Deep Generative Models

Chapter 6: Generation by Diffusion Process

Ali Bereyhi

ali.bereyhi@utoronto.ca

Department of Electrical and Computer Engineering

University of Toronto

Summer 2025

Diffusion by SDE: *Approximative Approach*

With discrete time steps $t = 0, \dots, T$: *an SDE of the form*

$$x_t = x_{t-1} - \beta_t x_{t-1} \mathrm{d}t + \sqrt{\gamma_t \mathrm{d}t} \varepsilon_t$$

We can make sure that the variance is preserved by setting

$$\gamma_t \mathrm{d}t + (1 - \beta_t \mathrm{d}t)^2 = 1$$

At the end of the day, we remain by

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1 - \alpha_t} \varepsilon_t$$

where in this process we have

- α_t *is close to one* $\rightsquigarrow 1 - \alpha_t$ *is close to zero*
- $\varepsilon_t \sim \mathcal{N}(0, 1)$ *is independent in each time step*

Diffusion by SDE: *Forward Diffusion*

We can now *build a forward diffusion by this SDE*

$$x_0 \xrightarrow{\beta_1} x_1 \xrightarrow{\beta_2} \cdots \longrightarrow x_{t-1} \xrightarrow{\beta_t} x_t \xrightarrow{\beta_{t+1}} \cdots \xrightarrow{\beta_T} x_T$$

Key Observation

This SDE is fundamentally defined by β_t

This diffusion process takes us from data to noise

- *We need a reverse diffusion to get back from noise to data*
- *This is described by the reverse SDE*

Diffusion by SDE: *Reverse Diffusion*

The reverse SDE formula *specifies the reverse diffusion*

$$x_0 \xrightarrow{\beta_1} x_1 \xrightarrow{\beta_2} \cdots \longrightarrow x_{t-1} \xrightarrow{\beta_t} x_t \xrightarrow{\beta_{t+1}} \cdots \xrightarrow{\beta_T} x_T$$

$$x_0 \xleftarrow{\beta_1} x_1 \xleftarrow{\beta_2} \cdots \longleftarrow x_{t-1} \xleftarrow{\beta_t} x_t \xleftarrow{\beta_{t+1}} \cdots \xleftarrow{\beta_T} x_T$$

It is important to keep in mind that

- *The samples in reverse and forward trajectories are different*
- *They are though coming from the same distribution if*

  the reverse trajectory traverses exactly reverse SDE

Diffusion by SDE: *Reverse Diffusion*

We can use the reverse SDE formula *to find the reverse diffusion*

$$x_0 \xleftarrow{\beta_1} x_1 \xleftarrow{\beta_2} \cdots \longleftarrow x_{t-1} \xleftarrow{\beta_t} x_t \xleftarrow{\beta_{t+1}} \cdots \xleftarrow{\beta_T} x_T$$

The reverse diffusion is described by

$$x_{t-1} = \left(2 - \sqrt{\alpha_t}\right) x_{t-1} + \left(1 - \alpha_t\right) s_t\left(x_t\right) + \sqrt{1 - \alpha_t} \varepsilon_t$$

where $s_t\left(x_t\right)$ is the score of distribution in time t, i.e.,

$$s_t\left(x_t\right) = \nabla_x \log P_t\left(x\right)$$

with $P_t\left(x\right)$ being the distribution of x_t

Our Initial Challenge: *Score Matching*

We need to estimate $s_t(x_t)$: *in last part we saw that we can*

- *use the noising process to estimate*

$$\hat{s}_t(x_t) = -\frac{\varepsilon_t}{\sqrt{1-\alpha_t}}$$

↳ *We sample several noise samples and compute these estimates*
↳ *We then train the model $s_{\mathbf{w}}(x_t, \alpha_t)$ on these samples*

- *use a computational denoiser to approximate the expression*

$$s_t(x_t) = \frac{\mathbb{E}\left\{\sqrt{\alpha_t} x_{t-1} | x_t\right\} - x_t}{1-\alpha_t}$$

↳ *We can train an AE to approximate the optimal denoiser*

Later Challenges

It turns out that *this approach does not lead to a stable solution*

1. *The score estimate is not accurate*
   - ↳ *The model is trained by extremely noisy samples*
   - ↳ *The model has limited capacity as compared to Bayes optimal*
2. *The reverse trajectory is a first-order approximation*

$$x\left(t + \mathrm{d}t\right) \approx x\left(t\right) + \mathrm{d}x\left(t\right)$$

but to be accurate we should write

$$x\left(t + \mathrm{d}t\right) = x\left(t\right) + \mathrm{d}x\left(t\right) + \frac{1}{2}\mathrm{d}^2 x\left(t\right) + \cdots$$

   - ↳ *We cannot access* $\mathrm{d}^j x\left(t\right)$ *for* $j > 1$ *as it is an SDE*
   - ↳ *Higher differentials can get very large due to Brownian motion*

Alternative Look: *Diffusion as Markov Chain*

As mentioned earlier: *we could see forward diffusion as a Markov chain*

$$x_0 \longrightarrow x_1 \longrightarrow \cdots \longrightarrow x_{t-1} \longrightarrow x_t \longrightarrow \cdots \longrightarrow x_T$$

We know mathematically that the reverse chain can exist

$$x_0 \longleftarrow x_1 \longleftarrow \cdots \longleftarrow x_{t-1} \longleftarrow x_t \longleftarrow \cdots \longleftarrow x_T$$

Maybe we could directly learn it: in the forward process we have

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1 - \alpha_t} \varepsilon_t$$

So, we could say the forward Markov chain is

$$Q\left(x_t | x_{t-1}\right) \equiv \mathcal{N}\left(\sqrt{\alpha_t} x_{t-1}, 1 - \alpha_t\right)$$

Alternative Look: *Reverse Diffusion*

$$x_0 \longleftarrow x_1 \longleftarrow \cdots \longleftarrow x_{t-1} \longleftarrow x_t \longleftarrow \cdots \longleftarrow x_T$$

Now the question is: *what is the reverse Markov chain*

$$P\left(x_{t-1}|x_t\right)$$

which takes from distribution P_t to distribution P_{t-1}?

Computational Solution

We consider a computational model $P_{\mathbf{w}}$

$$P_{\mathbf{w}}\left(x_{t-1}|x_t, t\right) = F_{\mathbf{w}}\left(x_{t-1}, t\right)$$

and try to find a way to train for mimicking the reverse trajectory

A Deep Look at *Forward Diffusion*

$$x_0 \xrightarrow{\alpha_1} x_1 \xrightarrow{\alpha_2} \cdots \longrightarrow x_{t-1} \xrightarrow{\alpha_t} x_t \xrightarrow{\alpha_{t+1}} \cdots \xrightarrow{\alpha_T} x_T$$

What does happen in forward process?

$$x_1 = \sqrt{\alpha_1} x_0 + \sqrt{1-\alpha_1}\varepsilon_1$$

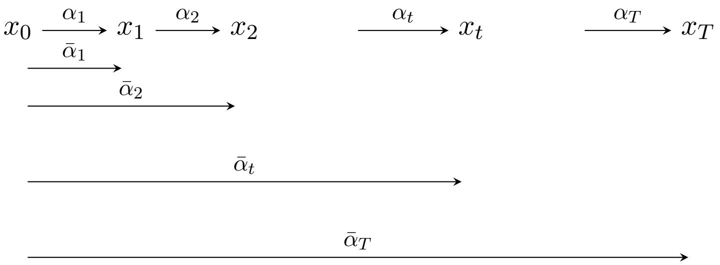
$$x_2 = \sqrt{\alpha_2} x_1 + \sqrt{1-\alpha_2}\varepsilon_2 = \sqrt{\alpha_1\alpha_2} x_0 + \underbrace{\sqrt{\alpha_2}\sqrt{1-\alpha_1}\varepsilon_1 + \sqrt{1-\alpha_2}\varepsilon_2}_{\sqrt{1-\alpha_1\alpha_2}\bar{\varepsilon}_2}$$

$$\vdots$$

$$x_t = \sqrt{\prod_{i=1}^{t} \alpha_i} x_0 + \sqrt{1-\prod_{i=1}^{t} \alpha_i} \bar{\varepsilon}_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\bar{\varepsilon}_t$$

Direct Forward Links

We could also describe it with direct links from x_0 *to* x_t



And, we note that

$$\lim_{t\uparrow\infty} \bar{\alpha}_t = \lim_{t\uparrow\infty} \sqrt{\prod_{i=1}^{t} \alpha_i} = 0$$

Direct Forward Links: *Explicit Expression*

$$x_0 \xrightarrow{\alpha_1} x_1 \xrightarrow{\alpha_2} \cdots \longrightarrow x_{t-1} \xrightarrow{\alpha_t} x_t \xrightarrow{\alpha_{t+1}} \cdots \xrightarrow{\alpha_T} x_T$$

What does happen in forward process?

Direct Forward Links

We can say that since

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

we have the direct forward conditional distributions as

$$Q\left(x_t | x_0\right) \equiv \mathcal{N}\left(\sqrt{\bar{\alpha}_t} x_0, 1 - \bar{\alpha}_t\right)$$

Learning Reverse Diffusion: *Reverse Processes*

$$x_0 \xleftarrow{P_{\mathbf{w},1}} x_1 \xleftarrow{P_{\mathbf{w},2}} \cdots \longleftarrow x_{t-1} \xleftarrow{P_{\mathbf{w},t}} x_t \longleftarrow \cdots \xleftarrow{P_{\mathbf{w},T}} x_T$$

In the reverse trajectory: *we start with $x_T \sim \mathcal{N}(0,1)$ and go as*

$$x_{t-1} \sim P_{\mathbf{w}}(x_{t-1}|x_t, t)$$

What would be the marginal distribution in reverse trajectory at $T-1$?

We can use marginalization to write

$$\hat{P}_{T-1}(x_{T-1}) = \int P(x_{T-1}, x_T)\, \mathrm{d}x_T$$
$$= \int P_T(x_T)\, P_{\mathbf{w}}(x_{T-1}|x_T, T)\, \mathrm{d}x_T$$

Learning Reverse Diffusion: *Reverse Processes*

$$x_0 \xleftarrow{P_{\mathbf{w},1}} x_1 \xleftarrow{P_{\mathbf{w},2}} \cdots \longleftarrow x_{t-1} \xleftarrow{P_{\mathbf{w},t}} x_t \longleftarrow \cdots \xleftarrow{P_{\mathbf{w},T}} x_T$$

What if we go all the way back to 0*?*

$$\begin{aligned}
\hat{P}_0\left(x_0\right) &= \int P\left(x_{0:T}\right) \prod_{t=1}^{T} \mathrm{d}x_t \\
&= \int P_T\left(x_T\right) P_{\mathbf{w}}\left(x_{T-1}|x_T, T\right) \dots P_{\mathbf{w}}\left(x_0|x_1, 1\right) \prod_{t=1}^{T} \mathrm{d}x_t \\
&= \int P_T\left(x_T\right) \prod_{t=1}^{T} P_{\mathbf{w}}\left(x_{t-1}|x_t, t\right) \mathrm{d}x_t
\end{aligned}$$

Learning Reverse Diffusion by *Maximum Likelihood*

$$x_0^j \xleftarrow{P_{\mathbf{w},1}} x_1 \xleftarrow{P_{\mathbf{w},2}} \cdots \longleftarrow x_{t-1} \xleftarrow{P_{\mathbf{w},t}} x_t \longleftarrow \cdots \xleftarrow{P_{\mathbf{w},T}} x_T$$

We want to see the same final distribution as out data

$$D_{\mathrm{KL}} \left(\hat{P}_0 \| P_{\mathrm{data}} \right) \approx 0$$

So, we need to maximize the likelihood

$$\log \mathcal{L} \left(\mathbf{w} \right) = \sum_j \log \hat{P}_0 \left(x_0^j \right)$$

$$= \sum_j \log \int P_T \left(x_T \right) P_{\mathbf{w}} \left(x_0^j | x_1, 1 \right) \prod_{t=2}^{T} P_{\mathbf{w}} \left(x_{t-1} | x_t, t \right) \mathrm{d}x_t \mathrm{d}x_1$$

Maximum Likelihood Learning

Maximum Likelihood on Reverse Trajectory

We learn reverse trajectory by maximizing the log-likelihood on our dataset

$$\max_{\mathbf{w}} \log \int P_T\left(x_T\right) \prod_{t=1}^{T} P_{\mathbf{w}}\left(x_{t-1} | x_t, t\right) \mathrm{d}x_t$$

+ *Don't we care about the samples in between?!*
– Why should we?!

MLE Learning: *Training Objective*

Let us do some notations simplification: *we define*

$$P_{\mathbf{w}}(x_0) = \int P_T(x_T) \prod_{t=1}^{T} P_{\mathbf{w}}(x_{t-1}|x_t, t)\, \mathrm{d}x_t$$

In MLE, we want to maximize

$$\log P_{\mathbf{w}}(x_0)$$

This is however computationally very hard!

+ *Isn't this the same thing we had in VAE?!*
– *Right!*

MLE Training: *Finding an ELBO*

We first write the equation compactly as

$$P_{\mathbf{w}}(x_0) = \int \underbrace{P_T(x_T) \prod_{t=1}^{T} P_{\mathbf{w}}(x_{t-1}|x_t, t)}_{P_{\mathbf{w}}(x_{0:T})} \mathrm{d}x_t = \int P_{\mathbf{w}}(x_{0:T}) \mathrm{d}x_{1:T}$$

Now we do importance sampling: say we know good distribution $\Lambda(x_{1:T}|x_0)$

$$\begin{aligned} \log P_{\mathbf{w}}(x_0) &= \log \int P_{\mathbf{w}}(x_{0:T}) \mathrm{d}x_{1:T} \\ &= \log \int \frac{P_{\mathbf{w}}(x_{0:T})}{\Lambda(x_{1:T}|x_0)} \Lambda(x_{1:T}|x_0) \mathrm{d}x_{1:T} \\ &= \log \mathbb{E}_{x_{1:T} \sim \Lambda} \left\{ \frac{P_{\mathbf{w}}(x_{0:T})}{\Lambda(x_{1:T}|x_0)} \right\} \end{aligned}$$

Finding ELBO

Next we use Jensen's inequality to write that

$$\log P_{\mathbf{w}}\left(x_0\right) = \log \mathbb{E}_{x_{1:T}\sim\Lambda}\left\{\frac{P_{\mathbf{w}}\left(x_{0:T}\right)}{\Lambda\left(x_{1:T}|x_0\right)}\right\}$$
$$\geqslant \mathbb{E}_{x_{1:T}\sim\Lambda}\left\{\log\frac{P_{\mathbf{w}}\left(x_{0:T}\right)}{\Lambda\left(x_{1:T}|x_0\right)}\right\} = \mathrm{ELBO}\left(\mathbf{w}|x_0\right)$$

This describes an ELBO

Implicit MLE via ELBO Maximization

We can maximize likelihood by maximizing the ELBO

+ *Shall we again think of Λ to be learned?!*
– *Not really! Actually we can explicitly compute a good Λ*

Posterior Calculation

The best $\Lambda\left(x_{1:T}|x_0\right)$ *is given by posterior, i.e.,*

$$\Lambda\left(x_{1:T}|x_0\right) = Q\left(x_{1:T}|x_0\right)$$

If we try to open up this posterior in reverse direction we have

$$\begin{aligned}\Lambda\left(x_{1:T}|x_0\right) &= Q\left(x_{1:T}|x_0\right)\\ &= Q\left(x_T|x_0\right)Q\left(x_{T-1}|x_T,x_0\right)\dots Q\left(x_{t-1}|x_{t:T},x_0\right)\dots\\ &= Q\left(x_T|x_0\right)\prod_t Q\left(x_{t-1}|x_{t:T},x_0\right)\end{aligned}$$

Posterior Calculation

We are interested in such distributions

$$Q\left(x_{t-1}|x_{t:T}, x_0\right) = Q\left(x_{t-1}|x_t, x_{t+1:T}, x_0\right)$$

Let's do a bit of calculations

$$\boxed{Q\left(x_{t-1}|x_t, x_{t+1:T}, x_0\right)} Q\left(x_{t+1:T}|x_t, x_0\right) = Q\left(x_{t-1}, x_{t+1:T}|x_t, x_0\right)$$

$$Q\left(x_{t-1}|x_t, x_{t+1:T}, x_0\right) \prod_{i=t}^{T-1} Q\left(x_{i+1}|x_i\right) =$$

Alternatively, we could say

$$Q\left(x_{t+1:T}|x_{t-1}, x_t, x_0\right) Q\left(x_{t-1}|x_t, x_0\right) = Q\left(x_{t-1}, x_{t+1:T}|x_t, x_0\right)$$

$$\prod_{i=t}^{T-1} Q\left(x_{i+1}|x_i\right) \boxed{Q\left(x_{t-1}|x_t, x_0\right)} =$$

Posterior Calculation

So, the posterior can be simplified as

$$Q\left(x_{1:T}|x_0\right) = Q\left(x_T|x_0\right) \prod_t Q\left(x_{t-1}|x_t, x_0\right)$$

+ *How can we compute this?*
– *We can use the forward process*

We do know $Q\left(x_t|x_{t-1}\right)$ *and* $Q\left(x_t|x_0\right)$*: so we could write*

$$\begin{aligned} Q\left(x_{t-1}|x_t, x_0\right) &= \frac{Q\left(x_{t-1}, x_t, x_0\right)}{Q\left(x_t, x_0\right)} \\ &= \frac{Q\left(x_{t-1}, x_t|x_0\right) P\left(x_0\right)}{Q\left(x_t|x_0\right) P\left(x_0\right)} = \frac{Q\left(x_{t-1}|x_0\right) Q\left(x_t|x_{t-1}\right)}{Q\left(x_t|x_0\right)} \end{aligned}$$

Posterior Calculation: *Gaussian Posterior*

Recall that

$$Q\left(x_t|x_{t-1}\right) \equiv \mathcal{N}\left(\sqrt{\alpha_t}x_{t-1}, 1-\alpha_t\right)$$
$$Q\left(x_t|x_0\right) \equiv \mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0, 1-\bar{\alpha}_t\right)$$

Thus, it is easy to show that

$$Q\left(x_{t-1}|x_t, x_0\right) \equiv \mathcal{N}\left(\eta_t\left(x_0, x_t\right), \rho_t^2\right)$$

for the mean and variance

$$\eta_t\left(x_0, x_t\right) = \frac{\sqrt{\bar{\alpha}_{t-1}}\left(1-\alpha_t\right)}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_{t-1}}\left(1-\bar{\alpha}_t\right)}{1-\bar{\alpha}_t}x_t$$
$$\rho_t^2 = \frac{\left(1-\bar{\alpha}_{t-1}\right)\left(1-\alpha_t\right)}{1-\bar{\alpha}_t}$$

ELBO Calculation

The ELBO is hence given by

$$
\begin{aligned}
\mathrm{ELBO}\left(\mathbf{w}|x_0\right) &= \mathbb{E}_{\Lambda}\left\{\log \frac{P_{\mathbf{w}}\left(x_{0:T}\right)}{\Lambda\left(x_{1:T}|x_0\right)}\right\} \\
&= \mathbb{E}_{Q}\left\{\log \frac{P\left(x_T\right)\prod_{t=1}^{T} P_{\mathbf{w}}\left(x_{t-1}|x_t\right)}{Q\left(x_T|x_0\right)\prod_{t=2}^{T} Q\left(x_{t-1}|x_t, x_0\right)}\right\} \\
&= \mathbb{E}_{Q}\left\{\log\left[\frac{P\left(x_T\right)}{Q\left(x_T|x_0\right)}\prod_{t=2}^{T}\frac{P_{\mathbf{w}}\left(x_{t-1}|x_t\right)}{Q\left(x_{t-1}|x_t, x_0\right)} P_{\mathbf{w}}\left(x_0|x_1\right)\right]\right\} \\
&= \mathbb{E}_{Q}\left\{-\log \frac{Q\left(x_T|x_0\right)}{P\left(x_T\right)} - \sum_{t=2}^{T}\log\frac{Q\left(x_{t-1}|x_t, x_0\right)}{P_{\mathbf{w}}\left(x_{t-1}|x_t\right)} + \log P_{\mathbf{w}}\left(x_0|x_1\right)\right\} \\
&= -D_{\mathrm{KL}}\left(Q_{0\to t}\|\mathcal{N}^0\right) - \sum_{t=2}^{T} D_{\mathrm{KL}}\left(Q_{t-1\leftarrow t,0}\|P_{\mathbf{w},t}\right) + \mathbb{E}_{Q}\left\{\log P_{\mathbf{w}}\left(x_0|x_1\right)\right\}
\end{aligned}
$$

ELBO Maximization: *Sample Loss*

Our ultimate goal is to

$$\max_{\mathbf{w}} \mathrm{ELBO}\left(\mathbf{w}|x_0\right)$$

which by dropping terms that do not depend on $\mathbf{w}$*, is done by*

$$\min_{\mathbf{w}} \sum_{t=2}^{T} D_{\mathrm{KL}}\left(Q_{t-1\leftarrow t,0} \| P_{\mathbf{w},t}\right) - \mathbb{E}_Q\left\{\log P_{\mathbf{w}}\left(x_0|x_1\right)\right\}$$

Thus, we have the following sample loss

$$R\left(\mathbf{w}|x_0\right) = \sum_{t=2}^{T} D_{\mathrm{KL}}\left(Q_{t-1\leftarrow t,0} \| P_{\mathbf{w},t}\right) - \mathbb{E}_Q\left\{\log P_{\mathbf{w}}\left(x_0|x_1\right)\right\}$$

ELBO Maximization: *Gaussian Reverse Process*

We typically consider a Gaussian reverse process

$$P_{\mathbf{w}}\left(x_{t-1}|x_t\right) \equiv \mathcal{N}\left(\mu_{\mathbf{w}}\left(x_t, t\right), \sigma_t^2\right)$$

This results in the risk

$$R\left(\mathbf{w}|x_0\right) \propto \sum_{t=2}^{T} \left\|\mu_{\mathbf{w}}\left(x_t, t\right) - \eta_t\left(x_0, x_t\right)\right\|^2 + \left\|\mu_{\mathbf{w}}\left(x_1, 1\right) - x_0\right\|^2$$

This looks like a reconstruction!

This motivated DPM and DDPM framework