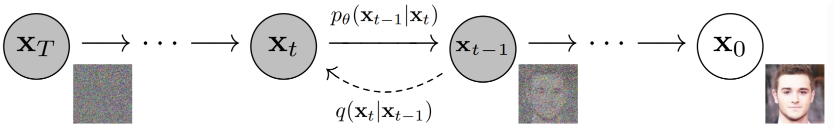
Denoising Diffusion Probabilistic Models

Jonathan Ho and Ajay Jain and Pieter Abbeel
NeurIPS 2020

Department of Statistics, Seoul National University
Presented by Sangmoon Han

2025-09-04

Overview



- Let $T \in \mathbb{N}$.
- Denoising Diffusion Probabilistic Models(DDPM) are generative models having two steps.
  1. Define a process which converts image data x_0 to standard gaussian data x_T.
  2. Define a reverse process to learn a finite-time reversal of this process.

Forward Process

- Let x_0 denote a data distributed according to some unknown distribution $q\left(x_0\right)$
- For $x_{0:T}$, DDPM considers the joint distribution $q(x_{0:T})$ by fixed $\beta_1, \ldots, \beta_T$ as

$$q(x_{0:T}) = q(x_0) \prod_{t=1}^{T} q(x_t | x_{t-1}), \quad q(x_t | x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t \mathbf{I})$$

- That means $q(x_{0:T})$ is considered as a Markov chain that gradually adds Gaussian noise.

$$x_t | x_{t-1} \overset{d}{=} \sqrt{1-\beta_t} x_{t-1} + \sqrt{\beta_t} \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

Forward Process

- Properties of the forward process of DDPM :
  1. Let $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, then

$$x_t|x_0 \sim q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$
$$x_t|x_0 \stackrel{d}{=} \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

  2. If $0 < \beta_t < 1$, for all $t = 1, \cdots, T$, then $\bar{\alpha}_t \to 0$ as $T \to \infty$ and

$$q(x_T|x_0) \approx \mathcal{N}(x_T; 0, \mathbf{I}) \quad \text{as} \quad T \to \infty$$

- First property means if one wishes to sample x_t, it can be drawn directly from $q(x_t|x_0)$ rather than using the full forward process $q(x_{1:t}|x_0)$.
- Second property means the sample x_T generated by $q(x_{0:T})$ can be viewed as the sample generated from $\mathcal{N}(0, \mathbf{I})$ for sufficient large T.

Diffusion models

- Although the forward process $q(x_t|x_{t-1})$ is Markov chain, the time-reversed sequence $y_t := x_{T-t}$ is not guaranteed to be Markov under q, hence directly sampling backward from x_T to x_0 is generally intractable.
- From the perspective of variational inference, Diffusion models consider $p_\theta(x_{0:T})$ called the reverse process as the approximate distribution of $q(x_{0:T})$.
- DDPM considers $p_\theta(x_{0:T})$ is defined as a Markov chain with Gaussian transitions starting at $p(x_T) = \mathcal{N}(x_T; \mathbf{0}, \mathbf{I})$:

$$p_\theta(x_{0:T}) := p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \quad p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \sigma_t^2\mathbf{I})$$

, where σ_t^2 are hyperparameters, θ are learnable parameters.

Objective function

- Training is performed by minimizing the KL divergence between these two:[1]

$$
\begin{aligned}
D_{\mathrm{KL}}\left(q(x_{0:T}) \| p_\theta(x_{0:T})\right) &= -\mathbb{E}_{q(x_{0:T})}\left[\log p_\theta\left(x_{0:T}\right)\right] + C_1 \\
&= \underbrace{\mathbb{E}_{q(x_0, x_1, \cdots, x_T)}\left[-\log p\left(x_T\right) - \sum_{t=1}^{T} \log \frac{p_\theta\left(x_{t-1}|x_t\right)}{q\left(x_t|x_{t-1}\right)}\right]}_{:=L} + C_2 \\
&\geq \mathbb{E}\left[-\log p_\theta\left(x_0\right)\right] + C_3, \text{ where } p_\theta(x_0) := \int p_\theta(x_{0:T}) dx_{1:T}.
\end{aligned}
$$

- Furthermore, L can be rewritten as :

$$
L = \mathbb{E}_q\left[\underbrace{D_{\mathrm{KL}}\left(q\left(x_T|x_0\right) \| p\left(x_T\right)\right)}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}\left(q\left(x_{t-1}|x_t, x_0\right) \| p_\theta\left(x_{t-1}|x_t\right)\right)}_{L_{t-1}} \underbrace{- \log p_\theta\left(x_0|x_1\right)}_{L_0}\right]
$$

- L_T does not include parameters and is therefore treated as a constant.
- Since x_0 is a discrete random vector taking values in $\{0, \ldots, 255\}$, $L_0 = -\log p_\theta\left(x_0|x_1\right)$ is approximated by a discrete probability density, but details are skiped here.

[1] In DDPM, "Training is performed by optimizing the usual variational bound on negative log likelihood"

Focusing on second term of L

- If $q(x) = \mathcal{N}\left(x|\mu_1, \sigma_1^2 I\right),\ p(x) = \mathcal{N}\left(x|\mu_2, \sigma_2^2 I\right)$,

$$D_{KL}(q\|p) \propto \frac{d}{2}\log\left(\frac{\sigma_2^2}{\sigma_1^2}\right) + \frac{d\left(\sigma_1^2 - \sigma_2^2\right) + \|\mu_1 - \mu_2\|^2}{2\sigma_2^2}$$

- $q\left(x_{t-1}|x_t, x_0\right)$ can be computed as

$$q\left(x_{t-1}|x_t, x_0\right) = \mathcal{N}\left(x_{t-1}; \tilde{\boldsymbol{\mu}}_t(x_t, x_0), \tilde{\beta}_t \mathbf{I}\right)$$

, where $\tilde{\boldsymbol{\mu}}_t\left(x_t, x_0\right) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}x_t$ and $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$

- $p_\theta\left(x_{t-1}|x_t\right) = \mathcal{N}\left(x_{t-1}; \boldsymbol{\mu}_\theta\left(x_t, t\right), \sigma_t^2 \mathbf{I}\right)$.
- For $t \in \{2, \cdots, T\}$, L_{t-1} can be rewritten as :

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\boldsymbol{\mu}}_t\left(x_t, x_0\right) - \boldsymbol{\mu}_\theta\left(x_t, t\right)\right\|^2\right] + C$$

Focusing on second term of L

- Reparameterize x_t, x_0 from $q(x_T|x_0) = \mathcal{N}(x_T; \sqrt{\bar{\alpha}_T}x_0, (1-\bar{\alpha}_T)\mathbf{I})$.
  1. $x_t \to x_t(x_0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  2. $x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t(x_0, \epsilon) - \sqrt{1-\bar{\alpha}_t}\epsilon\right)$
- Since $\tilde{\boldsymbol{\mu}}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t$,

$$\tilde{\boldsymbol{\mu}}_t(x_t, \boldsymbol{\epsilon}) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)$$

- $\|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2$ can be rewritten by reparameterization

$$L_{t-1} - C = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\boldsymbol{\mu}}_t(x_t, x_0) - \boldsymbol{\mu}_\theta(x_t, t)\|^2\right]$$
$$= \mathbb{E}_{x_0, \boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right) - \boldsymbol{\mu}_\theta(x_t(x_0, \boldsymbol{\epsilon}), t)\right\|^2\right]$$

Reparametrization of $\boldsymbol{\mu}_\theta(x_t, t)$

- Recall, $\tilde{\boldsymbol{\mu}}_t(x_t, \boldsymbol{\epsilon}) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)$.
- Since x_t is available as input to the model, we may choose the parametrization :

$$\boldsymbol{\mu}_\theta(x_t, t) = \tilde{\boldsymbol{\mu}}_t\left(x_t, \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(x_t)\right)\right) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(x_t, t)\right)$$

- L_{t-1} can be expressed as

$$\begin{aligned} L_{t-1} &= \mathbb{E}_{x_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(x_t, t)\right\|^2\right] \leftarrow x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \\ &= \mathbb{E}_{x_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right] \end{aligned}$$

- Finally, L can be expressed as

$$L = \mathbb{E}_q\left[L_T + \sum_{t>1}\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2 + L_0\right]$$

Algorithm

Algorithm 1 Training

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

Algorithm 2 Sampling

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

- Authors say we found it beneficial to sample quality (and simpler to implement) to train on the following variant of the variational bound.

  1. $L_{t-1} = \mathbb{E}_{x_0, \boldsymbol{\epsilon}}\left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right],$
  2. $x_{t-1} \overset{\mathrm{d}}{=} \boldsymbol{\mu}_\theta(x_t, t) + \sigma_t z, \quad z \sim \mathcal{N}(0, \mathbf{I}),$
  3. $\boldsymbol{\mu}_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(x_t, t)\right)$

Figure 1: Unconditional CIFAR10 progressive generation.

Experiments

Table 1: CIFAR10 results. NLL measured in bits/dim.

Model	IS	FID	NLL Test (Train)
Conditional			
EBM [11]	8.30	37.9	
JEM [17]	8.76	38.4	
BigGAN [3]	9.22	14.73	
StyleGAN2 + ADA (v1) [29]	**10.06**	**2.67**	
Unconditional			
Diffusion (original) [53]			≤ 5.40
Gated PixelCNN [59]	4.60	65.93	3.03 (2.90)
Sparse Transformer [7]			**2.80**
PixelIQN [43]	5.29	49.46	
EBM [11]	6.78	38.2	
NCSNv2 [56]		31.75	
NCSN [55]	8.87±0.12	25.32	
SNGAN [39]	8.22±0.05	21.7	
SNGAN-DDLS [4]	9.09±0.10	15.42	
StyleGAN2 + ADA (v1) [29]	**9.74** ± 0.05	3.26	
Ours (L, fixed isotropic $\mathbf{\Sigma}$)	7.67±0.13	13.51	≤ 3.70 (3.69)
Ours (L_{simple})	9.46±0.11	**3.17**	≤ 3.75 (3.72)

Table 2: Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

Objective	IS	FID
$\tilde{\mu}$ prediction (baseline)		
L, learned diagonal $\mathbf{\Sigma}$	7.28±0.10	23.69
L, fixed isotropic $\mathbf{\Sigma}$	8.06±0.09	13.22
$\lVert\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{\mu}}_\theta\rVert^2$	–	–
ϵ prediction (ours)		
L, learned diagonal $\mathbf{\Sigma}$	–	–
L, fixed isotropic $\mathbf{\Sigma}$	7.67±0.13	13.51
$\lVert\tilde{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}_\theta\rVert^2$ (L_{simple})	**9.46±0.11**	**3.17**

Figure 2: This table shows Inception scores(IS), FID scores, and negative log likelihoods(NLL) (lossless codelengths) on CIFAR10.

- Experiment setting
  1. Set $T = 1000$.
  2. Set the forward process variances to constants increasing linearly from $\beta_1 = 10^{-1}$ to $\beta_T = 0.02$.
  3. To represent the reverse process, DDPM use a [1, U-Net] backbone similar to an unmasked [2, PixelCNN++].

References I

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV]. URL: https://arxiv.org/abs/1505.04597.

Tim Salimans et al. *PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications*. 2017. arXiv: 1701.05517 [cs.LG]. URL: https://arxiv.org/abs/1701.05517.