating natural images from random noise. Let $\mathcal{X} \subset \mathbb{R}^D$ be our space/set of natural images. Let $z \sim \mathcal{N}(0, \mathbf{I})$ where $z \in \mathbb{R}^d$. We want to find a map $x_\theta : \mathbb{R}^d \mapsto \mathbb{R}^d$ parametrized in $\theta \in \mathbb{R}^m$ such that

$$x = x_\theta(z)$$

where $x \in \mathcal{X}$ is a convincing natural image, and $\theta \in \mathbb{R}^m$ are the tunable parameters. We can take this further by using some conditional information $y \in \mathbb{R}^m$. For instance, y could be some embedding of a caption of this image, or it may be the label corresponding to dog. In which case we want a map such that

$$x_y = x_\theta(z \mid y) \tag{1}$$

where x_y is a natural image that is likely image conditioned on observing y. But here I will say no more on conditional information.

The problem of generating natural images from noise is very challenging. The other direction, generating noise from natural images is easy. So we will start there.

2 Target Data using a Diffusion

Let $x_0 \in \mathcal{X}$ be a natural image. To train our model to generate images from noise, we start with x_0 and transform it into noise $x_T \sim \mathcal{N}(0, I)$ and then ask that our model to recover x_0 from x_T. This turns out to be too difficult. So to help the model, we show it a sequence of images x_t with increasing amounts of noise.

Diffusion. To make this inversion task easier (and possible), we generate a sequence of noisy images, each with more noise. Let $(\alpha_t) > 0$ be a sequence of positive numbers. Our sequence of noisy images is given by

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t, \quad \text{for } t = 1, \ldots, n \text{ and } \epsilon_t \sim \mathcal{N}(0, I). \tag{2}$$

Typically $\alpha_1 = 10^{-4}$, and α_t linearly increases to $\alpha_T = 0.02$, and $T = 1000$. These x_t's will be the target for our model. Indeed our model will try to learn x_{t-1} from x_t, which is possible if $\sqrt{1-\alpha_t}$ is small.

3 How are they Trained?

We now allow our model x_θ to take two inputs: A noisy image x_t and a position t. Our objective is to predict the slightly less noisy image x_{t-1} from x_t. To do this end, we minimize the L2 loss between $x_\theta(x_t, t)$ and x_{t-1} that is

$$\min_\theta \mathop{\mathbb{E}}_{t\sim U(1,\ldots,T),x_0\sim\mathcal{X}} \left[\|x_\theta(x_t,t) - x_{t-1}\|^2\right], \tag{3}$$

where $t \sim U(1,\ldots,T)$ uniformly samples t from $\{1,\ldots,T\}$ and $x_0 \sim \mathcal{X}$ samples a natural image. In practice our model is reparametrized to mimick the structure of x_{t-1}. To see this structure, first note that the reverse process of (2) is given by

$$x_{t-1} = \frac{x_t - \sqrt{1-\alpha_t}\epsilon_t}{\sqrt{\alpha_t}}.$$

We could easily reverse this process if we knew ϵ_t. But in practice, for unseen/test data we don't know ϵ_t. Since we have this functional form for x_{t-1}, we might as well choose x_θ to mimick this form, that is

$$x_\theta(x_t,t) = \frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x^t,t)}{\sqrt{\alpha_t}}, \tag{4}$$

where ϵ_θ is now a parametrized model with parameters $\theta \in \mathbb{R}^m$. The objective of this $\epsilon_\theta(x_t,t)$ is to output the noisy part of x_t. Substituting (4) into the loss (3) we get

$$\min_\theta \mathop{\mathbb{E}}_{t\sim U(1,\ldots,T),x_0\sim\mathcal{M},\epsilon\sim\mathcal{N}(0,I)} \left[\frac{1-\alpha_t}{\alpha_t}\|\epsilon_\theta(x_t,t) - \epsilon\|^2\right]. \tag{5}$$

The above has an issue with cost, since computing x_t requires either saving all images $x_1,\ldots,x_t$ (typically 1000 images) or recomputing the diffusion. Luckily there is a fix.

Skipping ahead. In practice x_t can be sampled directly without generating the whole sequence (2). Indeed, since x_t is a summation of Gaussian's, it is also Gaussian (conditioned on x_0). Let $\bar{\alpha}_t = \sqrt{\prod_{i=1}^t \alpha_i}$. We can see this by expanding the recurrence (2):

$$\begin{aligned} x_t &= \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t \\ &= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-1}) + \sqrt{1-\alpha_t}\epsilon_t \\ &= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\epsilon_{t-1} + \sqrt{1-\alpha_t}\epsilon_t \end{aligned}$$

Now since $\sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\epsilon_{t-1} \sim \mathcal{N}(0,(\alpha_t - \alpha_t\alpha_{t-1})\mathbf{I})$ and $\epsilon_t \sim \mathcal{N}(0,(1-\alpha_t)\mathbf{I})$ are both normal random variables, we can use the formula

$$\mathcal{N}(\mu_1,\sigma_1^2) + \mathcal{N}(\mu_2,\sigma_2^2) = \mathcal{N}(\mu_1+\mu_2,\sigma_1^2+\sigma_2^2)$$

to conclude that

$$\sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\epsilon_{t-1} + \sqrt{1-\alpha_t}\epsilon_t \sim \mathcal{N}(0,(\alpha_t - \alpha_t\alpha_{t-1} + 1 - \alpha_t)\mathbf{I}) = \mathcal{N}(0,(1-\alpha_t\alpha_{t-1})\mathbf{I}).$$

Thus let $\epsilon_t^* \sim \mathcal{N}(0,\mathbf{I})$ we have that

$$\begin{aligned} x_t &= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon_t^* \\ &\vdots \\ &= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_0^* \ \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0,(1-\bar{\alpha}_t)\mathbf{I}). \end{aligned} \tag{6}$$

So we can sample x_t directly from x_0 and a normal random variable.

Using (6) we can also substitute out x_t and drop the normalizing constant $\frac{1-\alpha_t}{\alpha_t}$ to give

$$\min_{\theta} \mathop{\mathbb{E}}_{t\sim U(1,\ldots,T), x_0\sim\mathcal{M}, \epsilon\sim\mathcal{N}(0,I)} \left[\|\epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) - \epsilon\|^2\right]. \tag{7}$$

This normalizing constant $\frac{1-\alpha_t}{\alpha_t}$ re-weight the terms depending on t. Thus dropping this normalizing constant effectively changes the distribution of t. That is, solving (7) is not equal to solving (5). Yet this is what is done in practice. This lack of formality in simply dropping this constant is also an issue with the VDM viewpoint.

The full training algorithm is given by Algorithm 1. This is apparently amongst the most used training loss and parametrization.

Algorithm 1 Training

```
for $k = 1, \ldots, K$ do
    Sample $x_0 \sim \mathcal{X}$
    Sample $t \in \{1, \ldots, T\}$ uniformly
    Sample $\epsilon \sim \mathcal{N}(0, I)$
    Update    $\theta_{k+1} = \theta_k - \gamma_k \nabla_\theta \|\epsilon_{\theta_k}(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) - \epsilon\|^2$    ▷ Gradient update
```

Generation/Inference. With the trained model, we can generate images by running the diffusion in the reverse direction. That is, since isolating x^{t-1} in (2) gives

$$x_{t-1} = \frac{x_t - \sqrt{1-\alpha_t}\epsilon_t}{\sqrt{\alpha_t}}, \quad \text{for } t = 1, \ldots, n \text{ and } \epsilon_t \sim \mathcal{N}(0, I). \tag{8}$$

Consequently our generation algorithm is

$$x_T \sim \mathcal{N}(0, I) \tag{9}$$

$$x_{t-1} = \frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x^t, t)}{\sqrt{\alpha_t}}, \quad \text{for } t = T, \ldots, 0. \tag{10}$$

4 Score based Viewpoint

Diffusion models are often described through their connection with score based modelling. Let $p(x_0)$ be the probability distribution of real images. The score based approach tries to fit a score function $s_\theta(x_t, t)$ to the gradient of true probability distribution as follows

$$\min_{\theta} \mathop{\mathbb{E}}_{x_0\sim p(x_0), x_t\sim p(x_t|x_0)} \left[\|s_\theta(x_t) - \nabla \log p(x_t)\|^2\right]. \tag{11}$$

Why do this? Intuitively (and formally) you can sample real images by following $\nabla \log p(x)$ towards the most probable images. That this, $\nabla \log p(x)$ points in the steepest ascent direction. Thus if we had access to this gradient, we could iterate gradient ascent, but with some noise to ensure exploration as follows

$$x_T \sim \mathcal{N}(0, \mathbf{I}) \tag{12}$$

$$x_{t-1} = x_t + \gamma_t \nabla \log p(x_t) + \sqrt{\gamma_t}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \mathbf{I}), \quad \text{for } t = T, \ldots, 1, \tag{13}$$

where $\gamma_t \to 0$ are the learning rate. This process is known as Langevin dynamics.

This can also be show to be equivalent to (7) which can be useful, in particular for modelling the conditional sampling. To show this equivalence, first we need to show that (11) is equivalent to (up to additive constants) solving

$$\min_{\theta} \mathop{\mathbb{E}}_{t\sim U(1,\ldots,T), x_0\sim p(x_0), x_t\sim p(x_t|x_0)} \left[\|s_\theta(x_t) - \nabla_{x_t} \log p(x_t|x_0)\|^2\right]. \tag{14}$$

Proof. Expanding the squares of (14) we have

$$\|s_\theta(x_t) - \nabla \log p(x_t|x_0)\|^2 = \underbrace{\|s_\theta(x_t)\|^2}_{=:I^*} - 2\underbrace{\langle s_\theta(x_t), \nabla \log p(x_t|x_0)\rangle}_{=:II^*} + \underbrace{\|\nabla \log p(x_t|x_0)\|^2}_{=:III^*}. \tag{15}$$

Expanding the squares of (11) we have

$$\|s_\theta(x_t) - \nabla \log p(x_t)\|^2 = \underbrace{\|s_\theta(x_t)\|^2}_{=:I} - 2\underbrace{\langle s_\theta(x_t), \nabla \log p(x_t)\rangle}_{=:II} + \underbrace{\|\nabla \log p(x_t)\|^2}_{=:III}. \tag{16}$$

Now note that III and III^* does not depend on θ, so we can drop it in our minimization problem. As for I, we have that $I = I^*$.

As for the II, using that

$$p(x_t) = \int p(x_0, x_t) dx_0 = \int p(x_t|x_0)p(x_0)dx_0.$$

we have that

$$\begin{aligned}
\mathop{\mathbb{E}}_{x_0 \sim p(x_0), x_t \sim p(x_t|x_0)} [\langle s_\theta(x_t), \nabla \log p(x_t)\rangle] &= \mathop{\mathbb{E}}_{x_t \sim p(x_t)} [\langle s_\theta(x_t), \nabla \log p(x_t)\rangle] \\
&= \mathop{\mathbb{E}}_{x_t \sim p(x_t)} \left[\left\langle s_\theta(x_t), \frac{\nabla_{x_t} p(x_t)}{p(x_t)}\right\rangle\right] \\
&= \int \langle s_\theta(x_t), \nabla p(x_t)\rangle \, dx_t \\
&= \int_{x_t} \left\langle s_\theta(x_t), \nabla_{x_t} \int p(x_t|x_0)p(x_0)dx_0 \right\rangle dx_t \\
&= \int_{x_t} \left\langle s_\theta(x_t), \int \nabla_{x_t} p(x_t|x_0)p(x_0)dx_0 \right\rangle dx_t \\
&= \int \left\langle s_\theta(x_t), \int p(x_t|x_0)p(x_0)\nabla_{x_t} \log(p(x_t|x_0))dx_0 \right\rangle dx_t \\
&= \int \int p(x_t|x_0)p(x_0) \langle s_\theta(x_t), \nabla_{x_t} \log(p(x_t|x_0))\rangle \, dx_0 dx_t \\
&= \mathop{\mathbb{E}}_{(x_0) \sim p(x_0)} \left[\mathop{\mathbb{E}}_{x_t \sim p(x_t|x_0)} [\langle s_\theta(x_t), \nabla_{x_t} \log(p(x_t|x_0))\rangle]\right] \\
&= II^*
\end{aligned}$$

This is now the cross product term of (14), thus (11) and (14) are equal up to additive terms. □

Now since we know that x_t given x_0 has a Gaussian distribution, we have from (6) that

$$p(x_t|x_0) = \frac{1}{\sqrt{1-\bar{\alpha}_t}\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t}}.$$

Taking the gradient of the above gives

$$\begin{aligned}
\nabla_{x_t} \log p(x_t|x_0) &= -\nabla_{x_t} \frac{\frac{1}{2}(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t} && (17) \\
&= -\frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)}{1-\bar{\alpha}_t}. && (18)
\end{aligned}$$

Now using (6) again, that is

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

gives

$$\nabla_{x_t} \log p(x_t|x_0) = -\frac{\sqrt{1-\bar{\alpha}_t}\epsilon}{1-\bar{\alpha}_t}. \tag{19}$$

Consequently if we chose to parametrize the score function as

$$s_\theta(x_t) = -\frac{\sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t)}{1-\bar{\alpha}_t}. \tag{20}$$

Substituting both (19) and (20) into (14) gives

$$\min_\theta \mathop{\mathbb{E}}_{t\sim U(1,\ldots,T), x_0\sim p(x_0), x_t \sim p(x_t|x_0)} \left[\frac{1}{\sqrt{1-\bar{\alpha}_t}} \|\epsilon_\theta(x_t,t) - \epsilon\|^2 \right], \tag{21}$$

which after substituting in $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, and again dropping the multiplicative factor $\frac{1}{\sqrt{1-\bar{\alpha}_t}}$ we return to (7).

5 Variational Diffusion Model Viewpoint

If you ever read a paper on diffusion, most likely they will adopt the VDM (Variational Diffusion Model) viewpoint. For instance the paper “Understanding Diffusion Models: A Unified Perspective” by Calvin Luo is an excellent read.

The VDM viewpoint models the reverse process using a variational family that is assumed to be Gaussian. Further this Gaussian and has the same variance as the forward process. Let

$$q(x^t \mid x^{t-1}) = \mathcal{N}(\sqrt{\alpha_t} x_{t-1}, (1-\alpha_t)\mathbf{I})$$

be the distributions of the forward diffusion. Since we use independent noise at each iteration we can decompose the full probability as

$$q(x_1, \ldots, x_T|x_0) = \prod_{t=1}^T q(x_t \mid x_{t-1}).$$

We now need to model the reverse process. Let $p_\theta(x)$ such that we want $p_\theta(x)$ to be large for $x \in \mathcal{X}$. Let’s assume that the probability reverse process is also Markov in that

$$p(x_{0:T}) = p(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t) \tag{22}$$

where we have imposed that

$$p_\theta(x_{t-1} \mid x_T, \ldots, x_t) = p_\theta(x_{t-1} \mid x_t).$$

Each $p_\theta(x_{t-1} \mid x_t)$ is modelled as a sequence of Gaussians with the same variance as the forward diffusion. Only the mean is left to determine.

From this perspective our objective is to fit a distribution p_θ such that $p_\theta(x)$ is large for $x \in \mathcal{X}$. That is we want to find θ^* such that $\log p_\theta(x)$ is large or

$$\theta^* \in \max \mathop{\mathbb{E}}_{x\sim p} \left[\log p_\theta(x)\right].$$

To minimize this we use that

$$\log p_\theta(x_0) = \int p_\theta(x_0, \dots, x_T) dx_{1:T}.$$

This can be done by maximizing the lower bound in the following

$$\log p_\theta(x_0) \geq \underset{q(x_{1:T} \mid x_0)}{\mathbb{E}} \left[\log \frac{p_\theta(x_0, \dots, x_T)}{q(x_{1:T} \mid x_0)} \right] = -KL(q(x_{1:T} \mid x_0), p_\theta(x_0, \dots, x_T)).$$

Using the decomposition (22), and after many steps we have that

$$KL(q(x_{1:T} \mid x_0), p_\theta(x_0, \dots, x_T)) = -\underset{q(x_1|x_0)}{\mathbb{E}} [\log p_\theta(x_0|x_1)] + \sum_{t=2}^{T} KL(q(x_{t-1}|x_t, x_0), p_\theta(x_{t-1}|x_t)).$$

Finally assuming that

$$p_\theta(x_t \mid x_{t-1}) \sim \mathcal{N}(\mu_\theta, \sigma_t^2 \mathbf{I}),$$

where σ_t is the variance of $q(x_{t-1}|x_t, x_0)$ then minimizing the ELBO is equivalent to (7) upto multiplicative constants, which are ignored anyway! The key behind this equivalence is that

$$KL(q(x_{t-1}|x_t, x_0), p_\theta(x_{t-1}|x_t)) = \frac{1}{\sigma_t^2} \|\mu_\theta - \mu_{q_t}\|^2.$$

The final step is just to compute the mean μ_{q_t}, which has the form

$$\mu_{q_t} = \text{const}_1 x_t + \text{const}_2 \epsilon$$

where $\epsilon \sim \mathcal{N}(0, I)$. We then simply choose to parametrize

$$\mu_\theta = \text{const}_1 x_t + \text{const}_2 \epsilon_\theta$$

which brings up back to exactly the same form (up to multiplicative constants).

A Auxiliary Lemmas

For the score interpretation we need Tweedie's formula

$$\mathbb{E}[\mu_z \mid z] = z + \Sigma_z \nabla_z \log p(z)$$

where $z \sim \mathcal{N}(\mu_z, \Sigma_z)$ and I guess $\mu \sim \mathcal{N}$.

Lemma A.1 (Stein's lemma)**.** Let $X \sim \mathcal{N}(0, 1)$ and let $f(x)$ be a function such that $\mathbb{E}[Xf(X)]$ and $\mathbb{E}[f'(X)]$ are finite. It follows that

$$\mathbb{E}[xf(x)] = \mathbb{E}[f'(x)]. \tag{23}$$

Furthermore if

$$Y = \mu + \sigma X$$

then

$$\mathbb{E}[(Y - \mu) f(Y)] = \sigma \mathbb{E}[f'(Y)].$$

Proof. Note that $\mathcal{N}(0, 1) = e^{-x^2/2}$. Also note that

$$\int x e^{-x^2/2} dx = -e^{-x^2/2} \tag{24}$$

which follows by the change of variable $y = x^2$.

Now let $g'(x) = xe^{-x^2/2}$. using integration by parts we have that

$$\begin{aligned}
\mathbb{E}\left[Xf(X)\right] &= \int f(x)g'(x)dx \\
&= \ f(x)g(x)|_{-\infty}^{\infty} - \int f'(x)g(x)dx \\
&= -f(x)e^{-x^2/2}|_{-\infty}^{\infty} + \int f'(x)e^{-x^2/2}dx && \text{Using (24)} \\
&= \mathbb{E}\left[f'(X)\right]
\end{aligned}$$

Finally letting $Y = \mu + \sigma X$ thus $\frac{Y-\mu}{\sigma} = X$ we have

$$\mathbb{E}\left[Yf(Y)\right] = \mathbb{E}\left[(\mu + \sigma X)f(\mu + \sigma X)\right] = \mu\mathbb{E}\left[f(Y)\right] + \sigma\mathbb{E}\left[Xf(\mu + \sigma X)\right].$$

Now we can compute $\mathbb{E}\left[Xf(\mu + \sigma X)\right]$ using the same steps as above

$$\begin{aligned}
\mathbb{E}\left[Xf(\mu + \sigma X)\right] &= \int f(\mu + \sigma x)g'(x)dx \\
&= \ f(\mu + \sigma x)g(x)|_{-\infty}^{\infty} - \int \sigma f'(\mu + \sigma x)g(x)dx \\
&= -f(\mu + \sigma x)e^{-x^2/2}|_{-\infty}^{\infty} + \sigma \int f'(\mu + \sigma X)e^{-x^2/2}dx && \text{Using (24)} \\
&= \sigma^2\mathbb{E}\left[f'(\mu + \sigma X)\right]
\end{aligned}$$

Thus

$$\mathbb{E}\left[(Y - \mu)f(Y)\right] = \sigma\mathbb{E}\left[f'(Y)\right].$$

□