Fondasi Matematika dari Denoising Diffusion Probabilistic Models

Dimas Tri Kurniawan

dimas.tri01@ui.ac.id

Abstract

Denoising Diffusion Probabilistic Models (DDPM) merupakan salah satu pendekatan terbaru dalam generasi citra yang telah menunjukkan hasil yang mengesankan dalam menghasilkan citra berkualitas tinggi. Model ini didasarkan pada prinsip difusi terbalik, yaitu mengubah distribusi data asli menjadi distribusi noise, lalu memulihkannya kembali ke data asli melalui proses denoising bertahap. Dalam paper ini, penulis ingin mengulas lebih dalam tentang fondasi matematika dari DDPM. Penulis menganggap bahwa masih terdapat kekurangan dalam penjelasan dari fondasi matematika yang ada pada paper DDPM. Oleh karena itu, penulis tertarik untuk mengulas berbagai rumus yang digunakan sebagai fondasi dari paper tersebut.

1 Pendahuluan

Denoising Diffusion Probabilistic Models (yang selanjutnya kita sebut Diffusion Models, untuk mempersingkat) telah muncul sebagai salah satu metode terbaru dan paling menjanjikan dalam generasi citra dan data berbasis probabilistik. Sejak pertama kali diperkenalkan, DDPM telah menunjukkan kemajuan yang signifikan dalam kualitas gambar yang dihasilkan, melampaui metode generatif lain seperti Generative Adversarial Networks (GANs) dalam hal stabilitas dan kemampuan untuk menghasilkan citra dengan detail yang sangat tinggi. Metode ini beroperasi melalui proses difusi, yang mengubah data asli menjadi noise secara bertahap, kemudian berusaha memulihkan data asli dari noise tersebut menggunakan model probabilistik. Pendekatan ini menawarkan cara yang baru dan menarik untuk memahami bagaimana model dapat mengatasi tantangan dalam generasi data, terutama dalam konteks pengolahan citra.

Meskipun hasil empiris DDPM sangat menjanjikan, sebagian besar literatur yang ada cenderung fokus pada implementasi algoritmik dan teknik optimisasi tanpa memberikan penjelasan yang mendalam mengenai fondasi matematis dari model ini. Padahal, pemahaman yang lebih mendalam tentang aspek matematis DDPM sangat penting untuk meningkatkan pemahaman kita tentang bagaimana model ini berfungsi, serta untuk membuka potensi aplikasi lebih lanjut dalam berbagai domain, seperti pemodelan distribusi data dan rekonstruksi citra.

Paper ini bertujuan untuk mengisi kekosongan tersebut dengan menjelaskan secara rinci dasar-dasar matematis yang mendasari Denoising Diffusion Probabilistic Models. Kami akan membahas bagaimana proses difusi terbalik bekerja dalam konteks probabilistik, serta menguraikan teori probabilitas yang relevan, termasuk peran distribusi noise dan pembelajaran berbasis optimisasi dalam meningkatkan kinerja model. Selain itu, kami juga akan menjelaskan tantangan utama yang dihadapi dalam penerapan DDPM, serta potensi arah pengembangan selanjutnya untuk model ini.

Dengan memberikan wawasan yang lebih mendalam mengenai dasar-dasar matematis dari DDPM, diharapkan pembaca dapat memperoleh pemahaman yang lebih komprehensif tentang bagaimana model ini bekerja dan bagaimana potensi teknologinya dapat dimanfaatkan di masa depan.

Preprint. Under review.

2 Latar Belakang

Diffusion models terdiri dari 2 proses utama, yaitu Forward/Diffusion Process dan Reverse Diffusion Process.

2.1 Forward Process

Forward process/diffusion process adalah proses di mana kita menambahkan noise secara berkala ke sebuah gambar sampai gambar tersebut konvergen ke distribusi gaussian dengan mean 0 dan variance 1. Dengan kata lain, gambar tersebut akan menjadi pure noise setelah t langkah. Forward process didefinisikan dengan Markov chain yang menambahkan Gaussian noise secara berkala:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) \coloneqq \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{1}$$

Persamaan (1) bermaksud bahwa kita ingin mencari distribusi gambar yang telah diberikan noise $\mathbf{x}_1$ sampai $\mathbf{x}_T$ jika diketahui gambar asli $\mathbf{x}_0$. Karena menggunakan asumsi Markov, gambar/state sebelumnya tidak diperlukan sehingga persamaan yang seharusnya:

$q(\mathbf{x}_{1:T}|\mathbf{x}_0) \coloneqq \frac{q(\mathbf{x}_T|\mathbf{x}_{T-1},\mathbf{x}_{T-2},\mathbf{x}_{T-3},...,\mathbf{x}_0)\ q(\mathbf{x}_{T-1}|\mathbf{x}_{T-2},\mathbf{x}_{T-3},\mathbf{x}_{T-4},...,\mathbf{x}_0)\ \cdots\ q(\mathbf{x}_0)}{q(\mathbf{x}_0)}$,

dapat kita sederhanakan dengan menghilangkan state sebelumnya:

$$\coloneqq \frac{q(\mathbf{x}_T|\mathbf{x}_{T-1},\cancel{\mathbf{x}_{T-2}},\cancel{\mathbf{x}_{T-3}},\cancel{...},\cancel{\mathbf{x}_0})\ q(\mathbf{x}_{T-1}|\mathbf{x}_{T-2},\cancel{\mathbf{x}_{T-3}},\cancel{\mathbf{x}_{T-4}},\cancel{...},\cancel{\mathbf{x}_0})\ \cdots\ \cancel{q(\mathbf{x}_0)}}{\cancel{q(\mathbf{x}_0)}}$$

menjadi:

$$\coloneqq \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

Forward process menambahkan Gaussian noise secara berkala berdasarkan linear variance schedule $\beta_1, \beta_2, \beta_3, ..., \beta_t$:

$q(\mathbf{x}_t|\mathbf{x}_{t-1}) \coloneqq \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$,

dengan $\mathbf{x}_t$ adalah output, $\sqrt{1-\beta_t}$ mean, dan β_t variance dari $\mathbf{x}_t$. Melalui reparameterization trick $\mathcal{N}(\mu, \sigma^2) = \mu + \sigma\epsilon$, persamaan di atas dapat ditulis kembali menjadi:

$$\coloneqq \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}.$$

Didefinisikan $\alpha_t \coloneqq 1 - \beta_t$, maka persamaan di atas menjadi:

$$\coloneqq \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}.$$

Subtitusi $\mathbf{x}_{t-1}$:

$$\coloneqq \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-2}) + \sqrt{1-\alpha_t}\epsilon_{t-1}.$$

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1}.$$

Jika dua 2 gaussian noise $\mathcal{N}(\mu, \sigma_1^2)$ dan $\mathcal{N}(\mu, \sigma_2^2)$ ditambahkan, maka hasilnya yaitu $\mathcal{N}(\mu, (\sigma_1^2 + \sigma_2^2))$. Ubah 2 term terakhir:

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1} + 1 - \alpha_t}\epsilon.$$

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_t\alpha_{t-1}}\epsilon.$$

Subtitusi $\mathbf{x}_{t-2}$:

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}}\mathbf{x}_{t-3} + \sqrt{1 - \alpha_t\alpha_{t-1}\alpha_{t-2}}\epsilon.$$

Subtitusi $\mathbf{x}_{t-3}$:

$$\coloneqq \sqrt{\alpha_t \alpha_{t-1} \alpha_{t-2} \alpha_{t-3}} \mathbf{x}_{t-4} + \sqrt{1 - \alpha_t \alpha_{t-1} \alpha_{t-2} \alpha_{t-3}} \epsilon.$$

Jika kita terus subtitusi sampai dengan $\mathbf{x}_1$:

$$\coloneqq \sqrt{\alpha_t \alpha_{t-1} \alpha_{t-2} \alpha_{t-3} ... \alpha_1} \mathbf{x}_0 + \sqrt{1 - \alpha_t \alpha_{t-1} \alpha_{t-2} \alpha_{t-3} ... \alpha_1} \epsilon.$$

Diberikan definisi $\bar{\alpha}_t \coloneqq \prod_{t=1}^{T} \alpha_t$, maka:

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) \coloneqq \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon. \tag{2}$$

Penyederhaan $q(\mathbf{x}_t | \mathbf{x}_{t-1})$ menjadi persamaan (2) membuat rumus menjadi lebih mudah diselesaikan. Daripada melakukan iterasi dari $\mathbf{x}_T$ sampai $\mathbf{x}_{t-1}$, kita cukup menentukan $\bar{\alpha}_t$, lalu subtitusi ke persamaan (2).

2.2 Reverse Process

Diffusion models adalah sebuah model latent variable dalam bentuk $p_\theta(\mathbf{x}_0) \coloneqq \int p_\theta(\mathbf{x}_{0:T}) \, d_{\mathbf{x}_{1:T}}$, di mana $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_T$ adalah latents dengan dimentionality yang sama dengan data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. Joint distribution $p_\theta(\mathbf{x}_{0:T})$ adalah reverse process:

$$p_\theta(\mathbf{x}_{0:T}) \coloneqq p_\theta(\mathbf{x}_0 | \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_T) \, p_\theta(\mathbf{x}_1 | \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, ..., \mathbf{x}_T) \, ... \, p(\mathbf{x}_T).$$

Sama seperti forward process, reverse process didefinisikan sebagai Markov chain:

$$\coloneqq p_\theta(\mathbf{x}_0 | \mathbf{x}_1, \cancel{\mathbf{x}_2}, \cancel{\mathbf{x}_3}, \cancel{...}, \cancel{\mathbf{x}_T}) \, p_\theta(\mathbf{x}_1 | \mathbf{x}_2, \cancel{\mathbf{x}_3}, \cancel{\mathbf{x}_4}, \cancel{...}, \cancel{\mathbf{x}_T}) \, ... \, p(\mathbf{x}_T),$$

sehingga kita mendapatkan persamaan akhir:

$$p_\theta(\mathbf{x}_{0:T}) \coloneqq p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t). \tag{3}$$

Sejauh ini, kita tidak tahu distribusi asli dari reverse process. Oleh karena itu, kita menggunakan aproksimasi p_θ yang nantinya akan dipelajari oleh neural network. Khusus untuk $\mathbf{x}_T$, kita tidak menggunakan aproksimasi p_θ karena kita tahu nilai asli dari $\mathbf{x}_T$.

Reverse process merupakan Gaussian transition dengan μ dan σ^2 yang akan dipelajari selanjutnya, dimulai dari $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; 0, \mathbf{I})$, yaitu pure noise dengan mean 0 dan variance 1 (matriks identitas):

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) \coloneqq \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)). \tag{4}$$

Namun, author dari paper DDPM memutuskan untuk memberikan nilai yang pasti pada σ^2 berdasarkan linear schedule. Dengan kata lain, σ tidak perlu dipelajari pada reverse process.

3 Loss Function

Kita tidak mengetahui distribusi dari reverse process. Untuk itu, kita akan melatih model agar dapat melakukan aproksimasi distribusi dari p sebaik mungkin. Caranya cukup mirip pada paper Variational Autoencoder (VAE). Pada paper tersebut, kita tidak tahu distribusi asli dari x, diberikan distribusi Z. Untuk itu, kita akan mengaproksimasinya melalui neural network. Di sini, kita ingin neural network agar dapat mengaproksimasi probability dari p of **x** sehingga dapat generate gambar yang semirip mungkin dengan distribusi **x** (training data). Untuk itu, kita akan memaksimalkan log-likelihood dari p. Fungsi log dibutuhkan agar nilai likelihood tidak exploding (terlalu besar hingga tidak bisa disimpan komputer). Fungsi log bersifat monotonic sehingga memaksimalkan log-likelihood berarti

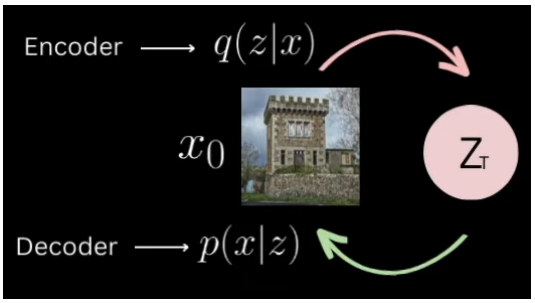

Figure 1: aproksimasi x pada VAE

juga memaksimalkan true likelihood.

$$\begin{aligned}\log p(x) &= \log \int p(x,z)\,dz. \\ &= \log \int p(x,z)\frac{q(z|x)}{q(z|x)}\,dz \\ &= \log \mathbb{E}_{q(z|x)}\left[\frac{p(x,z)}{q(z|x)}\right]\end{aligned}$$

Logaritma merupakan fungsi concave. Hal ini dapat dibuktikan karena turunan kedua dari logaritma adalah negatif. Oleh sebab itu, berlaku: $\log \mathbb{E}_{q(z|x)}\left[\frac{p(x,z)}{q(z|x)}\right] = \mathbb{E}_{q(z|x)}\left[\log \frac{p(x,z)}{q(z|x)}\right]$, sehingga:

$$\log p(x) \geq \mathbb{E}_{q(z|x)}\left[\log \frac{p(x,z)}{q(z|x)}\right]$$

Kita dapat menerapkan hal yang sama pada diffusion models. Perbedaannya, kita tidak ke Z_T, tetapi melalui urutan latent variable dari x_1 ke x_T. Kita juga tidak lagi mencari loss function, melainkan ekspektasi dari loss function. Disamping itu, kita tidak memaksimalkan log-likelihood,

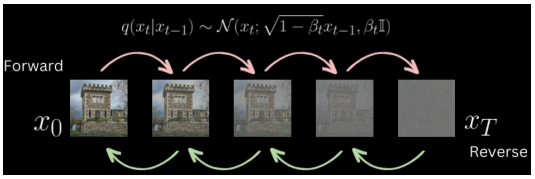

Figure 2: reverse process

melainkan meminimalkan negative log-likelihood. Sebenarnya, kedua hal tersebut sama. Namun, optimizer biasanya meminimalkan fungsi sehingga akan lebih cocok jika kita menggunakan negative log-likelihood. Seperti namanya, yaitu loss function, kita ingin meminimalkan fungsi ini sekecil mungkin.

$$\begin{aligned}-\log p_\theta(\mathbf{x}_0) &= -\log \int p_\theta(\mathbf{x}_{0:T})\,d\mathbf{x}_{1:T}. \\ &= -\log \int p_\theta(\mathbf{x}_{0:T})\frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\,d\mathbf{x}_{1:T} \\ &= -\log \int q(\mathbf{x}_{1:T}|\mathbf{x}_0)\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\,d\mathbf{x}_{1:T} \\ &= -\log \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] \\ &\leq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] \\ &\leq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right]\end{aligned}$$

Selanjutnya, kita dapat melakukan optimasi pada term yang berada di dalam ekspektasi.

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}$$

$q(\mathbf{x}_t|\mathbf{x}_{t-1})$ dapat kita ubah menjadi $\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t)\ q(\mathbf{x}_t)}{q(\mathbf{x}_{t-1})}$. Namun, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ akan memiliki variance yang tinggi karena kandidat untuk $\mathbf{x}_t$ bisa bermacam-macam.

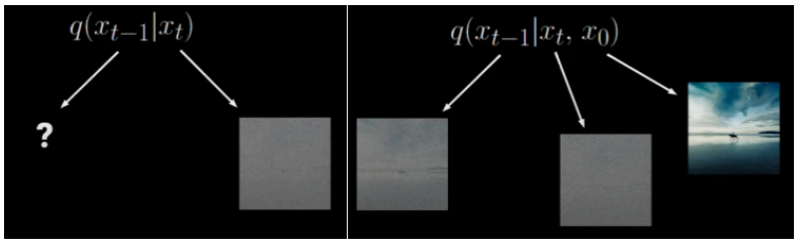

Figure 3: high variance vs. low variance

Oleh karena itu, kita akan menambahkan gambar asli $\mathbf{x}_0$ agar kandidat untuk $\mathbf{x}_{t-1}$ dapat menyesuaikan gambar aslinya.

$$= \log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}{p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}$$

Namun, terdapat hal yang janggal. Jika kita teliti lebih dalam, term pertama pada numerator mengandung self-loop! Lebih tepatnya, $q(\mathbf{x}_1|\mathbf{x}_0, \mathbf{x}_0) = \frac{q(\mathbf{x}_0|\mathbf{x}_1, \mathbf{x}_0)\ q(\mathbf{x}_1|\mathbf{x}_0)}{q(\mathbf{x}_0|\mathbf{x}_0)}$. Untuk mengatasi hal ini, keluarkan term pertama dari product series. Setelah itu, kita baru bisa menambahkan $\mathbf{x}_0$ pada conditional probability di product series.

$$= \log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}{p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}$$

$$= \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)\prod_{t=2}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}{p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}$$

$$= \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)\prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\ q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)\ p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}$$

$$= \log \frac{\cancel{q(\mathbf{x}_1|\mathbf{x}_0)}\ q(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{x}_0)\ q(\mathbf{x}_T|\mathbf{x}_0)\ q(\mathbf{x}_{T-2}|\mathbf{x}_{T-1},\mathbf{x}_0)\ \cancel{q(\mathbf{x}_{T-1}|\mathbf{x}_0)}\ \dots\ q(\mathbf{x}_1|\mathbf{x}_2,\mathbf{x}_0)\ \cancel{q(\mathbf{x}_2|\mathbf{x}_0)}}{\cancel{q(\mathbf{x}_{T-1}|\mathbf{x}_0)}\ q(\mathbf{x}_{T-2}|\mathbf{x}_0)\ \dots\ \cancel{q(\mathbf{x}_2|\mathbf{x}_0)}\ \cancel{q(\mathbf{x}_1|\mathbf{x}_0)}\ p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}$$

$$= \log \frac{q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}$$

$$= \log \frac{q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p(\mathbf{x}_T)\ p_\theta(\mathbf{x}_0|\mathbf{x}_1) \prod_{t=2}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}$$

$$= \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$$

Masukkan kembali $\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}$ pada loss function:

$$-\log p_\theta(\mathbf{x}_0) \leq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right].$$

Ubah setiap term pada loss function ke bentuk KL (Kullback–Leibler divergence) divergence:

$$\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right]$$
$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)) + \sum_{t=2}^{T} D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right]$$

Didapatkan:

$$L = \mathbb{E}_q \left[D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)) + \sum_{t=2}^{T} D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \quad (5)$$

dengan

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}), \quad (6)$$

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) \coloneqq \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t, \quad (7)$$

$$\tilde{\beta}_t \coloneqq \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}}\beta_t. \quad (8)$$

Kita dapat menguraikan L menjadi:

$$L_T = D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)). \quad (9)$$

$$L_{t-1} = D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)). \quad (10)$$

$$L_0 = \log p_\theta(\mathbf{x}_0|\mathbf{x}_1). \tag{11}$$

3.1 Forward process dan L_T

Sebenarnya, β_t dapat dipelajari melalui reparameterization. Namun, nilai β_t akan dibuat tetap menjadi konstanta sehingga q tidak ada parameter yang dapat dipelajari. Hal ini membuat L_T menjadi konstan selama training dan dapat diabaikan.

3.2 Reverse process dan $L_{1:T-1}$

Kita ingin mempelajari mean dan variance dari reverse diffusion process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$, dengan $1 < t < T$. Di sini, author memutuskan untuk tidak mempelajari variance dan menjadikannya konstanta:

$$\Sigma_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I} \tag{12}$$

sehingga

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I}) \tag{13}$$

Untuk merepresentasikan mean $\mu_\theta(\mathbf{x}_t, t)$, penulis DDPM melakukan parameterisasi berdasarkan analisis terhadap L_t, yaitu $D_{KL}(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}))$, dengan menggunakan mean square error:

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)\|^2 \right] + C, \tag{14}$$

dengan C adalah konstanta yang tidak bergantung pada θ. Subtitusi $\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0)$ pada (12):

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \| \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{x}_t - \mu_\theta(\mathbf{x}_t, t) \|^2 \right] + C,$$

Pada persamaan (2), kita telah diperlihatkan bagaimana mencari $\mathbf{x}_t$ jika diberikan $\mathbf{x}_0$. Persamaan tersebut dapat kita gunakan untuk mencari $\mathbf{x}_0$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$
$$\iff \mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon)$$

Subtitusi $\mathbf{x}_0$ pada (14):

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

Penulis DDPM mereparameterisasi x_t agar dapat memprediksi noise pada reverse diffusion process:

$$= \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \| \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t(\mathbf{x}_0, \epsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon) - \mu_\theta(\mathbf{x}_t(\mathbf{x}_0, \epsilon), t) \|^2 \right] + C, \tag{15}$$

dengan $\mathbf{x}_t(\mathbf{x}_0, \epsilon) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$. Persamaan (10) memperlihatkan bahwa μ_θ harus memprediksi $\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t(\mathbf{x}_0, \epsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon)$, diberikan **t**. Karena **t** sudah ada sebagai input, kita

dapat melakukan parameterisasi sebagai berikut:

$$\begin{aligned}\mu_\theta(\mathbf{x}_t, t) &= \bar{\mu}_t(\mathbf{x}_t, \mathbf{x}_0((x)_t, \epsilon_\theta)) \\ &= \bar{\mu}_t(\mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t))) \\ &= \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)) \end{aligned} \tag{16}$$

Subtitusi persamaan (16) pada persamaan (15):

$$\begin{aligned}L_{t-1} &= \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon) - \mu_\theta(\mathbf{x}_t, t)\|^2\right] + C \\ &= \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon) - \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t))\|^2\right] + C \\ &= \mathbb{E}_q\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right] + C \end{aligned} \tag{17}$$

Setelah melakukan percobaan, ternyata penulis DDPM berpendapat bahwa menghilangkan term pertama pada (17) pada proses training akan meningkatkan kualitas sample dan juga lebih mudah dalam implementasi:

$$L_{simple}(\theta) \coloneqq \mathbb{E}_q\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right] \tag{18}$$

Recall $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$, kita dapat mensample $\mathbf{x}_{t-1}$ melalui reparameterization trick $\mathcal{N}(\mu, \sigma^2) = \mu + \sigma\epsilon$:

$$\begin{aligned}\mathbf{x}_{t-1} &= \mu_\theta(\mathbf{x}_t, t) + \Sigma_\theta(\mathbf{x}_t, t) + \epsilon \\ &= \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)) + \sqrt{\beta_t}\epsilon \end{aligned} \tag{19}$$

4 Data Scaling