DDPM, Every Equation

mateo.guaman1998

June 2025

1 Abstract

In this work, the authors present a generative model for image synthesis, which is a class of latent variable models, where the latents are same dimension as the input and correspond to intermediate noisy versions of the data when going smoothly from full noise to clean data.

They train a "weighted variational bound", meaning that the loss of the harder problems (denoising when the input is mostly noise, little to no signal) will be weighted more than the loss of the easier problems (denoising an almost clean image).

They obtain this variational bound by making a new connection between diffusion probabilistic models and denoising score matching with Langevin dynamics. Explicitly, what this means is that they formulate this process of denoising iteratively as estimating the noise at a certain timestep and subtracting that noise from the current noisy image, rather than estimating the 1-timestep denoised image mean directly from the current noisy image, or other potential alternatives.

If you think of the score as the negative of the noise, i.e. adding the score would denoise a noisy image, pushing it closer to the original data distribution, then this training process is equivalent to score matching.

And the connection to Langevin dynamics, in which to maximize the logprob of the data you move the generated samples in the direction of the score plus add some random noise, is that the denoising process that the authors suggest will also involve adding some noise during the denoising process.

This is similar to autoregressive decoding given that they progressively generate cleaner versions of an image, but rather than doing this over spatial coordinates, they do it over noise levels.

2 Introduction

DDPM learns the parameterized transitions of a denoising Markov Chain using variational inference to generate samples from the original data distribution. These learned transitions reverse a separate noising Markov Chain which starts with the original data, and adds small Gaussian noise at every transition. Given

that the noising process uses small Gaussian noise, the transitions of this learned denoising model can also be parameterized as Gaussians, which is amenable to learning.

The main result of this paper is that DDPM is capable of state-of-the-art image generation with high sample quality, matching and exceeding FID and inception scores of other SotA models such as GANs or autoregressive models like PixelCNN. The primary contribution is that parameterizing this denoising process by learning to predict the noise ϵ at each timestep (rather than directly predicting $\tilde{\mu}_t$, the one-step denoised mean, or x_0, the fully denoised image) reveals an equivalence to denoising score matching with annealed Langevin dynamics. This ϵ-parameterization achieved the best sampling results.

While the generated samples are of high quality, the log-likelihoods of the original data computed by the learned model are lower than those of other state-of-the-art approaches. However, these log-likelihoods can be computed exactly for DDPMs via ELBO, unlike EBMs and score-matching models whose log-likelihoods need to be estimated via potentially noisy processes such as annealed importance sampling, leading to unreliable estimates.

The authors provide an information-theoretic analysis showing two key insights:

1. More than half of the bits in the codelength budget are spent learning to denoise imperceptible details, which explains why the log-likelihoods of DDPMs are lower than other models. There is a direct relation between bits/dim and log-likelihoods:

$$n_{\text{bits}} = -\log_2(p_\theta(x_0))$$

so lower n_{bits} means higher (better) log-likelihood. If DDPMs didn't waste bits on imperceptible noise, they would have competitive log-likelihoods. This also explains why image quality remains high despite poor log-likelihood: the capacity spent on data structure is high and leads to good visual generation, while the extra bits only hurt compression performance but don't hurt image quality.

2. DDPMs perform a generalized version of progressive decoding via denoising, compared to traditional autoregressive decoding which uses spatial order.

3 Background

3.1 ELBO Derivation

The log likelihood of the model is $\log p_\theta(x_0)$. The training objective is to maximize the log likelihood of the model with respect to the training data:

$$\theta^* = \max_\theta \mathbb{E}_{x_0 \sim q(x_0)}[\log p_\theta(x_0)] = \min_\theta \mathbb{E}_{x_0 \sim q(x_0)}[-\log p_\theta(x_0)]$$

Derivation:

$$
\begin{aligned}
\mathbb{E}_{x_0\sim q(x_0)}[-\log p_\theta(x_0)] &= \mathbb{E}_{x_0\sim q(x_0)}\left[-\log \int p_\theta(x_{0:T})\, dx_{1:T}\right] \\
&= \mathbb{E}_{x_0\sim q(x_0)}\left[-\log \int p_\theta(x_{0:T}) \frac{q(x_{1:T} \mid x_0)}{q(x_{1:T} \mid x_0)}\, dx_{1:T}\right] \\
&= \mathbb{E}_{x_0\sim q(x_0)}\left[-\log \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right]\right] \\
&\leq \mathbb{E}_{x_0\sim q(x_0)}\, \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right] \quad \text{(Jensen's Inequality)} \\
&= \mathbb{E}_{x_0\sim q(x_0)}\, \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[-\log \frac{p_\theta(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)}{\prod_{t=1}^{T} q(x_t \mid x_{t-1})}\right] \\
&= \mathbb{E}_{x_0\sim q(x_0)}\, \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[-\log p_\theta(x_T) - \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})}\right]
\end{aligned}
$$

From the paper:

$$
\mathbb{E}_{x_0}[-\log p_\theta(x_0)] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right] = \mathbb{E}_q\left[-\log p(x_T) - \sum_{t\geq 1} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t|x_{t-1})}\right] =: L
$$

Note: The paper uses shorthand notation $\mathbb{E}_q[\cdot]$ to represent the joint expectation $\mathbb{E}_{x_0\sim q(x_0)}\mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}[\cdot]$.

Data Scaling, Reverse Process Decoder, and L_0

The authors assume that the image data is represented as integers in $\{0, 1, \ldots, 255\}$ scaled linearly to $[-1, 1]$. The authors want to compute an exact likelihood for the data generated, but the issue is that the last denoising step produces a continuous Gaussian distribution, while the actual data consists of discrete integer pixels. To deal with this, they compute the probability of each independent pixel using binning (integrating the continuous Gaussian over integer-width bins), and multiply it all together to obtain the probability of the image. For boundary values, they integrate bins that extend to infinity. This simple but elegant binning strategy allows them to obtain a variational bound that provides a lossless codelength for discrete data, without any extra operations such as adding random noise to make discrete values continuous, or Jacobians required for scaling.

4 Sum of Gaussians and Reparameterization Trick

In DDPMs, the noising distribution

$$
q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)\mathbf{I}) \tag{1}
$$

can be rewritten as:

$$
x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}) \tag{2}
$$

using the reparameterization trick for Gaussians, with $\beta_t = 1 - \alpha_t$ the variance of noise added at each step $t \in [T]$.

We'd like to have a way to get the noised sample x_t at any timestep from just the original data point x_0 at any timestep (rather than having to run a recursion), and sample Gaussian noise only once. We will show below that, by sum of Gaussians and reparameterization trick, this is possible using:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_0, \epsilon_0 \sim \mathcal{N}(0, \mathbf{I}) \tag{3}$$

where $\bar{\alpha_t} = \prod_{i=1}^{t} \alpha_i$.

4.1 Reparameterization Trick

For $z \sim \mathcal{N}(\mu, \sigma^2)$:

$$z = \mu + \sigma\epsilon, \epsilon \sim \mathcal{N}(0, 1)$$

and vice-versa. The same applies for multivariate Gaussians.

4.2 Sum of Gaussians

Let X and Y be independent, normally distributed random variables:

$$\begin{aligned} X &\sim \mathcal{N}(\mu_X, \sigma_X^2) \\ Y &\sim \mathcal{N}(\mu_Y, \sigma_Y^2) \\ Z &= X + Y \end{aligned}$$

Then their sum is also normally distributed according to:

$$Z \sim \mathcal{N}(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2) \tag{4}$$

4.3 Unroll the DDPM Recursion

Let's unroll the recursion for the noising distribution:

$$\begin{aligned} x_1 &= \sqrt{\alpha_1} x_0 + \sqrt{1 - \alpha_1} \epsilon_0, \epsilon_0 \sim \mathcal{N}(0, \mathbf{I}) \\ x_2 &= \sqrt{\alpha_2} x_1 + \sqrt{1 - \alpha_2} \epsilon_1, \epsilon_1 \sim \mathcal{N}(0, \mathbf{I}) \\ &= \sqrt{\alpha_2} \left(\sqrt{\alpha_1} x_0 + \sqrt{1 - \alpha_1} \epsilon_0 \right) + \sqrt{1 - \alpha_2} \epsilon_1, \epsilon_0, \epsilon_1 \sim \mathcal{N}(0, \mathbf{I}) \quad \text{Plug in } x_1 \\ &= \underbrace{\sqrt{\alpha_1 \alpha_2} x_0 + \sqrt{(1 - \alpha_1)\alpha_2} \epsilon_0}_{m} + \underbrace{\sqrt{1 - \alpha_2} \epsilon_1}_{n}, \epsilon_0, \epsilon_1 \sim \mathcal{N}(0, \mathbf{I}) \quad \text{Expand} \end{aligned}$$

Let's split up the expanded equation into two terms:

$$
\begin{aligned}
x_2 &= m + n \\
m &= \sqrt{\alpha_1 \alpha_2} x_0 + \sqrt{(1-\alpha_1)\alpha_2}\epsilon_0, \epsilon_0 \sim \mathcal{N}(0, \mathbf{I}) \\
n &= \sqrt{1-\alpha_2}\epsilon_1, \epsilon_1 \sim \mathcal{N}(0, \mathbf{I})
\end{aligned}
$$

If we revert the parameterization trick, we can write these down as Gaussian random variables:

$$
\begin{aligned}
m &\sim \mathcal{N}\left(\sqrt{\alpha_1 \alpha_2} x_0, (1-\alpha_1)\alpha_2\right) \\
n &\sim \mathcal{N}\left(0, 1-\alpha_2\right)
\end{aligned}
$$

Note the lack of square root for the variance, given that when we use the reparameterization trick, we use standard deviation σ, but when defining a Gaussian, we use variance σ^2.

By sum of Gaussians:

$$
\begin{aligned}
x_2 &\sim \mathcal{N}\left(\sqrt{\alpha_1 \alpha_2} x_0, (1-\alpha_1)\alpha_2 + 1 - \alpha_2\right) && (5) \\
&\sim \mathcal{N}\left(\sqrt{\alpha_1 \alpha_2} x_0, \alpha_2 - \alpha_1\alpha_2 + 1 - \alpha_2\right) && (6) \\
&\sim \mathcal{N}\left(\sqrt{\alpha_1 \alpha_2} x_0, 1 - \alpha_1\alpha_2\right) && (7)
\end{aligned}
$$

Applying the reparameterization trick yet again to this last equation yields:

$$
\begin{aligned}
x_2 &= \sqrt{\alpha_1 \alpha_2} x_0 + \sqrt{1-\alpha_1\alpha_2}\epsilon_0, \epsilon_0 \sim \mathcal{N}(0, \mathbf{I}) && (8) \\
&= \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_0, \epsilon_0 \sim \mathcal{N}(0, \mathbf{I}) && (9)
\end{aligned}
$$

The same can be shown for future timesteps.

5 DDPM Objective Derivation

From the DDPM paper:

Training is performed by optimizing the usual variational bound on negative log likelihood:

$$
\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L \tag{10}
$$

So let's try to derive this.

The log likelihood of the model is $\log p_\theta(x_0)$. The training objective is to maximize the log likelihood of the model wrt the training data:

$$
\begin{aligned}
\theta^* &= \max_\theta \mathbb{E}_{x_0 \sim q(x_0)}\left[\log p_\theta(x_0)\right] && (11) \\
&= \min_\theta \mathbb{E}_{x_0 \sim q(x_0)}\left[-\log p_\theta(x_0)\right] && (12)
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}_{x_0 \sim q(x_0)}[-\log p_\theta(x_0)] &= \mathbb{E}_{x_0 \sim q(x_0)}\left[-\log \int p_\theta(x_{0:T})\, dx_{1:T}\right] \\
&= \mathbb{E}_{x_0 \sim q(x_0)}\left[-\log \int p_\theta(x_{0:T}) \frac{q(x_{1:T} \mid x_0)}{q(x_{1:T} \mid x_0)}\, dx_{1:T}\right] \\
&= \mathbb{E}_{x_0 \sim q(x_0)}\left[-\log \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)}\left[\frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right]\right] \\
&\leq \mathbb{E}_{x_0 \sim q(x_0)}\, \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)}\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right] && \text{Via Jensen's Inequal} \\
&= \mathbb{E}_{x_0 \sim q(x_0)}\, \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)}\left[-\log \frac{p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)}{\prod_{t=1}^{T} q(x_t \mid x_{t-1})}\right] \\
&= \mathbb{E}_{x_0 \sim q(x_0)}\, \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)}\left[-\log p(x_T) - \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})}\right]
\end{aligned}
$$

In the paper, in Equation 5, the authors rewrite this objective in the following way:

$$
\begin{aligned}
L &= \mathbb{E}_{x_0 \sim q(x_0)}\, \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)}\left[-\log p(x_T) - \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})}\right] \\
&= \mathbb{E}_{x_0 \sim q(x_0)}\, \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)}\left[-\log p(x_T) - \log \frac{p_\theta(x_0 \mid x_1)}{q(x_1 \mid x_0)} - \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})}\right] && \text{Extract } t=1 \text{ from sum}
\end{aligned}
$$

The eventual goal is for some of these terms to be converted into KL divergence for a different interpretation, but currently

$$\log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})}$$

does not have the right KL form. I believe this is why they then use Bayes theorem to get the next step, but not fully sure. But effectively, we are going to convert the denominator into a different thing that results in that division to look like a KL divergence:

$$
\begin{aligned}
q(x_t \mid x_{t-1}) &= q(x_t \mid x_{t-1}, x_0) && \text{Using Markov Property} \\
&= \frac{q(x_{t-1}, x_t \mid x_0)}{q(x_{t-1} \mid x_0)} \\
&= \frac{q(x_{t-1} \mid x_t, x_0) q(x_t \mid x_0)}{q(x_{t-1} \mid x_0)}
\end{aligned}
$$

Plugging this back into the denominator of the sum, we get:

$$L = \mathbb{E}_{x_0 \sim q(x_0)} \mathbb{E}_{x_{1:T} \sim q(x_{1:T} \mid x_0)} \left[-\log p(x_T) - \log \frac{p_\theta(x_0 \mid x_1)}{q(x_1 \mid x_0)} - \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} \frac{q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)} \right]$$

We can expand this last term to see if we can cancel out any terms:

$$\begin{aligned}
&\sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} \frac{q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)} \\
&= \log \prod_{t=2}^{T} \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} \frac{q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)} \\
&= \log \left(\frac{p_\theta(x_1 \mid x_2)}{q(x_1 \mid x_2, x_0)} \frac{q(x_1 \mid x_0)}{q(x_2 \mid x_0)} \cdot \frac{p_\theta(x_2 \mid x_3)}{q(x_2 \mid x_3, x_0)} \frac{q(x_2 \mid x_0)}{q(x_3 \mid x_0)} \cdot \ldots \cdot \frac{p_\theta(x_{T-1} \mid x_T)}{q(x_{T-1} \mid x_T, x_0)} \frac{q(x_{T-1} \mid x_0)}{q(x_T \mid x_0)} \right) \\
&= \log \left[\prod_{t=2}^{T} \left(\frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} \right) \frac{q(x_1 \mid x_0)}{q(x_T \mid x_0)} \right] \\
&= \log \frac{q(x_1 \mid x_0)}{q(x_T \mid x_0)} + \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)}
\end{aligned}$$

We plug this back in:

$$\begin{aligned}
L &= \mathbb{E}_{x_0 \sim q(x_0)} \mathbb{E}_{x_{1:T} \sim q(x_{1:T} \mid x_0)} \left[-\log p(x_T) - \log \frac{p_\theta(x_0 \mid x_1)}{q(x_1 \mid x_0)} - \log \frac{q(x_1 \mid x_0)}{q(x_T \mid x_0)} - \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} \right] \\
&= \mathbb{E}_{x_0 \sim q(x_0)} \mathbb{E}_{x_{1:T} \sim q(x_{1:T} \mid x_0)} \left[-\log \left(p(x_T) \cdot \frac{p_\theta(x_0 \mid x_1)}{q(x_1 \mid x_0)} \cdot \frac{q(x_1 \mid x_0)}{q(x_T \mid x_0)} \right) - \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} \right] \\
&= \mathbb{E}_{x_0 \sim q(x_0)} \mathbb{E}_{x_{1:T} \sim q(x_{1:T} \mid x_0)} \left[-\log \left(\frac{p(x_T)}{q(x_T \mid x_0)} \cdot p_\theta(x_0 \mid x_1) \right) - \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} \right] \\
&= \mathbb{E}_{x_0 \sim q(x_0)} \mathbb{E}_{x_{1:T} \sim q(x_{1:T} \mid x_0)} \left[-\log \frac{p(x_T)}{q(x_T \mid x_0)} - \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} - \log p_\theta(x_0 \mid x_1) \right] \\
&= \mathbb{E}_{x_0 \sim q(x_0)} \mathbb{E}_{x_{1:T} \sim q(x_{1:T} \mid x_0)} \left[-\log \frac{p(x_T)}{q(x_T \mid x_0)} \right] \\
&+ \mathbb{E}_{x_0 \sim q(x_0)} \mathbb{E}_{x_{1:T} \sim q(x_{1:T} \mid x_0)} \left[\sum_{t=2}^{T} -\log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} \right] \\
&+ \mathbb{E}_{x_0 \sim q(x_0)} \mathbb{E}_{x_{1:T} \sim q(x_{1:T} \mid x_0)} \left[-\log p_\theta(x_0 \mid x_1) \right]
\end{aligned}$$

Now, I have the following 3 terms:

Term 1

$$
\begin{aligned}
&\mathbb{E}_{x_0\sim q(x_0)}\mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[-\log\frac{p(x_T)}{q(x_T\mid x_0)}\right]\\
&=\int q(x_0)\left(\int q(x_{1:T}\mid x_0)\left(-\log\frac{p(x_T)}{q(x_T\mid x_0)}\right)dx_{1:T}\right)dx_0\\
&=\int q(x_0)\left(\int q(x_T\mid x_0)q(x_{1:T-1}\mid x_T,x_0)\left(-\log\frac{p(x_T)}{q(x_T\mid x_0)}\right)dx_{1:T-1}dx_T\right)dx_0\\
&=\int q(x_0)\left(\int q(x_{1:T-1}\mid x_T,x_0)dx_{1:T-1}\cdot\int q(x_T\mid x_0)\left(-\log\frac{p(x_T)}{q(x_T\mid x_0)}\right)dx_T\right)dx_0\\
&=\int q(x_0)\left(\int q(x_T\mid x_0)\left(-\log\frac{p(x_T)}{q(x_T\mid x_0)}\right)dx_T\right)dx_0\\
&=\mathbb{E}_{x_0\sim q(x_0)}\left[\mathbb{E}_{x_T\sim q(x_T|x_0)}\left[-\log\frac{p(x_T)}{q(x_T\mid x_0)}\right]\right]\\
&=\mathbb{E}_{x_0\sim q(x_0)}\left[\mathbb{E}_{x_T\sim q(x_T|x_0)}\left[\log\frac{q(x_T\mid x_0)}{p(x_T)}\right]\right]\quad\text{by log rules}\\
&=\mathbb{E}_{x_0\sim q(x_0)}\left[D_{\mathrm{KL}}\big(q(x_T\mid x_0)\,\|\,p(x_T)\big)\right]\quad\text{by def. of KL.}
\end{aligned}
$$

Term 2

$$
\mathbb{E}_{x_0\sim q(x_0)}\mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\sum_{t=2}^{T}-\log\frac{p_\theta(x_{t-1}\mid x_t)}{q(x_{t-1}\mid x_t,x_0)}\right]
$$

Let's start with a single term of the sum, given x_0 and t:

$$
\begin{aligned}
&\mathbb{E}_{x_0\sim q(x_0)}\mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[-\log\frac{p_\theta(x_{t-1}\mid x_t)}{q(x_{t-1}\mid x_t,x_0)}\right]\\
&=\int q(x_0)\left(\int q(x_{1:T}\mid x_0)\left(-\log\frac{p_\theta(x_{t-1}\mid x_t)}{q(x_{t-1}\mid x_t,x_0)}\right)dx_{1:T}\right)dx_0
\end{aligned}
$$

The integrand depends only on (x_0, x_t, x_{t-1}), so we'll integrate out all other terms:

$$
q(x_1,\ldots,x_{t-2},x_{t-1},x_t,x_{t+1},\ldots,x_T\mid x_0)=q(x_{t-1},x_t\mid x_0)\,q(x_{\neg\{t-1,t\}}\mid x_{t-1},x_t,x_0)
$$

Plugging back in and integrating out, we get:

$$= \int q(x_0) \left[\int \int q(x_t \mid x_0) q(x_{t-1} \mid x_t, x_0) \left(-\log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} \right) dx_{t-1} dx_t \right] dx_0$$

$$= \int q(x_0) \left[\int q(x_t \mid x_0) \left[\int q(x_{t-1} \mid x_t, x_0) \left(-\log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} \right) dx_{t-1} \right] dx_t \right] dx_0$$

$$= \mathbb{E}_{x_0 \sim q(x_0)} \left[\mathbb{E}_{x_t \sim q(x_t \mid x_0)} \left[D_{\mathrm{KL}}\big(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t)\big) \right] \right].$$

Term 3

$$\begin{aligned}
\mathbb{E}_{x_0 \sim q(x_0)} \mathbb{E}_{x_{1:T} \sim q(x_{1:T} \mid x_0)} \left[-\log p_\theta(x_0 \mid x_1) \right] &= \int q(x_0) \left(\int q(x_{1:T} \mid x_0) \left(-\log p_\theta(x_0 \mid x_1) \right) dx_{1:T} \right) dx_0 \\
&= \int q(x_0) \left(\int q(x_1 \mid x_0) \left(-\log p_\theta(x_0 \mid x_1) \right) dx_1 \right) dx_0 \\
&= \mathbb{E}_{x_0 \sim q(x_0)} \mathbb{E}_{x_1 \sim q(x_1 \mid x_0)} \left[-\log p_\theta(x_0 \mid x_1) \right].
\end{aligned}$$

So altogether:

$$\begin{aligned}
L = \mathbb{E}_{x_0 \sim q(x_0)} \Bigg[& D_{\mathrm{KL}}\big(q(x_T \mid x_0) \,\|\, p(x_T)\big) + \sum_{t=2}^{T} \mathbb{E}_{x_t \sim q(x_t \mid x_0)} \Big[D_{\mathrm{KL}}\big(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t)\big) \Big] \\
& + \mathbb{E}_{x_1 \sim q(x_1 \mid x_0)} \left[-\log p_\theta(x_0 \mid x_1) \right] \Bigg].
\end{aligned}$$

6 Obtaining an analytical Gaussian for $q(x_{t-1}|x_t, x_0)$

In this section, the goal is to obtain an analytical Gaussian for $q(x_{t-1}|x_t, x_0)$ with known mean and variance so that we can in turn compute the KL divergence for the term inside the sum analytically, instead of having to estimate it.

In this section, we will ignore and remove terms that do not depend on x_{t-1}. Our higher-level goal is to compute the KL divergence of two Gaussians analytically. To compute the KL divergence between two multivariate Gaussians, we only need to know their means and covariance matrices, but we don't need to care about scaling factors. Further, we can get the mean and std of this Gaussian in quadratic form only from the x_{t-1}^2 and x_{t-1} coefficients, and can ignore everything else.

Just a tad more formally, for an isotropic multivariate Gaussian distribution $p(x)$, $\log p(x) = ax^2 + bx + c$. Then, we can compute the mean and variance vectors:

$$\sigma = -\frac{1}{2a} \tag{13}$$

$$\mu = \sigma b \tag{14}$$

Using Bayes' rule, we can see that $q(x_{t-1}|x_t, x_0)$ factors into Gaussian terms:

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1})q(x_{t-1}|x_0)}{q(x_t|x_0)} \tag{15}$$
$$\propto q(x_t|x_{t-1})q(x_{t-1}|x_0) \tag{16}$$

We can ignore the denominator term since it does not depend on x_{t-1}. Now, let's recall some definitions and results from earlier:

$$\alpha_t := 1 - \beta_t \tag{17}$$
$$\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s \tag{18}$$

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{19}$$
$$q(x_{t-1}|x_0) := \mathcal{N}(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}}x_0, (1-\bar{\alpha}_{t-1})I) \tag{20}$$
$$q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I) \tag{21}$$

Product of Gaussians

The product of two Gaussians is also a Gaussian. We want to find the parameters of the resulting Gaussian from the product of $q(x_t|x_{t-1})$ and $q(x_t|x_0)$.

$$\begin{aligned}
&q(x_t|x_{t-1}) \cdot q(x_t|x_0) \\
&= (2\pi)^{-k/2} \det(\beta_t I)^{-1/2} \exp\left(-\frac{1}{2}(x_t - \sqrt{1-\beta_t}x_{t-1})^T(\beta_t I)^{-1}(x_t - \sqrt{1-\beta_t}x_{t-1})\right) \\
&\cdot (2\pi)^{-k/2} \det((1-\bar{\alpha}_{t-1})I)^{-1/2} \exp\left(-\frac{1}{2}(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)^T((1-\bar{\alpha}_{t-1})I)^{-1}(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)\right) \\
&\propto \exp\left(-\frac{1}{2}\left(\frac{(x_t - \sqrt{\alpha_t}x_{t-1})^2}{\beta_t} + \frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)^2}{1-\bar{\alpha}_{t-1}}\right)\right) \\
&= \exp\left(-\frac{1}{2}\left(\frac{x_t^2 - 2\sqrt{\alpha_t}x_t x_{t-1} + \alpha_t x_{t-1}^2}{\beta_t} + \frac{x_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}}x_0 x_{t-1} + \bar{\alpha}_{t-1}x_0^2}{1-\bar{\alpha}_{t-1}}\right)\right) \\
&\propto \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)x_{t-1}^2 + \left(\frac{-2\sqrt{\alpha_t}x_t}{\beta_t} + \frac{-2\sqrt{\bar{\alpha}_{t-1}}x_0}{1-\bar{\alpha}_{t-1}}\right)x_{t-1}\right)\right)
\end{aligned}$$

For a Gaussian $p(x)$, $\log p(x) = Ax^2 + Bx + C$. The parameters of the resulting Gaussian, the variance Σ and the mean μ, can be found using the formulas:

$$\Sigma^{-1} = -2A$$
$$\Sigma^{-1}\mu = B$$

This implies:

$$\Sigma = -\frac{1}{2A}$$
$$\mu = \Sigma \cdot B = -\frac{B}{2A}$$

From our expression, we can identify A and B:

$$A = -\frac{1}{2}\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right) = -\frac{1}{2}\left(\frac{\alpha_t - \bar{\alpha}_t + \beta_t}{\beta_t(1-\bar{\alpha}_{t-1})}\right)$$
$$B = -\frac{1}{2}\left(\frac{-2\sqrt{\alpha_t}x_t}{\beta_t} - \frac{2\sqrt{\bar{\alpha}_{t-1}}x_0}{1-\bar{\alpha}_{t-1}}\right) = \frac{\sqrt{\alpha_t}}{\beta_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0$$

Then, we can calculate Σ:

$$\begin{aligned}\Sigma &= -\frac{1}{2 \cdot \left(-\frac{1}{2}\left(\frac{\alpha_t - \bar{\alpha}_t + \beta_t}{\beta_t(1-\bar{\alpha}_{t-1})}\right)\right)} \\ &= \frac{\beta_t(1-\bar{\alpha}_{t-1})}{\alpha_t - \bar{\alpha}_t + \beta_t} \\ &= \beta_t \frac{(1-\bar{\alpha}_{t-1})}{\alpha_t - \bar{\alpha}_t + 1 - \alpha_t} \\ \Sigma &= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\end{aligned}$$

And now we calculate μ:

$$\mu = \left(\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\right)\left(\frac{\sqrt{\alpha_t}}{\beta_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)$$
$$\mu = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\beta_t\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}x_0$$

7 Reverse process and $L_{1:T-1}$

So we have

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}|\tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I)$$

where,

$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t$$
$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

We also know we are going to learn:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad \text{for } 1 < t \leq T$$

Let $\Sigma_\theta(x_t, t) = \sigma_t^2 I$ be untrained, time-dependent constants

$$\rightarrow p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$$

Now we can choose how to parameterize $\mu_\theta(x_t, t)$:

$$L_{t-1} = \mathbb{E}_q[D_{KL}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t))]$$

$$= D_{KL}(\mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I) \parallel \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I))$$

The KL divergence between two multivariate Gaussian distributions is:

$$D_{KL}(\mathcal{N}_0 \parallel \mathcal{N}_1) = \frac{1}{2}\left[\text{tr}(\Sigma_1^{-1}\Sigma_0) - k + (\mu_1 - \mu_0)^T\Sigma_1^{-1}(\mu_1 - \mu_0) + \ln\frac{\det\Sigma_1}{\det\Sigma_0}\right]$$

In our case:

$$D_{KL}(\mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I) \parallel \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)) \tag{22}$$

$$= \frac{1}{2}\left[\text{tr}((\sigma_t^2 I)^{-1}\tilde{\beta}_t I) - K + (\mu_\theta(x_t, t) - \tilde{\mu}_t(x_t, x_0))^T(\sigma_t^2 I)^{-1}(\mu_\theta(x_t, t) - \tilde{\mu}_t(x_t, x_0)) + \ln\frac{\det(\sigma_t^2 I)}{\det(\tilde{\beta}_t I)}\right] \tag{23}$$

$$= \frac{1}{2\sigma_t^2}||\mu_\theta(x_t, t) - \tilde{\mu}_t(x_t, x_0)||^2 + C \quad \text{Simply by separating terms that depend on } \theta \text{ and constants wrt } \theta \tag{24}$$

In the paper, they flip the order of μ:

$$L_{t-1} = \mathbb{E}_{x_0, x_t|x_0}\left[\frac{1}{2\sigma_t^2}||\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)||^2 + C\right]$$

Now, recall $x_t(x_0, \epsilon) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$

$$\rightarrow x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t(x_0, \epsilon) - \sqrt{1 - \bar{\alpha}_t}\epsilon)$$

We can plug this back into $\tilde{\mu}_t(x_t, x_0)$

$$\begin{aligned}
\tilde{\mu}_t(x_t, x_0) &= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\left(\frac{1}{\sqrt{\bar{\alpha}_t}}x_t(x_0,\epsilon) - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\epsilon\right) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t \\
&= \frac{\beta_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}x_t - \frac{\beta_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\epsilon + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t \\
&= \left(\frac{\beta_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\right)x_t - \frac{\beta_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\epsilon \\
&= \frac{(1-\bar{\alpha}_t)(1-\alpha_t) + \alpha_t(1-\bar{\alpha}_{t-1})(1-\bar{\alpha}_t)}{(1-\bar{\alpha}_t)^2\sqrt{\alpha_t}}x_t - \frac{\beta_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\epsilon \\
&= \frac{(1-\alpha_t) + \alpha_t(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}x_t - \frac{\beta_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\epsilon \\
&= \frac{1-\alpha_t+\alpha_t-\alpha_t\bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}x_t - \frac{\beta_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\epsilon \\
&= \frac{1-\bar{\alpha}_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}x_t - \frac{\beta_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\epsilon \\
&= \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{1-\bar{\alpha}_t}\epsilon\right)
\end{aligned}$$

$$\rightarrow L_{t-1} - C = \mathbb{E}_{x_0, x_t|x_0,\epsilon}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(x_t(x_0,\epsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right) - \mu_\theta(x_t(x_0,\epsilon), t)\right\|^2\right]$$

Note that x_t is an input to the model, so we can choose to parameterize μ_θ as follows:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\underbrace{\epsilon_\theta(x_t, t)}_{\text{learned}}\right)$$

7.1 Training Objective

Now, recall the training objective for L_{t-1}:

$$
\begin{aligned}
L_{t-1} - C &= \mathbb{E}_{x_0, x_t | x_0, \epsilon} \left[\frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right) - \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) \right\|^2 \right] \\
&= \mathbb{E}_{x_0, x_t | x_0, \epsilon} \left[\frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} x_t - \frac{\beta_t}{\sqrt{a_t}\sqrt{1-\bar{\alpha}_t}} \epsilon - \frac{1}{\sqrt{\alpha_t}} x_t + \frac{\beta_t}{\sqrt{\alpha_t}\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right\|^2 \right] \\
&= \mathbb{E}_{x_0, x_t | x_0, \epsilon} \left[\frac{1}{2\sigma_t^2} \left\| \frac{\beta_t}{\sqrt{\alpha}\sqrt{1-\bar{\alpha}_t}} (\epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) - \epsilon) \right\|^2 \right] \\
&= \mathbb{E}_{x_0, x_t | x_0, \epsilon} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)} \left\| \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) - \epsilon \right\|^2 \right] \\
\implies L_{t-1} - C &= \mathbb{E}_{x_0, \epsilon} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)} \left\| \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) - \epsilon \right\|^2 \right]
\end{aligned}
$$

7.2 Simplified training objective

For simplicity, the authors simplified the full training objective by:

1. Getting rid of the L_T term since it does not contain any trainable parameters
2. Getting rid of the scaling factor, leading to effectively training a weighted objective that places higher loss weight on the large denoising steps, and smaller on the near-imperceptible denoising ones.
3. Effectively ignore the continuous to discrete conversion edge effects and added noise

They claim they obtained higher sample quality.

$$
L_{\text{simple}}(\theta) := \mathbb{E}_{t \sim U[-1,1], x_0, \epsilon \sim \mathcal{N}(\epsilon; 0, I)} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{q - \bar{\alpha}_t}\epsilon, t) \right\|^2 \right] \quad (25)
$$

7.3 Inference time denoising

And to denoise image x_t at timestep t, we can sample

$$
\begin{aligned}
x_{t-1} &\sim p_\theta(x_{t-1} | x_t) \\
&\sim \mathcal{N}\left(x_{t-1}; \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right), \sigma_t^2 I \right)
\end{aligned}
$$

via reparameterization trick:

$$
x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z, \quad z \sim \mathcal{N}(0, I)
$$

8 Loss to Gradient

Scratch:

$$L = \mathbb{E}_{x_0, x_{1:T}} \left[-\log p(x_T) - \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})} \right]$$

$$\begin{aligned}
\nabla_\theta L &= -\mathbb{E}_{x_0, x_{1:T}} \left[\sum_{t=1}^{T} \nabla_\theta \log \left(\frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})} \right) \right] \\
&= -\mathbb{E}_{x_0, x_{1:T}} \left[\sum_{t=1}^{T} \nabla_\theta \left(\log p_\theta(x_{t-1} \mid x_t) - \log q(x_t \mid x_{t-1}) \right) \right] \\
&= -\mathbb{E}_{x_0, x_{1:T}} \left[\sum_{t=1}^{T} \nabla_\theta \log p_\theta(x_{t-1} \mid x_t) \right]
\end{aligned}$$

So to obtain this gradient, we could:

1. Sample x_0 from dataset
2. Run forward process to obtain $x_1, \ldots, x_T$, with some pre-determined variance schedule for $\beta_1, \ldots, \beta_T$.
3. For $t = 1, \ldots, T$, compute using automatic differentiation:

$$\begin{aligned}
-\nabla_\theta \log p_\theta(x_{t-1} \mid x_t) &= -\nabla_\theta \log \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \\
&= -\nabla_\theta \log \left((2\pi)^{-\frac{k}{2}} \det(\Sigma_\theta)^{-\frac{1}{2}} + \exp \left(-\frac{1}{2} (x_t - \mu_\theta)^T \Sigma_\theta^{-1} (x_t - \mu_\theta) \right) \right) \\
&= -\nabla_\theta \left(\log(2\pi)^{-\frac{k}{2}} + \log \det(\Sigma_\theta)^{-\frac{1}{2}} + \left(-\frac{1}{2} (x_t - \mu_\theta)^T \Sigma_\theta^{-1} (x_t - \mu_\theta) \right) \right) \\
&= -\nabla_\theta \left(\log \det(\Sigma_\theta)^{-\frac{1}{2}} + \left(-\frac{1}{2} (x_t - \mu_\theta)^T \Sigma_\theta^{-1} (x_t - \mu_\theta) \right) \right)
\end{aligned}$$

4. Repeat for a batch b samples of x_0, get an empirical average over samples to get $\nabla_\theta L$, and update $\theta_{t+1} = \theta - \alpha \nabla_\theta L$ (SGD, but can use other optimizers like Adam)

Questions remaining:

1. How do we parameterize $\mu_\theta(x_t, t)$?

2. How do we parameterize $\Sigma_\theta(x_t, t)$? Is it a full covariance matrix? Is it diagonal? Is it learned? Is it fixed?

3. Under this algorithm, training requires a full roll out of the forward process from x_0, which is quite slow and not easy to parallelize.