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

Preprint. Under review.

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.

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:

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

dapat kita sederhanakan dengan menghilangkan state sebelumnya:

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

menjadi:

$$\coloneqq \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{4}$$

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

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) \coloneqq \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{5}$$

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:

$$\coloneqq \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}. \tag{6}$$

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

$$\coloneqq \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}. \tag{7}$$

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

$$\coloneqq \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-2}) + \sqrt{1-\alpha_t}\epsilon_{t-1} \tag{8}$$

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1} \tag{9}$$

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:

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1} + 1 - \alpha_t}\epsilon \tag{10}$$

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_t\alpha_{t-1}}\epsilon \tag{11}$$

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

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}}\mathbf{x}_{t-3} + \sqrt{1 - \alpha_t\alpha_{t-1}\alpha_{t-2}}\epsilon. \tag{12}$$

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

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}\alpha_{t-3}}\mathbf{x}_{t-4} + \sqrt{1 - \alpha_t\alpha_{t-1}\alpha_{t-2}\alpha_{t-3}}\epsilon \tag{13}$$

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

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}\alpha_{t-3}...\alpha_1}\mathbf{x}_0 + \sqrt{1 - \alpha_t\alpha_{t-1}\alpha_{t-2}\alpha_{t-3}...\alpha_1}\epsilon \tag{14}$$

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

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) \coloneqq \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon \tag{15}$$

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:

$$p_\theta(\mathbf{x}_{0:T}) \coloneqq p_\theta(\mathbf{x}_0|\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_T)\ p_\theta(\mathbf{x}_1|\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, ..., \mathbf{x}_T)\ ...\ p(\mathbf{x}_T) \tag{16}$$

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

$$\coloneqq p_\theta(\mathbf{x}_0|\mathbf{x}_1, \cancel{\mathbf{x}_2}, \cancel{\mathbf{x}_3}, \cancel{...}, \cancel{\mathbf{x}_T})\ p_\theta(\mathbf{x}_1|\mathbf{x}_2, \cancel{\mathbf{x}_3}, \cancel{\mathbf{x}_4}, \cancel{...}, \cancel{\mathbf{x}_T})\ ...\ p(\mathbf{x}_T), \tag{17}$$

sehingga kita mendapatkan persamaan akhir:

$$p_\theta(\mathbf{x}_{0:T}) \coloneqq p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t). \tag{18}$$

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):

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \coloneqq \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \tag{19}$$

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. Ternyata, kita dapat mencari lower bound menggunakan cara yang sama pada Variational Autoencoder (VAE). Untuk itu, kita akan mempelajari apa itu Latent Variable, Deep Latent Variable Models, VAE, dan apa koneksinya dengan DDPM.

3.1 Latent Variable

Latent variable adalah variable yang merupakan bagian dari model, tetapi tidak dapat kita observasi karena sifatnya yang hidden/tersembunyi. Oleh karena itu, latent

variable bukan bagian dari dataset. Biasanya, kita mendefinisikan **z** sebagai latent variable dan **x** sebagai observed variable. Distribusi marginal dari observed variables $p_\theta(x)$, yaitu:

$$p_\theta(x) = \int p_\theta(x, z) dz \tag{20}$$

3.2 Deep Latent Variable Models

Deep Latent Variable Models (DLVM) adalah latent variable model $p_\theta(x, y)$ yang distribusinya terparameterisasi oleh neural network. DVLM yang sering dijumpai (dan mungkin yang paling mudah) yaitu:

$$p_\theta(x, z) = p_\theta(z)\, p_\theta(x|z) \tag{21}$$

3.3 Intractabilities

Kendala utama untuk mencari $p_\theta(z|x)$ yaitu baik $p_\theta(x, z)$ dan $p_\theta(x)$ intractable dalam DLVM sehingga $p_\theta(z|x)$ menjadi intractable. Oleh karena itu, dibutuhkan teknik approximate inference agar kita dapat mengaproksimasi posterior $p_\theta(z|x)$. Untuk mengubah posterior inference yang intractable pada DVLM menjadi trackable, kita akan menggunakan sebuah parametric inference model $q_\phi(z|x)$. Model ini dikenal juga dengan sebutan encoder atau recognition model. Kita menggunakan ϕ sebagai sebuah parameter dari inference model tersebut dan dapat kita sebut juga variational parameters. Selanjutnya, kita akan mengoptimasi variational parameters ϕ agar:

$$q_\phi(z|x) \approx p_\theta(z|x) \tag{22}$$

Kedepannya, aproksimasi terhadap posterior ini akan membantu kita dalam mengoptimasi marginal likelihood.

3.4 VAE

VAE adalah salah satu contoh dari DLVM. VAE menggunakan deep neural networks untuk memparameterisasi distribusi probabilitas yang mendefinisikan latent variable model.

Perhatikan DVLM berikut:

$$p_\theta(x) = \int p_\theta(x, z)\, dz \tag{23}$$

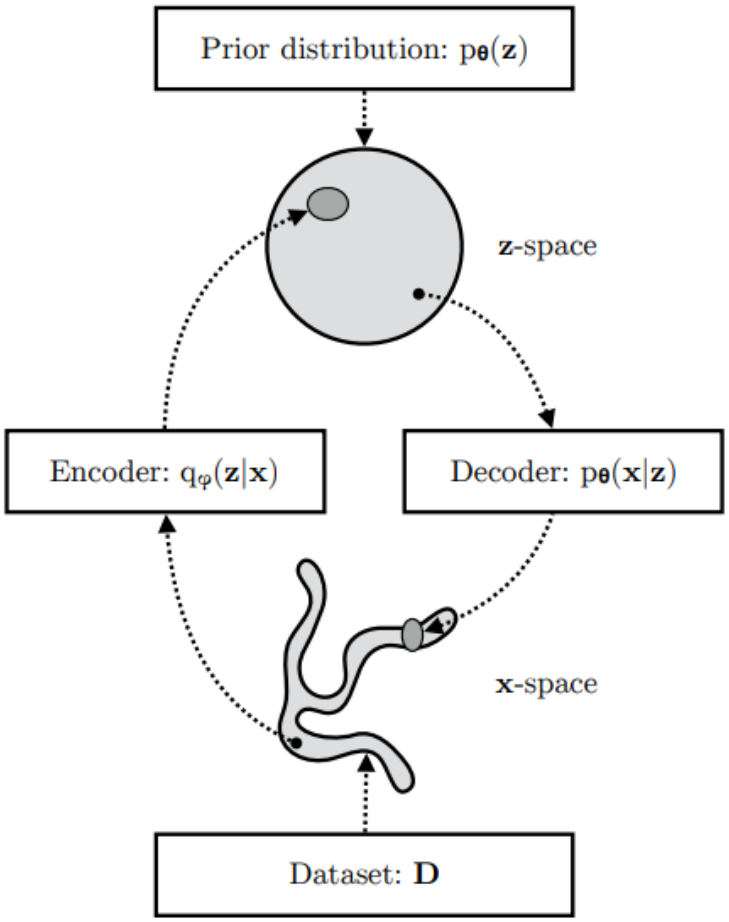

Figure 1: VAE mempelajari stochastic mapping antara observed x-space yang distribusi empirisnya $q_D(x)$ biasanya rumit dan sebuah latent z-spaceyang distribusinya relatif mudah (misalnya bola, seperti pada gambar). Genarative model mempelajari joint distribution $p_\theta(x, z)$ yang seringkali (tapi tidak selalu) difaktorkan menjadi $p_\theta(x, z) = p_\theta(z)\ p_\theta(x|z)$, dengan sebuah prior distribution over latent space $p_\theta(z)$, dan sebuah stochastic decoder $p_\theta(x|z)$. Stochastic encoder tersebut, $q_\phi(z|x)$, disebut juga inference model, mengaproksimasi true intractable posterior $p_\theta(z|x)$ dari generative model.

Untuk sembarang inference model $q_\phi(z|x)$ dan juga sembarang variational parameters ϕ:

$$p_\theta(x) = \int p_\theta(x, z) \frac{q_\phi(z|x)}{q_\phi(z|x)}\, dz \tag{24}$$

Karena $q_\phi(z|x) \approx p_\theta(z|x)$:

$$p_\theta(x) = \int p_\theta(x, z) \frac{q_\phi(z|x)}{p_\theta(z|x)}\, dz \tag{25}$$

$$p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}\left[\frac{p_\theta(x,z)}{p_\theta(z|x)}\right] \tag{26}$$

$$\log p_\theta(x) = \log \mathbb{E}_{q_\phi(z|x)}\left[\frac{p_\theta(x,z)}{p_\theta(z|x)}\right] \tag{27}$$

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

$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{p_\theta(z|x)}\right] \tag{28}$$

$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\frac{q_\phi(z|x)}{p_\theta(z|x)}\right]\right] \tag{29}$$

$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right] \tag{30}$$

Term kedua pada persamaan (30) merupakan Kullback-Leibler (KL) divergence antara $q_\phi(z|x)$ dan $p_\theta(z|x)$ yang sifatnya non-negative:

$$D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x)) \geq 0 \tag{31}$$

dan bernilai 0 jika dan hanya jika $q_\phi(z|x)$ sama dengan distribusi true posterior. Term pertama pada persamaan (30) adalah variational lower bound atau bisa juga disebut dengan evidence lower bound (ELBO):

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] \tag{32}$$

Karena KL divergence bersifat non-negatif, maka ELBO adalah lower bound dari log-likelihood dari data.

$$\mathcal{L}_{\theta,\phi}(x) = \log p_\theta(x) - D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x)) \tag{33}$$

$$\mathcal{L}_{\theta,\phi}(x) \leq \log p_\theta(x), \tag{34}$$

sehingga persamaan (30) dapat kita sederhanakan:

$$\log p_\theta(x) = \mathcal{L}_{\theta,\phi}(x) + D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x)) \tag{35}$$

$$\log p_\theta(x) \leq \log p_\theta(x) + D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x)) \tag{36}$$

3.5 DDPM

Kita dapat mengoptimasi log-likelihood menggunakan cara yang mirip pada VAE. Perhatikan DLVM berikut:

$$\log p_\theta(x) = \log \int p_\theta(x, z)\, dz \tag{37}$$

$$\log p_\theta(x) = \log \int p_\theta(x, z) \frac{q(z|x)}{q(z|x)}\, dz \tag{38}$$

$$\log p_\theta(x) = \log \mathbb{E}_{q(z|x)} \left[\frac{p_\theta(x, z)}{q(z|x)} \right] \tag{39}$$

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

$$\log p_\theta(x) \geq \mathbb{E}_{q(z|x)} \left[\log \frac{p_\theta(x, z)}{q(z|x)} \right] \tag{40}$$

Kita dapat menerapkan hal yang sama pada diffusion models. Perbedaannya, latent variable yang kita gunakan yaitu $\mathbf{x}_{1:T}$ dan juga observed variable $\mathbf{x}_0$. Kita juga tidak lagi mencari loss function, melainkan ekspektasi dari loss function. Disamping

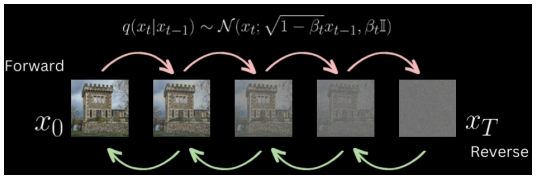

Figure 2: reverse process

itu, kita tidak memaksimalkan log-likelihood, 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.

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(x_0)] = \mathbb{E}_{q(\mathbf{x}_0)}\left[-\log \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]\right] \tag{41}$$

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(x_0)] \leq \mathbb{E}_{q(\mathbf{x}_0)}\left[\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]\right] \tag{42}$$

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(x_0)] \leq \mathbb{E}_{q(\mathbf{x}_0)}\left[\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right]\right] \tag{43}$$

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(x_0)] \leq \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \tag{44}$$

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

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{45}$$

$q(\mathbf{x}_t|\mathbf{x}_{t-1})$ 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.

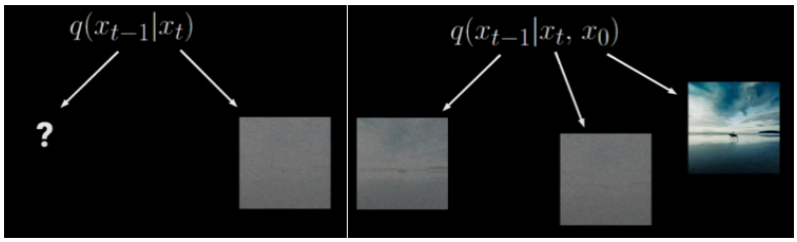

Figure 3: high variance vs. low variance

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

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}{p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{46}$$

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.

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}{p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{47}$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}{p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{48}$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\ q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)\ p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{49}$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} =$$

$$\log \frac{\cancel{q(\mathbf{x}_1|\mathbf{x}_0)}\, q(\mathbf{x}_{T-1}|\mathbf{x}_T, \mathbf{x}_0)\, q(\mathbf{x}_T|\mathbf{x}_0)\, q(\mathbf{x}_{T-2}|\mathbf{x}_{T-1}, \mathbf{x}_0)\, \cancel{q(\mathbf{x}_{T-1}|\mathbf{x}_0)} \ldots q(\mathbf{x}_1|\mathbf{x}_2, \mathbf{x}_0)\, \cancel{q(\mathbf{x}_2|\mathbf{x}_0)}}{\cancel{q(\mathbf{x}_{T-1}|\mathbf{x}_0)}\, q(\mathbf{x}_{T-2}|\mathbf{x}_0) \ldots \cancel{q(\mathbf{x}_2|\mathbf{x}_0)}\, \cancel{q(\mathbf{x}_1|\mathbf{x}_0)}\, p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \quad (50)$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \quad (51)$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p(\mathbf{x}_T)\, p_\theta(\mathbf{x}_0|\mathbf{x}_1) \prod_{t=2}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \quad (52)$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \quad (53)$$

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

$$-\log p_\theta(\mathbf{x}_0) \leq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right]. \quad (54)$$

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

$$\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \quad (55)$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \Bigg[D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)) + \sum_{t=2}^{T} D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\tag{56}$$

Didapatkan:

$$L = \mathbb{E}_q \Bigg[D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)) + \sum_{t=2}^{T} D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1 \tag{57}$$

dengan

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}), \tag{58}$$

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) \coloneqq \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t, \tag{59}$$

$$\tilde{\beta}_t \coloneqq \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}}\beta_t. \tag{60}$$

Kita dapat menguraikan L menjadi:

$$L_T = D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)). \tag{61}$$

$$L_{t-1} = D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)). \tag{62}$$

$$L_0 = \log p_\theta(\mathbf{x}_0|\mathbf{x}_1). \tag{63}$$

3.6 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.

3.7 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:

$$\Sigma_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I} \tag{64}$$

sehingga

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I}) \tag{65}$$

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:

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \| \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t) \|^2 \right] + C, \tag{66}$$

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

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \| \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{x}_t - \mu_\theta(\mathbf{x}_t, t) \|^2 \right] + C, \tag{67}$$

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

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{68}$$

$$\iff \mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon) \tag{69}$$

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

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \| \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{x}_t - \mu_\theta(\mathbf{x}_t, t) \|^2 \right] + C \tag{70}$$

$$= \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \| \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon) - \mu_\theta(\mathbf{x}_t, t) \|^2 \right] + C \tag{71}$$

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

$$= \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \| \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t(\mathbf{x}_0, \epsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon) - \mu_\theta(\mathbf{x}_t(\mathbf{x}_0, \epsilon), t) \|^2 \right] + C, \tag{72}$$

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:

$$\mu_\theta(\mathbf{x}_t, t) = \bar{\mu}_t(\mathbf{x}_t, \mathbf{x}_0((x)_t, \epsilon_\theta)) \tag{73}$$

$$= \bar{\mu}_t(\mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t))) \tag{74}$$

$$= \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)) \tag{75}$$

Subtitusi persamaan (16) pada persamaan (15):

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \| \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon) - \mu_\theta(\mathbf{x}_t, t) \|^2 \right] + C \tag{76}$$

$$= \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \| \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon) - \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t)) \|^2 \right] + C \tag{77}$$

$$= \mathbb{E}_q \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)} \| \epsilon - \epsilon_\theta(\mathbf{x}_t, t) \|^2 \right] + C \tag{78}$$

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:

$$L_{simple}(\theta) \coloneqq \mathbb{E}_q \left[\| \epsilon - \epsilon_\theta(\mathbf{x}_t, t) \|^2 \right] \tag{79}$$

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 $\mathcal{N}(\mu, \sigma^2) = \mu + \sigma\epsilon$:

$$\mathbf{x}_{t-1} = \mu_\theta(\mathbf{x}_t, t) + \Sigma_\theta(\mathbf{x}_t, t) + \epsilon \tag{80}$$

$$= \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)) + \sqrt{\beta_t}\epsilon \tag{81}$$

4 Data Scaling