



# Denoising Diffusion Probabilistic Models

Giovani Tavares de Andrade

Instituto de Matemática e Estatística  
(IME-USP)

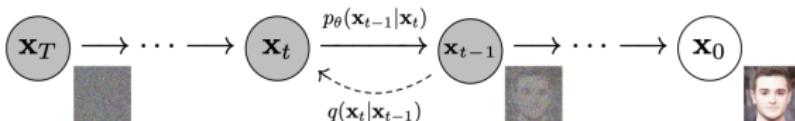
11 / 2025

# Outline

- ① What are DDPMs
- ② Forward Process
- ③ Reverse Process
- ④ Training Algorithm
- ⑤ Sampling Algorithm
- ⑥ Bibliography

# Overview

- In simple terms, Denoising Diffusion Probabilistic Models (DDPMs) learn what part of a signal is noise — image generation is just what happens when you keep removing it.
- By using DDPM's output, a sampling algorithm can be used to remove the noise from a noisy input which results in a denoised output.



# DDPMs Building Blocks

DDPMs are made of two processes: a **forward** and a **reversion** process.

- ① **Forward Process:** gradually adds noise to a image by sampling from a normal distribution according to a Markov Chain
- ② **Reverse Process:** removes added noise by sampling from another normal distribution according to another Markov Chain

The core idea of the training of DDPMs involve learning the reversion process' Gaussian Noises to be removed. The transitions distributions from the reversion process chain have parameters are functions of the noise predicted by the DDPM. Hence, **DDPMs make it possible to sample from such distributions.**

# DDPMs Building Blocks

## ① Forward Process' Distribution

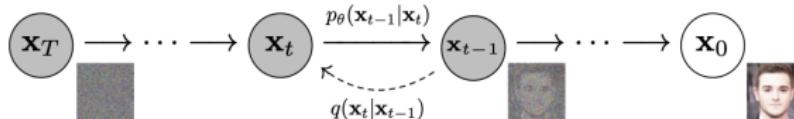
$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1 - \beta_t} \times x_{t-1}; \beta_t I) \quad (1)$$

## ② Reverse Process' Distribution

$$p(x_T) = \mathcal{N}(x_T; 0; 1) \quad (2)$$

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t) \quad (3)$$

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t); \Sigma_\theta(x_t, t)) \quad (4)$$



# Forward Process' Transitions Distribution

- 

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1 - \beta_t} \times x_{t-1}; \beta_t I) \quad (5)$$

- According to the original DDPM paper, the **Variance Schedule**  $\beta_1, \dots, \beta_T$  sequence that defines the noisy images distribution are held constant
- Ideally, one would need a single transition from  $x_0$  to get to  $x_t$ , with  $t > 0$ .
- **The authors of the paper achieve such ideal scenario by defining a cumulative noise  $\alpha_t$  presented in the following slide.**

# Cumulative Noise

The cumulative noise parameter ( $\bar{\alpha}_t$ ) up to the  $t$ -th step is defined as:

$$\alpha_t := 1 - \beta_t \quad (6)$$

$$\bar{\alpha}_t := \prod_{i=1}^t \alpha_i \quad (7)$$

which leads to: (8)

$$q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I) \quad (9)$$

This follows from the Markov property of the forward diffusion process.

# Reverse Process' Transitions Distribution

- 

$$p(x_T) = \mathcal{N}(x_T; 0; 1) \quad (10)$$

$$p_{\theta}(x_{0:T}) := p(x_T) \prod_{t=1}^T p_{\theta}(x_{t-1}|x_t) \quad (11)$$

$$p_{\theta}(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_{\theta}(x_t, t); \Sigma_{\theta}(x_t, t)) \quad (12)$$

- What we ultimately want is to have a  $\mathbf{x}_0$  that is as likely as possible. We can marginalize  $\mathbf{x}_0$  using the latent variables  $\mathbf{x}_{1:T}$ , i.e., by using the noisy images.

## Reverse Process' Prior

- Ideally, the Reverse Process would enable us to sample from:

$$p(\mathbf{x}_0) = \int p(\mathbf{x}_0, \mathbf{x}_{1:T}) d\mathbf{x}_{1:T} \quad (13)$$

(14)

- **From the definition above, we see that  $p(\mathbf{x}_0)$  is very complex due to its multidimensionality, which makes it intractable. That is why in DDPMs,  $p(\mathbf{x}_0)$  is never computed directly, but instead its lower bound.**

# Evidence Lower Bound (ELBO)

- DDPMs are not trained to sample from  $p(\mathbf{x}_0)$ , but instead to maximize its lower bound ELBO ( $L$ )
- 

$$\log[p_\theta(\mathbf{x}_0)] = \log \int_{\mathbf{x}_{1:T}} p(\mathbf{x}_0, \mathbf{x}_{1:T}) d\mathbf{x}_{1:T} \quad (15)$$

$$\log[p_\theta(\mathbf{x}_0)] = \log \int_{\mathbf{x}_{1:T}} p(\mathbf{x}_0, \mathbf{x}_{1:T}) \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} d\mathbf{x}_{1:T} \quad (16)$$

by definition, (17)

$$\mathbb{E}_q[f(\mathbf{x}_{1:T})] = \int q(\mathbf{x}_{1:T}|\mathbf{x}_0) f(\mathbf{x}_{1:T}) d\mathbf{x}_{1:T} \quad (18)$$

$$\implies \log[p_\theta(\mathbf{x}_0)] = \log(\mathbb{E}_q \left[ \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right]) \quad (19)$$

the Jensen's inequality tells us (20)

$$f(\mathbb{E}(\mathbf{X})) \geq \mathbb{E}(f(\mathbf{X})) \quad (21)$$

for any concave function  $f$ .  $\log$  is concave, hence: (22)

$$\log[p_\theta(\mathbf{x}_0)] \geq \mathbb{E}_q \left[ \log \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \quad (23)$$

We define the Evidence Lower Bound  $L$  as: (24)

$$L := \mathbb{E}_q \left[ -\log \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \quad (25)$$

## Evidence Lower Bound (ELBO)

- DDPMs are not trained to sample from  $p(\mathbf{x}_0)$ , but instead to maximize its lower bound ELBO ( $L$ )
- With more algebraic manipulation and using the fact that both the forward and reverse processes are Markov Chains, one can derive the following equation:

$$L := \mathbb{E}_q \left[ -\log \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \quad (27)$$

$$L = \mathbb{E}_q \left[ -\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} \right] - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) - \mathbb{E}_q \left[ \sum_{t=2}^T D_{KL} [q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) || p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)] \right] \quad (28)$$

- **We can conclude that maximizing ELBO ( $L$ ) is equivalent to minimizing the KL-Divergence between  $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$  and  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ , i.e., minimizing the divergence between the forward and reverse processes' distributions.**

# Defining $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0) q(\mathbf{x}_{t-1} | \mathbf{x}_0)}{q(\mathbf{x}_t | \mathbf{x}_0)} \quad (29)$$

we know the q distribution from Definition, hence (30)

$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$  is a product of known Gaussians over another known Gaussian (31)

$$\mu_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{(1 - \bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x}_t + (1 - \alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0}{(1 - \bar{\alpha}_t)} \quad (32)$$

$$\Sigma_q(t) = \frac{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{I} \quad (33)$$

$$\implies q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_q(\mathbf{x}_t, \mathbf{x}_0); \Sigma_q(t)) \quad (34)$$

# Defining $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t); \Sigma_\theta(t)) \quad (35)$$

from DDPM's paper: (36)

$$\Sigma_\theta(t) = \Sigma_q(t) \quad (37)$$

we are only left with the distribution's mean  $\mu_\theta$  (38)

$$\implies p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t); \Sigma_q(t)) \quad (39)$$

## Computing the KL Divergence Between $q$ and $p_\theta$

- Now we know we are trying to compute the KL Divergence between two Gaussians with the exact same variance.
- For that, there is the following result that arrives from the definition of such divergence

$$d_1(x) = \mathcal{N}(\mu_1, \sigma^2) \quad (40)$$

$$d_2(x) = \mathcal{N}(\mu_2, \sigma^2) \quad (41)$$

The KL divergence  $D_{KL}(d_1 \| d_2)$  is given by: (42)

$$D_{KL}(d_1 \| d_2) = \frac{(\mu_1 - \mu_2)^2}{2\sigma^2} \quad (43)$$

Hence, (44)

$$\mathbf{D}_{KL}[q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) || p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)] = \mathbf{D}_{KL}(\mathcal{N}(\mathbf{x}_{t-1}; \mu_q(\mathbf{x}_t, \mathbf{x}_0); \Sigma_q(t)), \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t); \Sigma_q(t))) \quad (45)$$

$$= \frac{1 - \bar{\alpha}_t}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \left\| (\mu_q - \mu_\theta)_2^2 \right\| \quad (46)$$

- We just need to minimize the difference between the means of the reverse and forward processes' distributions, i.e., minimize  $\left\| (\mu_q - \mu_\theta)_2^2 \right\|$ .

# Defining The Model's Prediction

- We can use the prediction of our model as the forward process' mean
- 

$$\mu_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{(1 - \bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x}_t + (1 - \alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0}{(1 - \bar{\alpha}_t)} \quad (47)$$

we can define the prediction (48)

$$\mu_\theta(\mathbf{x}_t) := \hat{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) \quad (49)$$

$$= \frac{(1 - \bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x}_t + (1 - \alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_\theta}{(1 - \bar{\alpha}_t)} \quad (50)$$

$$\implies \mathbf{D}_{\mathbf{KL}}(\mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t); \Sigma_q(t)), \mathcal{N}(\mathbf{x}_{t-1}; \mu_q(\mathbf{x}_t, \mathbf{x}_0); \Sigma_q(t))) \quad (51)$$

$$= \frac{(1 - \bar{\alpha}_t)(\bar{\alpha}_{t-1})}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \|(\mathbf{x}_\theta - \mathbf{x}_0)_2^2\| \quad (52)$$

- **Theoretically, this equation could be used as the loss function directly, but we can rewrite the images  $\mathbf{x}_\theta$  and  $\mathbf{x}_0$  in function of the Gaussian noises.**

# Defining The Model's Prediction

- We can rewrite  $\mathbf{x}_t$  and  $\mathbf{x}_0$  as functions of the added gaussian noises
- 

$$q(\mathbf{x}_t | \mathbf{x}_0) := \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} \times x_0; (1 - \bar{\alpha}_t)\mathbb{I}) \quad (53)$$

which let's us write

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon \quad (55)$$

$$\Rightarrow \mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon}{\sqrt{\bar{\alpha}_t}} \quad (56)$$

for a Standard Gaussian Noise  $\epsilon$ . (57)

We can now define our prediction  $\hat{\epsilon} = \epsilon_\theta$  (58)

$$\mathbf{x}_\theta = \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta}{\sqrt{\bar{\alpha}_t}} \quad (59)$$

$$\Rightarrow \frac{(1 - \bar{\alpha}_t)(\bar{\alpha}_{t-1})}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \|(\mathbf{x}_\theta - \mathbf{x}_0)_2^2\| = \frac{(1 - \bar{\alpha}_t)(\bar{\alpha}_{t-1})}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \frac{(1 - \alpha_t)^2}{(1 - \bar{\alpha}_t)\alpha_t} \|(\epsilon_\theta - \epsilon)_2^2\| \quad (60)$$

- **The DDPM paper authors mention that optimizing  $\|(\epsilon_\theta - \epsilon)_2^2\|$  without the scaling factor with the cumulative noise  $\alpha_t$  is enough.**

# Noise Predictor Training

- 1: **repeat**
- 2:  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$  ▷ Sample image from training set
- 3:  $t \sim \text{Uniform}(\{1, \dots, T\})$  ▷ Sample the step of the Forward Process Markov Chain
- 4:  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  ▷ Sample standard gaussian noise to be added to the input
- 5:  $\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \epsilon$  ▷ Forward Process/ Generating Noisy Image
- 6: Take Gradient Descent Step on  $\nabla_{\theta}(\|\epsilon - \epsilon_{\theta}(\mathbf{x}_t, t)\|)$
- 7: **until** converged

# Sampling Algorithm Derivation

- 

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t); \Sigma_\theta(t)) \quad (61)$$

$$\mu_\theta(\mathbf{x}_t) = \frac{(1 - \bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x}_t + (1 - \alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_\theta}{(1 - \bar{\alpha}_t)} \quad (62)$$

$$\mathbf{x}_\theta = \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta}{\sqrt{\bar{\alpha}_t}} \quad (63)$$

$$\implies \mu_\theta(\mathbf{x}_t) = \frac{\mathbf{x}_t}{\sqrt{\alpha_t}} - \frac{(1 - \alpha_t)(\sqrt{1 - \bar{\alpha}_t})}{(1 - \bar{\alpha}_t)(\sqrt{\alpha_t})}\epsilon_\theta = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}}\epsilon_\theta \right) \quad (64)$$

$$\Sigma_\theta(t) = \Sigma_q(t) = \frac{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{I} \quad (65)$$

We now have defined  $\mathbf{x}_{t-1}$ 's mean and variance given  $\mathbf{x}_t$  which let's us write it as

(66)

$$\mathbf{x}_{t-1} = \mu_\theta(\mathbf{x}_t) + \sqrt{\Sigma_\theta(t)}\mathbf{z} \quad (67)$$

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (68)$$

# Sampling Algorithm

```
1: repeat
2:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$                                 ▷ Sample random noisy image
3:  $T \sim \text{Uniform}(\{1, \dots, 1000\})$                 ▷ Sample random length of the Denoising Chain
4: for  $t = T, \dots, 1$  do
5:    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$  else  $\mathbf{z} = \mathbf{0}$ 
6:    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sqrt{\Sigma_\theta(t)} \mathbf{z}$     ▷ Sampling  $\mathbf{x}_{t-1}$  from  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$ 
7: end for
8: return  $\mathbf{x}_0$ 
```

# References I

- [1] Ho, J., Jain, A., and Abbeel, P. (2020). *Denoising Diffusion Probabilistic Models*. <https://arxiv.org/pdf/2006.11239>
- [2] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. <https://arxiv.org/abs/1503.03585>
- [3] 3Blue1Brown. (n.d.). *Neural Networks [7.8]: Deep Learning - Variational Bound (YouTube)*.  
<https://www.youtube.com/watch?v=pStDscJh2Wo>
- [4] Jake Tae. (2021). *A Step Up with Variational Autoencoders*.  
<https://jaketae.github.io/study/vae/>
- [5] Jake Tae. (2021). *From ELBO to DDPM*.  
<https://jaketae.github.io/study/elbo/>

## References II

- [6] Yang, X. (2017). *Understanding the Variational Lower Bound*. <https://xyang35.github.io/2017/04/14/variational-lower-bound/>
- [7] AI Coffee Break with Letitia. (n.d.). *Denoising Diffusion Probabilistic Models — DDPM Explained (YouTube)*.  
<https://www.youtube.com/watch?v=H45lF4sUgiE&t=880s>

# Hora da Implementação :)