Diffusion Models Unpacked

1 Denoising Diffusion Probabilistic Models

This section recaps the DDPM [1, 3].

1.1 Forward Diffusion

For each time step, define a weighting hyper-parameter β_t (from 10^{-4} to 0.02 in the paper), $\alpha_t = 1 - \beta_t$. According to Markov Chain,

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1 - \alpha_t} \epsilon_{t-1} \tag{1}$$

where $\epsilon_t \sim \mathcal{N}(0, I)$. While β increases in the process, more and more noises will be added.

x_t can be computed directly from x_0 with Reparameterization Sampling as

$$x_t = \sqrt{\overline{\alpha}_t} x_0 + \sqrt{1 - \overline{\alpha}_t} \epsilon \tag{2}$$

where $\overline{\alpha}_t = \alpha_t \alpha_{t-1} \dots \alpha_1$ and ϵ_t is noise following Gaussian Distribution.

Proof. Since $x_{t-1} = \sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{1 - \alpha_{t-1}} \epsilon_{t-2}$, by inserting it to Equation 1, we obtain

$$x_t = \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{\alpha_t \left(1 - \alpha_{t-1}\right)} \epsilon_{t-2} + \sqrt{1 - \alpha_t} \epsilon_{t-1}$$

Since both ϵ_{t-2} and ϵ_{t-1} follow $\mathcal{N}(0, I)$ and are independent [1] [2],
the equation above can be re-written as

$$x_t = \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}} \overline{\epsilon}_{t-2}$$

where $\overline{\epsilon}_{t-2} \sim \mathcal{N}(0, I)$. By continuing this deduction, we will get Equation 2.

1.2 Reverse Diffusion

In the reverse diffusion during training, q instead of p is used to denote the probability. The goal of reverse diffusion is to compute $q(x_0|x_T)$. According to Bayes Theorem,

$$q\left(x_{t-1}|x_t\right) = \frac{q\left(x_t|x_{t-1}\right) q\left(x_{t-1}\right)}{q\left(x_t\right)} \tag{3}$$

$q(x_{t-1}|x_t)$ is posterior, $q(x_t|x_{t-1})$ is likelihood, $q(x_{t-1})$ is prior and $q(x_t)$ is evidence.
Assuming we know x_0, according to the forward diffusion Equation 1 and 2,

[1] The sum of two independent random variables with Gaussian distributions also follows a Gaussian distribution according to the convolution theorem.

[2] $Var(cX) = c^2 Var(X)$ where c is a constant and X is a random variable. For example, $Var(\sqrt{1 - \alpha_t}\epsilon_{t-1}) = (1 - \alpha_t) Var(\epsilon_{t-1})$. $Var(X + Y) = Var(X) + Var(Y)$, if random variables X and Y are independent.

$$\begin{aligned} q\left(x_{t-1}|x_0\right) &\sim \mathcal{N}\left(\sqrt{\overline{\alpha}_{t-1}}x_0, 1-\overline{\alpha}_{t-1}\right) \\ q\left(x_t|x_{t-1}, x_0\right) &\sim \mathcal{N}\left(\sqrt{\alpha_t}x_{t-1}, 1-\alpha_t\right) \\ q\left(x_t|x_0\right) &\sim \mathcal{N}\left(\sqrt{\overline{\alpha}_t}x_0, 1-\overline{\alpha}_t\right) \end{aligned} \tag{4}$$

It is noted that x_t and x_0 are known while x_{t-1} is unknown. Since $q(x_{t-1}|x_t) \sim \mathcal{N}(0, I)$, we only need to compute its expectation and variance to obtain its PDF [3]. For the moment, we only consider the exponential part[4]. The right side of the Bayes equation becomes

$$\exp\left\{-\frac{1}{2}\left[\frac{\left(x_t-\sqrt{\alpha_t}x_{t-1}\right)^2}{\beta_t}+\frac{\left(x_{t-1}-\sqrt{\overline{\alpha}_{t-1}}x_0\right)^2}{1-\overline{\alpha}_{t-1}}-\frac{\left(x_t-\sqrt{\overline{\alpha}_t}x_0\right)^2}{1-\overline{\alpha}_t}\right]\right\}$$

or

$$\exp\left\{-\frac{1}{2}\left[\left(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\overline{\alpha}_{t-1}}\right)x_{t-1}^2-\left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t+\frac{2\sqrt{\overline{\alpha}_{t-1}}}{1-\overline{\alpha}_{t-1}}x_0\right)x_{t-1}+C\left(x_t, x_0\right)\right]\right\}$$

where C is a constant. The exponential part of a standard Gaussian is:

$$e^{-\frac{(x-\mu)^2}{2\sigma^2}} = e^{-\frac{1}{2}\left(\frac{1}{\sigma^2}x^2-\frac{2\mu}{\sigma^2}x+\frac{\mu^2}{\sigma^2}\right)}$$

By comparing the coefficients for x_{t-1}^2 and for x_{t-1} in Equation 1.2 and 1.2, respectively, we can compute the expectation and variance.

$$\widetilde{\sigma}_t^2 = \frac{\beta_t\left(1-\overline{\alpha}_{t-1}\right)}{1-\overline{\alpha}_t} \tag{5}$$

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

x_0 can be computed from x_t based on Equation 1, so

$$\widetilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon\right) \tag{6}$$

In Equation 5 and 6, the only unknown variable, the noise ϵ, will be found in the network.

1.3 Training

It can be proved that in order to optimize the likelihood of the value represented by Equation 3, the predicted noise generated by the network should be as close as possible to the noise ϵ as defined in Equation 6. Hence, a network is required to train the value of ϵ. The training algorithm is shown in Figure 1.

x_t can be computed from x_0, $\overline{\alpha}_t$ and a Gaussian distribution ϵ (see Figure 1b). Using the time embedding and x_t as input, a network ϵ_θ (typically a UNet variant) will predict ϵ. The algorithm optimizes on a MSE loss (Step 5 in Algorithm 1). The model parameters are shared across all time steps. Given the infinitesimal nature of each time step, utilizing the same prediction model for ϵ at every step is reasonable.

[3] According to Gaussian Bayesian, if the likelihood and prior are Gaussian, the posterior is also Gaussian.

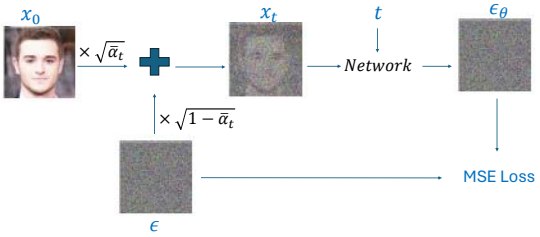
[4] The Probability Density Function for Gaussian distribution is

$$f\left(x|\mu, \sigma\right) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

(a) DDPM training algorithm [1].



(b) Illustration of DDPM training.

Fig. 1: DDPM training.

1.4 Sampling

During sampling (for inference), x_t is restored to x_{t-1}, then to x_{t-2}, and so on, until x_0 is restored (see Figure 2.

2 Image Generation

Here, we illustrate the process of generating images from text using DALL·E 2 (see Fig.3) [2], focusing on the diffusion model.

The training process involves two key models: the prior and the decoder. The steps are:

1. A pre-trained CLIP model generates embeddings for input image.
2. The same CLIP model generates embeddings for input text.
3. The text embedding passes through a prior ("autoregressive or diffusion prior" in the original paper), resulting in an image embedding. The image embedding generated in Step 1 serves as the ground truth.
4. The decoder employs diffusion models to convert the image embedding from Step 3 into the final image output.

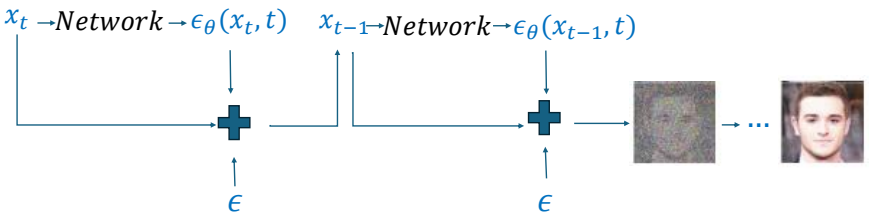
References

1. Ho, J.: Denoising diffusion probabilistic models. NeurIPS (2020)
2. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents (2022), https://arxiv.org/pdf/2204.06125
3. Weng, L.: What are diffusion models? (2021), https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

(a) DDPM training algorithm [1].



(b) Illustration of DDPM training.

Fig. 2: DDPM training.

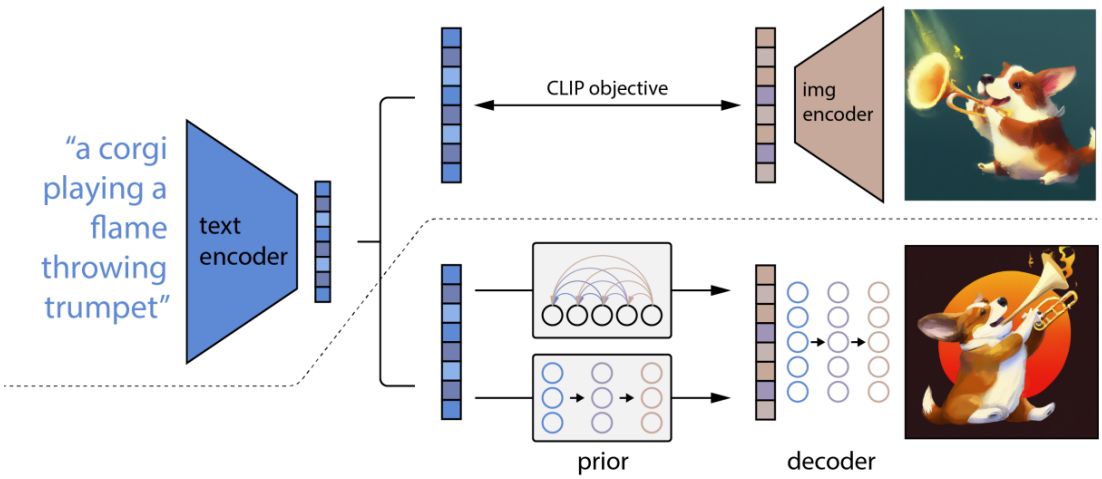


Fig. 3: High-level overview of $DALL \cdot E$ 2 [2].