6.8300 Pset 5 Part2 writeup

Zhi Ren

April 16, 2025

1 Preliminary and math preparations

1.1 Diffusion and forward processes

Question 1. $q(x_t|x_0) =?$

Answer: We claim $q(x_t|x_0) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is a standard normal random variable and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ with $\alpha_i = 1 - \beta_i$. We prove this by induction: when $t = 1$, by assumption, we have $q(x_1|x_0) = \mathcal{N}(\sqrt{1-\beta_1}x_0, \beta_1 I)$, which follows a normal distribution. By the reparameterization trick, we can write $q(x_1|x_0) = \sqrt{1-\beta_1}x_0 + \sqrt{\beta_1}\epsilon_1$ for some $\epsilon_1 \sim \mathcal{N}(0, I)$. Thus the claim is true.

Suppose the claim is true for $t-1$. Again by the reparameterization trick, we have $q(x_t|x_{t-1}) = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_t$ for $\epsilon_t \sim \mathcal{N}(0, I)$. By the induction hypothesis, we can write $q(x_{t-1}|x_0) = \sqrt{\bar{\alpha}_{t-1}}x_0 + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon_{t-1}$ for some $\epsilon_{t-1} \sim \mathcal{N}(0, I)$. Thus we have $q(x_t|x_0) = \sqrt{1-\beta_t}(\sqrt{\bar{\alpha}_{t-1}}x_0 + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon_{t-1}) + \sqrt{\beta_t}\epsilon_t = \sqrt{(1-\beta_t)\bar{\alpha}_{t-1}}x_0 + \sqrt{(1-\beta_t)(1-\bar{\alpha}_{t-1})}\epsilon_{t-1} + \sqrt{\beta_t}\epsilon_t$.

Note $\sqrt{(1-\beta_t)\bar{\alpha}_{t-1}} = \sqrt{\bar{\alpha}_t}$. On the other hand, $\sqrt{(1-\beta_t)(1-\bar{\alpha}_{t-1})}\epsilon_{t-1} + \sqrt{\beta_t}\epsilon_t$ is again a normal random variable with mean zero and variance $(1-\beta_t)(1-\bar{\alpha}_{t-1}) + \beta_t = 1 - \bar{\alpha}_{t-1} - \beta_t + \beta_t\bar{\alpha}_{t-1} + \beta_t = 1 - \bar{\alpha}_{t-1} + \beta_t\bar{\alpha}_{t-1} = 1 - (\beta_t - 1)\bar{\alpha}_{t-1} = 1 - \bar{\alpha}_t$. Hence we complete the proof.

□

Question 2. $\lim_{T\to\infty} q(x_T) =?$

Answer: Since $\beta_t \in (0, 1)$ and $\liminf_{t\to\infty} \beta_t > c$, we can conclude $\bar{\alpha}_t \to 0$ and $1 - \bar{\alpha}_t \to 1$. Thus x_T converges to the standard normal random variable and $q(x_T) = \mathcal{N}(0, I)$. □

1.2 Reverse diffusion process

Question 3. $\tilde{\beta}_t =?$

Answer: Using conditional probabilities, we can write $q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1},x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)}$

$$
\begin{aligned}
q(x_{t-1}|x_t, x_0) &= \frac{q(x_t, x_{t-1}, x_0)}{q(x_t, x_0)} = \frac{q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)q(x_0)}{q(x_t|x_0)q(x_0)} \\
&= \frac{q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)} = \frac{q(x_t|x_{t-1})q(x_{t-1}|x_0)}{q(x_t|x_0)} \\
&\propto \exp\left(-\frac{1}{2}\frac{\|x_t - \sqrt{\alpha_t}x_{t-1}\|_2}{\beta_t} - \frac{1}{2}\frac{\|x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0\|_2^2}{1-\bar{\alpha}_{t-1}} + \frac{1}{2}\frac{\|x_t - \sqrt{\bar{\alpha}_t}x_0\|_2^2}{1-\bar{\alpha}_t}\right) \\
&\propto \exp\left(-\frac{-2\sqrt{\alpha_t}x_t^T x_{t-1} + \alpha_t\|x_{t-1}\|_2^2}{2\beta_t} - \frac{\|x_{t-1}\|_2^2 - 2\sqrt{\bar{\alpha}_{t-1}}x_{t-1}^T x_0}{2(1-\bar{\alpha}_{t-1})}\right) + C(x_0, x_t), \\
&= \exp\left(-\frac{1}{2}(\frac{1}{1-\bar{\alpha}_{t-1}} + \frac{\alpha_t}{\beta_t})\|x_{t-1}\|_2^2 + (\frac{\sqrt{\alpha_t}x_t^T}{\beta_t} + \frac{\sqrt{\bar{\alpha}_{t-1}}x_0^T}{1-\bar{\alpha}_{t-1}})x_{t-1}\right) + C(x_0, x_t)
\end{aligned}
$$

where $C(x_0, x_t)$ is some function that only depends on x_0, x_t.

Completing the square in the exponent, we get the variance $\tilde{\beta}_t = 1/(\frac{1}{1-\bar{\alpha}_{t-1}} + \frac{\alpha_t}{\beta_t})) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ □

Question 4. *Denote $\tilde{\mu}_t(x_t, x_0) = A \cdot x_t + B \cdot x_0$, then $A =$; $B =$*

Answer: From the previous problem, we can compute the mean by completing the square in the exponent, and we get

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

Thus $A = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$ and $B = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}$. □

Question 5. *Using the distribution derived in (1) and the reparameterization trick, express x_0 as a combination x_t and a Gaussian noise ϵ_t: $x_0 =$*

Answer: From Question 1, we have $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, from which we can write $x_0 = \frac{x_t}{\sqrt{\bar{\alpha}_t}} - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\epsilon$ □

Question 6. *Using the expression of x_0 in (5) to replace the x_0 in (4), $\tilde{\mu}_t$ can be written as a combination of x_t and ϵ_t: $\tilde{\mu}_t(x_t, \epsilon_t) = A' \cdot x_t + B' \cdot \epsilon_t$. Then $A' =$; $B' =$*

Answer: Putting in the expression for x_0, we have

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

collecting similar terms gives $A' = \frac{1}{\sqrt{\alpha_t}}$ and $B' = -\frac{\beta_t}{\sqrt{(\alpha_t)(1-\bar{\alpha}_t)}}$. □

1.3 Analyzing different variance schedules

The figures for the four different β_t schedules are presented below.

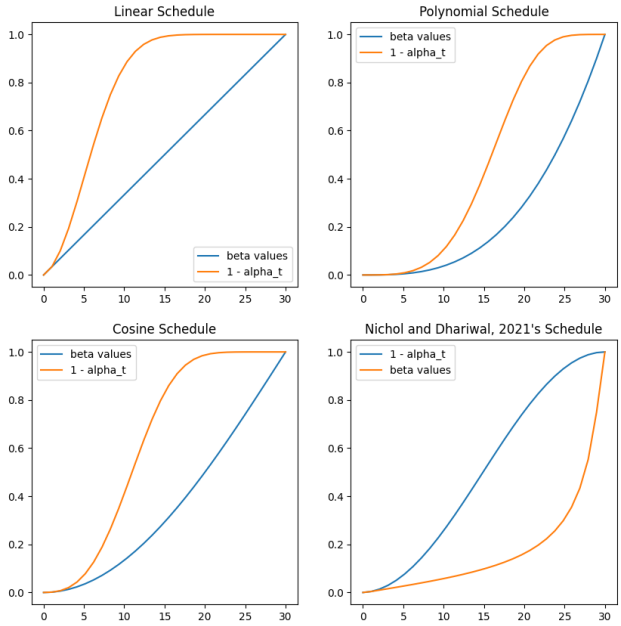


Figure 1: Graphs for different noise schedules

Question 7. *Comparing the schedules and based on the formula of $q(x_t|x_0)$ in (1), what can you say about their impact to the diffusion training?*

Answer: For all these schedules, $1 - \bar{\alpha}_t$ and β_t converges to one, which means when we start from target distribution q_0, we eventually arrive at a standard normal at $t = T$. In each intermediate step, we model $q(x_t|x_{t-1}, x_0)$ as a normal random variable with mean and variance controlled by β_t. For the first three schedules, β_t follows a simple law that controls the rate at which we learn the distributions. (The more noise we add in one step, the less information there is to be learned in that step). □

Question 8. *Based on the observations, [Nichol and Dhariwal, 2021] proposed to construct a different noise schedule in terms of $\bar{\alpha}_t$:*

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, f(t) = \cos(\frac{t/T + s}{1 + s} \cdot \frac{\pi}{2})^2$$

For this schedule, generate plots with y-axis being the value of β_t and $1 - \bar{\alpha}_t$ and x-axis being the diffusion step t/T, and use $s = 0.008$ and $T = 30$. How does this schedule differ from the previous two, and why might it lead to improved performance in the trained diffusion models?

Answer: The difference between this schedule and the first three is that β_t increases very slowly at the beginning until it hits some transition point, after which it then increases very fast. The advantage of such a schedule is that at the beginning, there is a large amount

of information to be learned from q_0 so it makes sense to go slower in each step and inject less noise. After fine-grained details of the distributions have been learned, we can go faster towards the destination (which is pure noise). □

2 Training Diffusion Models on a Toy Dataset

We include the generated plots for training/testing losses and samples with different steps below.

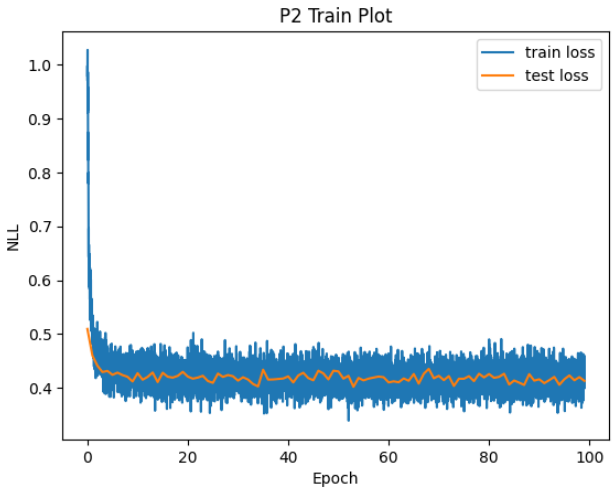


Figure 2: Training and testing losses vs epochs

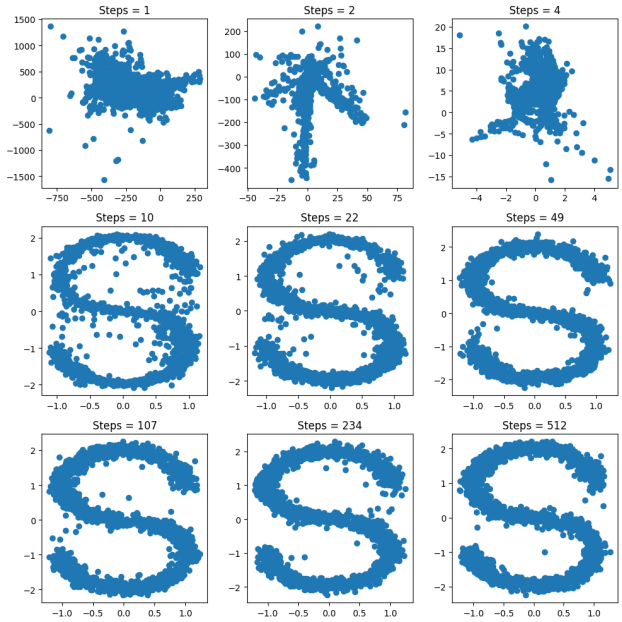


Figure 3: Samples generated using diffusion models with different steps

3 MNIST and Conditional Generation

Question 9. *Please explain each components of the architecture above (each one of FourierEncoder, ResidualLayer, Encoder, Decoder, or Midcoder) in your own words, (1) their role in the U-Net, (2) their inputs and outputs, and (3) a brief description of how the inputs turn into outputs.*

Answer: The Fourier encoder maps time to some high-dimensional space by taking the corresponding Fourier features. In the native implementation, the time variable is concatenated with the space variable, but with the Fourier encoder, the model can be more expressive in terms of the time variable. The input is time t and it maps to a dim-dimensional space, where dim is a hyperparameter.

The ResNet layer is a very fundamental building block in modern deep learning: it helps reduce the vanishing gradient problem and enhances stabilities in training. It takes a space variable x (image features), time and label embeddings. The output is the result of passing through the residual block, added to the original x.

The encoder acts as part of the U-Net "encoder" path. It processes incoming feature maps with multiple residual blocks and then downsamples the spatial resolution.

The midcoder works at the bottleneck of the U-Net architecture. It continues to process the feature maps at the reduced resolution without further altering the spatial dimensions.

The decoder provides the decoding (up-sampling) path of the U-Net: it increases the spatial resolution and fuses it with skip connections from the encoder.

□

The training/testing losses and the generated hand-written digits are presented below.

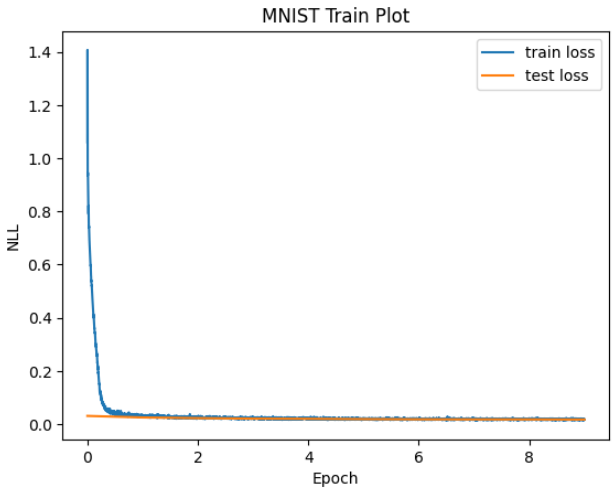
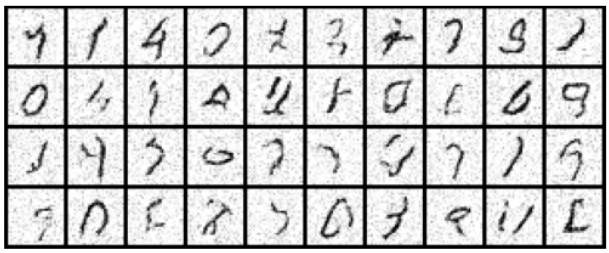
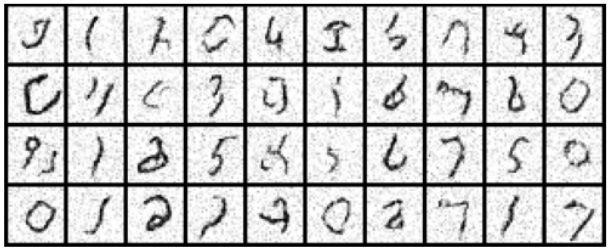


Figure 4: MNIST training and testing losses over epoch

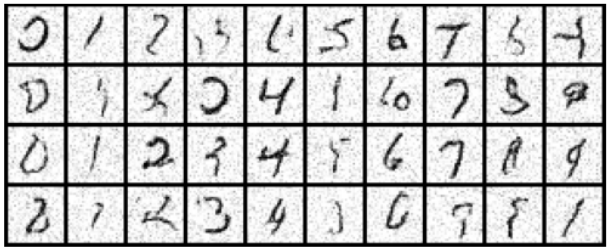
w=0.0



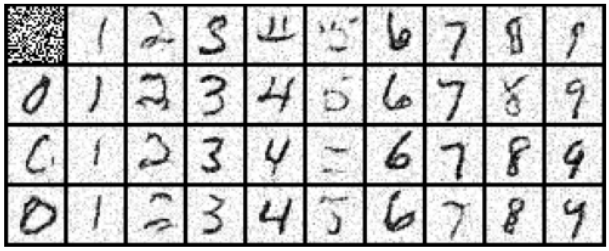
w=0.5



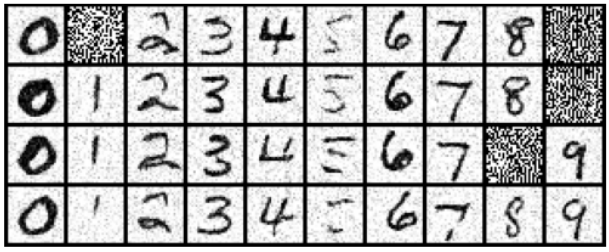
w=1.0



w=2.0



w=4.0



Question 10. *Comparing the results with different w values, what can you say about its impact to the generation performance?*

Answer: There is a clear pattern that when the weight w increases, hand-written digits sampled will be more accurate. This makes sense because $w = 0$ corresponds to unguided generation and as w increases, we incorporate in more information about the labels. □

4 Sampling based on pre-trained models

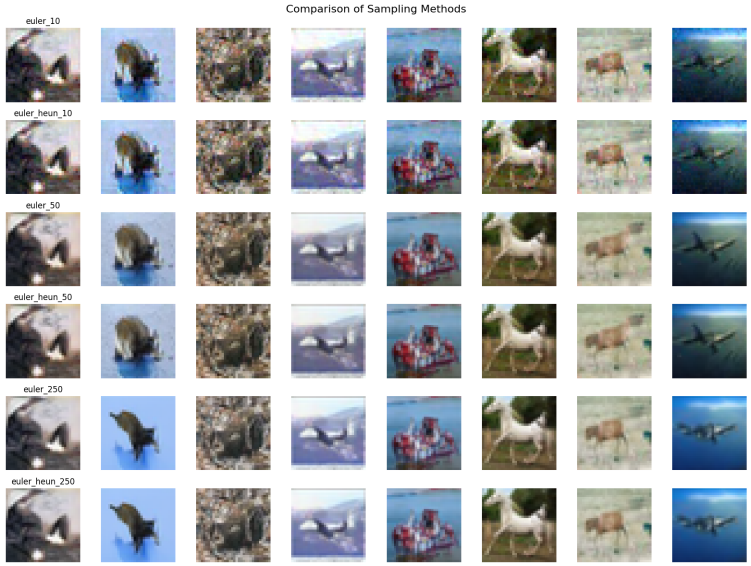


Figure 5: Comparison between different ODE samplers

Question 11. *Complete the Euler Heun sampler function and evaluate it using the given comparison code. What can you conclude based on the generated images using different samplers and number of steps?*

Answer: The generated pictures become clearer and more accurate as number of steps increases. Compared to Euler scheme, Heun correction makes the pictures smoother. □