Denoising Diffusion Probabilistic Models

Jonathan Ho, Ajay Jain, and Pieter Abbeel
UC Berkeley
NeurIPS 2020
Presented by Minho Park
Contributions

• **Novel generative model** which obtain sample quality similar to ProgressiveGAN (SOTA).
  • A latent variable models training on a weighted variational bound.

• **Connection between diffusion probabilistic models and denoising score matching with Langevin dynamics.**
  • Score matching will be presented next week.

• Naturally admit a progressive lossy decompression scheme.
Generative Models

- A **generative model** is a statistical model of the joint distribution $P(X, Y)$ on given observable variable $X$ and target variable $Y$.

- A **discriminative model** is a model of the conditional probability $P(Y|X = x)$ of the target $Y$, given an observation $x$.
  - And classifiers computed without using a probability model are also referred to loosely as “discriminative”.

Discriminative and generative models of handwritten digits.
Generative Models in Computer Vision

- Generative models that work via the principle of maximum likelihood.

A taxonomy of deep generative models.

Maximum Likelihood Estimation

- MLE process: $\theta^*_{MLE} = \arg\max_\theta P(\mathcal{D}|\theta)$
  - Log likelihood: $\log P(\mathcal{D}|\theta) = \mathbb{E}_{x \sim p_{data}} \left[ \log p_{model}(x; \theta) \right]$

- KL divergence between data generating distribution and the model
  - $\theta^*_{MLE} = \arg\min_\theta D_{KL}(p_{data}(x) \parallel p_{model}(x; \theta))$
  - $= \arg\min_\theta \mathbb{E}_{x \sim p_{data}} \left[ \log p_{data}(x) - \log p_{model}(x; \theta) \right] = \arg\max_\theta \mathbb{E}_{x \sim p_{data}} \left[ \log p_{model}(x; \theta) \right]$

Optimization results using forward (a) / reverse (b), (c) KL divergence

Maximum likelihood estimation process.

Diffusion Models: Notation

- $p(x_{0:T})$: Joint distribution of $x_0, x_1, ..., x_T$
- $q$: Real distribution ($p_{data}$)
- $p_\theta$: Modelled distribution parameterized by $\theta$ ($p_{model}(\cdot; \theta)$)
- **Forward process (diffusion process)**: $q(x_t | x_{t-1}) \triangleq \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, 1 - \alpha_t I)$ Scheduled by $\alpha_t$ and $\beta_t = 1 - \alpha_t$
  - Then, $q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I)$ where $\bar{\alpha}_t \triangleq \prod_{s=1}^{t} \alpha_t$. Derived from Mathematical Induction and the definition of the Gaussian distribution.
- **Reverse process (denoising process)**: $p_\theta(x_{t-1} | x_t)$

Figure 2: The directed graphical model considered in this work.
Diffusion Models: Maximum Likelihood Estimation

- Let’s maximize the log-likelihood $\mathbb{E}_{x_0 \sim q} [\log p_\theta(x_0)]$.

$$p_\theta(x_0) = \int p_\theta(x_{0:T})dx_{1:T}$$

$$= \int p_\theta(x_{0:T}) \cdot \frac{q(x_{1:T}|x_0)}{q(x_{1:T}|x_0)} dx_{1:T}$$

$$= \int p_\theta(x_T) \cdot \frac{\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1})} \cdot q(x_{1:T}|x_0)dx_{1:T}$$

$$= \int p_\theta(x_T) \cdot q(x_{1:T}|x_0) \cdot \prod_{t=1}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} dx_{1:T}$$

$$= \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \left[ p_\theta(x_T) \right] \cdot \prod_{t=1}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}$$
Diffusion Models: Maximum Likelihood Estimation

- Let's maximize the log-likelihood $\mathbb{E}_{x_0 \sim q} \left[ \log p_{\theta}(x_0) \right]$.

  $$
  \mathbb{E}_{x_0 \sim q} \left[ \log p_{\theta}(x_0) \right] \\
  = \int \log p_{\theta}(x_0) \cdot q(x_0) dx_0
  $$

  $$
  = \int \log \left( \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \left[ p_{\theta}(x_T) \cdot \prod_{t=1}^{T} \frac{p_{\theta}(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right] \right) \cdot q(x_0) dx_0
  $$

  $$
  \geq \int \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \left[ \log \left( p_{\theta}(x_T) \cdot \prod_{t=1}^{T} \frac{p_{\theta}(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right) \right] \cdot q(x_0) dx_0
  $$

  Previous slide

Jensen’s inequality

Equality holds iff $p_{\theta}(x_T) \cdot \prod_{t=1}^{T} \frac{p_{\theta}(x_{t-1}|x_t)}{q(x_t|x_{t-1})}$ is constant for all $x_{1:T}$.

Evidence Lower BOund (ELBO) or Variational bound
Diffusion Models: Deriving the ELBO

• Understanding equality of the ELBO w.r.t. KL divergence.

\[
\log p_\theta(x_0) = \log \left( \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \left[ \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right] \right)
\geq \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \left[ \log \left( \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right) \right]
\]

• Then,

\[
\log p_\theta(x_0) - \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \left[ \log \left( \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right) \right] = \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \left[ \log p_\theta(x_0) - \log \left( \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right) \right]
\]

\[
= \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \left[ \log \left( \frac{q(x_{1:T}|x_0)}{p_\theta(x_{1:T}|x_0)} \right) \right]
\]

\[
= D_{KL} \left( q(x_{1:T}|x_0) \parallel p_\theta(x_{1:T}|x_0) \right) \geq 0
\]

Equality holds iff \( q(x_{1:T}|x_0) = p_\theta(x_{1:T}|x_0) \) is constant for all \( x_{1:T} \)
Diffusion Models: Training Objective

• Let’s minimize the negative log-likelihood $-\mathbb{E}_{x_0 \sim q}[\log p_\theta(x_0)]$.

$$-\mathbb{E}_{x_0 \sim q}[\log p_\theta(x_0)] = \int -\mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \left[ \log \left( p_\theta(x_T) \cdot \prod_{t=1}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right) \right] \cdot q(x_0) dx_0$$

$$= \mathbb{E}_{x_{0:T} \sim q} \left[ -\log p_\theta(x_T) - \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right] \triangleq L$$

Assumption of the Markov Chain

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

$$= \frac{1}{\sqrt{1 - \alpha_t}} \cdot \frac{\exp \left( \frac{||x_t - \sqrt{\alpha_t}x_{t-1}||^2}{2(1 - \alpha_t)} \right)}{\sqrt{1 - \alpha_{t-1}}} \cdot \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{1 - \alpha_{t-1}}} \cdot \frac{\exp \left( -\frac{||x_{t-1} - \sqrt{\alpha_{t-1}}x_0||^2}{2(1 - \alpha_{t-1})} \right)}{\sqrt{1 - \alpha_t}}$$

$$= \mathcal{N}(x_{t-1}; \frac{\sqrt{\alpha_{t-1}}\beta_t}{1 - \alpha_t} x_0 + \frac{\sqrt{\alpha_t}(1 - \alpha_{t-1})}{1 - \alpha_t} x_t, \frac{1 - \alpha_{t-1}}{1 - \alpha_t} \beta_t I)$$

1. We cannot change it to the KL divergence form directly.
2. We don’t know $q(x_{t-1}|x_t)$.

Therefore, we change it into a gaussian distribution $q(x_{t-1}|x_t, x_0)$, and make it in the form of KL divergence.
Diffusion Models: Training Objective

\[ L \triangleq \mathbb{E}_{x_0:T \sim q} \left[ -\log p_\theta (x_T) - \sum_{t=1}^{T} \log \frac{p_\theta (x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})} \right] \]

\[ = \mathbb{E}_{x_0:T \sim q} \left[ -\log p_\theta (x_T) - \sum_{t=2}^{T} \log \frac{p_\theta (x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})} - \log \frac{p_\theta (x_0 \mid x_1)}{q(x_1 \mid x_0)} \right] \]

\[ = \mathbb{E}_{x_0:T \sim q} \left[ -\log p_\theta (x_T) - \sum_{t=2}^{T} \log \frac{p_\theta (x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})} \cdot \frac{q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)} - \log \frac{p_\theta (x_0 \mid x_1)}{q(x_1 \mid x_0)} \right] \]

\[ = \mathbb{E}_{x_0:T \sim q} \left[ -\log \frac{p_\theta (x_T)}{q(x_T \mid x_0)} - \sum_{t=2}^{T} \log \frac{p_\theta (x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} - \log p_\theta (x_0 \mid x_1) \right] \]

\[ = \mathbb{E}_{x_0:T \sim q} \left[ D_{KL}(q(x_T \mid x_0) \parallel p_\theta (x_T)) + \sum_{t=2}^{T} D_{KL}(q(x_{t-1} \mid x_t, x_0) \parallel p_\theta (x_{t-1} \mid x_t)) - \log p_\theta (x_0 \mid x_1) \right] \]

\[ = \mathbb{E}_{x_0:T \sim q} \left[ L_T + \sum_{t=2}^{T} L_{t-1} - L_0 \right] \]

\[ \text{Handle the } q(x_1 \mid x_0) \]

Desired term
Forward Process and $L_T$

$$L_T = D_{KL}(q(x_T|x_0) \parallel p_\theta(x_T))$$

- $q(x_T|x_0)$ converges to standard Gaussian distribution.
- We assume $p_\theta(x_T)$ as the standard Gaussian distribution.
- $\Rightarrow$ Do not have to train $\theta$ in this term.
Reverse Process and $L_{T-1}$

$$L_{t-1} = D_{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t))$$

- $q(x_{t-1}|x_t, x_0) = N\left(x_{t-1}, \frac{\sqrt{\alpha_{t-1}}\beta_t}{(1-\alpha_t)}x_0 + \frac{\sqrt{\alpha_t(1-\alpha_{t-1})}}{1-\alpha_t}x_t, 1-\alpha_{t-1} \beta_t I\right)$

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

- Reparametrize with $x_t(x_0, t) = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon$ for $\epsilon \sim N(0, I)$ and estimate $\epsilon_\theta$ then minimize $L_{t-1}$ can be same as

$$\mathbb{E}_{x_0, \epsilon} \left[ \frac{\beta_t^2}{2\sigma_t^2 \alpha_t(1-\alpha_t)} \| \epsilon - \epsilon_\theta(x_t, t) \|^2 \right]$$

- which resembles denoising score matching over multiple noise scales indexed by $t$. 

Estimate $\mu_\theta(x_t, t)$  
Estimate $\Sigma_\theta(x_t, t)$ or use $\sigma_t^2$  
1. $\mu_\theta(x_t, t)$  
2. $x_0(x_t, t)$  
3. $\epsilon_\theta(x_t, t)$
Simplified Training Objective

• Simplified objective aggregates $L_{t-1}$ and $L_0$.

• Ignoring weight of $t$ and discrete Gaussian distribution.

\[
L_{t-1} = \mathbb{E}_{x_0,\epsilon} \left[ \frac{\beta_t^2}{2\alpha_t^2\alpha_t(1-\alpha_t)} \| \epsilon - \epsilon(\sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t} \epsilon, t) \|^2 \right]
\]

\[
L_{simple}(\theta) := \mathbb{E}_{t,x_0,\epsilon} \left[ \| \epsilon - \epsilon(\sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t} \epsilon, t) \|^2 \right]
\]

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.

<table>
<thead>
<tr>
<th>Objective</th>
<th>IS</th>
<th>FID</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$, learned diagonal $\Sigma$</td>
<td>7.28 ± 0.10</td>
<td>23.69</td>
</tr>
<tr>
<td>$L$, fixed isotropic $\Sigma$</td>
<td>8.06 ± 0.09</td>
<td>13.22</td>
</tr>
<tr>
<td>$| \mu - \mu_\theta |^2$</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>$\epsilon$ prediction (ours)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>$L$, learned diagonal $\Sigma$</td>
<td>7.67 ± 0.13</td>
<td>13.51</td>
</tr>
<tr>
<td>$L$, fixed isotropic $\Sigma$</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>$| \epsilon - \epsilon_\theta |^2$ ($L_{simple}$)</td>
<td>9.46 ± 0.11</td>
<td>3.17</td>
</tr>
</tbody>
</table>

Coefficient = \[
\frac{\beta_t^2}{2\alpha_t^2\alpha_t(1-\alpha_t)}
\]
Data Scaling, Reverse Process Decoder, and $L_0$

$$L_0 = \mathbb{E}_{x_0; \tau \sim q}[-\log p_\theta(x_0 | x_1)]$$

- We assume that image data consists of integers in $\{0, 1, \ldots, 255\}$ scaled linearly to $[-1, 1]$.
- Set the last term of the reverse process to an independent discrete decoder derived from the Gaussian $\mathcal{N}(x_0; \mu_\theta(x_1, 1), \sigma_1^2 I)$.

\[ p_\theta(x_0 | x_1) = \prod_{i=1}^{D} \left( \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}(x; \mu_\theta^i(x_1, 1), \sigma_1^2) \, dx \right) \]

\[ \delta_+(x) = \begin{cases} \infty & \text{if } x = 1 \\ x + \frac{1}{255} & \text{if } x < 1 \end{cases} \quad \delta_-(x) = \begin{cases} -\infty & \text{if } x = -1 \\ x - \frac{1}{255} & \text{if } x > -1 \end{cases} \]

Clipping into $[-1, 1]$

In practice, this term is optimized by MSE Loss. Furthermore, there is no independent decoder.
**Inference Algorithm**

- Although training can be done by single step, inference can not be done by single step.
  - E.g., autoregressive models.

![Figure 2: The directed graphical model considered in this work.](image)

---

**Algorithm 1 Training**

1: repeat
2: \( x_0 \sim q(x_0) \)
3: \( t \sim \text{Uniform}\{1, \ldots, T\} \)
4: \( \epsilon \sim \mathcal{N}(0, I) \)
5: Take gradient descent step on
   \[ \nabla_\theta \| \epsilon - \epsilon_\theta(\sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon, t) \|^2 \]
6: until converged

**Algorithm 2 Sampling**

1: \( x_T \sim \mathcal{N}(0, I) \)
2: for \( t = T, \ldots, 1 \) do
3: \( z \sim \mathcal{N}(0, I) \) if \( t > 1 \), else \( z = 0 \)
4: \( x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z \)
5: end for
6: return \( x_0 \)
Architecture

- U-Net backbone similar to an unmasked PixelCNN++.
- Parameter are shared across time, which is specified to the network using the Transformer sinusoidal positional embedding.
- Use self-attention at the $16 \times 16$ feature map resolution.

Figure 2: Like van den Oord et al. (2016c), our model follows a two-stream (downward, and downward+rightward) convolutional architecture with residual connections; however, there are two significant differences in connectivity. First, our architecture incorporates downsampling and upsampling, such that the inner parts of the network operate over larger spatial scale, increasing computational efficiency. Second, we employ long-range skip-connections, such that each $k$-th layer provides a direct input to the $(K - k)$-th layer, where $K$ is the total number of layers in the network. The network is grouped into sequences of six layers, where most sequences are separated by downsampling or upsampling.
Qualitative Results

Figure 3: LSUN Church samples. FID=7.89

Figure 4: LSUN Bedroom samples. FID=4.90
# Quantitative Results

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

<table>
<thead>
<tr>
<th>Model</th>
<th>IS</th>
<th>FID</th>
<th>NLL Test (Train)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Conditional</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JEM [17]</td>
<td>8.76</td>
<td>38.4</td>
<td></td>
</tr>
<tr>
<td>StyleGAN2 + ADA (v1) [29]</td>
<td>10.06</td>
<td>2.67</td>
<td></td>
</tr>
<tr>
<td><strong>Unconditional</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diffusion (original) [53]</td>
<td></td>
<td></td>
<td>$\leq 5.40$</td>
</tr>
<tr>
<td>Gated PixelCNN [59]</td>
<td>4.60</td>
<td>65.93</td>
<td>3.03 (2.90)</td>
</tr>
<tr>
<td>Sparse Transformer [7]</td>
<td></td>
<td></td>
<td>2.80</td>
</tr>
<tr>
<td>PixelIQN [43]</td>
<td>5.29</td>
<td>49.46</td>
<td></td>
</tr>
<tr>
<td>NCSNv2 [56]</td>
<td></td>
<td></td>
<td>31.75</td>
</tr>
<tr>
<td>NCSN [55]</td>
<td>8.87±0.12</td>
<td>25.32</td>
<td></td>
</tr>
<tr>
<td>SNGAN [39]</td>
<td>8.22±0.05</td>
<td>21.7</td>
<td></td>
</tr>
<tr>
<td>SNGAN-DDLS [4]</td>
<td>9.09±0.10</td>
<td>15.42</td>
<td></td>
</tr>
<tr>
<td>StyleGAN2 + ADA (v1) [29]</td>
<td>9.74±0.05</td>
<td>3.26</td>
<td></td>
</tr>
<tr>
<td><strong>Ours (L, fixed isotropic $\Sigma$)</strong></td>
<td>7.67±0.13</td>
<td>13.51</td>
<td>$\leq 3.70 (3.69)$</td>
</tr>
<tr>
<td><strong>Ours (L_{simple})</strong></td>
<td>9.46±0.11</td>
<td><strong>3.17</strong></td>
<td>$\leq 3.75 (3.72)$</td>
</tr>
</tbody>
</table>
Progressive Coding: Rate-Distortion Theory

- Rate (bits/dim): $L_t + \cdots + L_T$
- Distortion (RMSE): $\sqrt{||x_0 - \hat{x}_0||^2 / D}$

Algorithm 3 Sending $x_0$
1: Send $x_T \sim q(x_T|x_0)$ using $p(x_T)$
2: for $t = T - 1, \ldots, 2, 1$ do
3: Send $x_t \sim q(x_t|x_{t+1}, x_0)$ using $p\theta(x_t|x_{t+1})$
4: end for
5: Send $x_0$ using $p\theta(x_0|x_1)$

Algorithm 4 Receiving
1: Receive $x_T$ using $p(x_T)$
2: for $t = T - 1, \ldots, 1, 0$ do
3: Receive $x_t$ using $p\theta(x_t|x_{t+1})$
4: end for
5: return $x_0$

Figure 5: Unconditional CIFAR10 test set rate-distortion vs. time. Distortion is measured in root mean squared error on a $[0, 255]$ scale. See Table 4 for details.

Decrease steeply
Majority of the bits are indeed allocated to imperceptible distortions
Progressive Coding: Estimate $\hat{x}_0$ directly.

- We can directly estimate $\hat{x}_0$ for each timestep using $\hat{x}_0 = \left( x_t - \sqrt{1 - \bar{a}_t \epsilon_\theta(x_t)} \right) / \sqrt{\bar{a}_t}$.

Figure 6: Unconditional CIFAR10 progressive generation ($\hat{x}_0$ over time, from left to right). Extended samples and sample quality metrics over time in the appendix (Figs. 10 and 14).

Figure 7: When conditioned on the same latent, CelebA-HQ $256 \times 256$ samples share high-level attributes. Bottom-right quadrants are $x_t$, and other quadrants are samples from $p_\theta(x_0|x_t)$. 

Interpolation

- Large $t$ results in coarser and more varied interpolations, with novel samples at $t = 1000$.

Figure 9: Coarse-to-fine interpolations that vary the number of diffusion steps prior to latent mixing.
Contributions

• **Novel generative model** which obtain sample quality similar to ProgressiveGAN (SOTA).
  • A latent variable models training on a weighted variational bound.

• **Connection between diffusion probabilistic models and denoising score matching with Langevin dynamics.**
  • Score matching will be presented next week.

• Naturally admit a progressive lossy decompression scheme.