Image Denoising via Advanced Diffusion Generative Modeling

Speaker: LI Xingyao   Supervisor: Prof. ZHOU Xiang

Department of Mathematics
City University of Hong Kong

April 23, 2024

Contents

Introduction

Introduction: Overview of Generative Models

- Various deep generative models have showcased the capability to generate high-quality samples, including Generative Adversarial Networks (GANs) [1, 2], Normalizing Flows [3], and Variational Autoencoders (VAEs) [4].
- These models aim to learn a distribution that assigns high likelihood to observed data samples, either through adversarial learning or likelihood-based approaches.
- However, these models have certain limitations, leading to the development of alternative techniques such as energy-based modeling [5] and score-based generative modeling [6, 7].

Introduction: Score-Based Generative Models

- Score-based generative modeling approaches, including Score-Matching Langevin Dynamics (SMLD) [6] and Denoising Diffusion Probabilistic Models (DDPM) [8, 9], have made significant progress in generating images [6], audio [10, 11], and other domains.
- SMLD estimates scores at each noise scale using Langevin dynamics and generates samples by sampling decreasing noise scales.
- DDPM models the data distribution through a parameterized Markov chain, gradually degrading the training data by increasing noise levels, then learning to reverse this process to generate samples.
- Furthermore, the use of Stochastic Differential Equations (SDEs) has expanded the capabilities of score-based generative models [7], allowing for new sampling methods.

Analysis of Diffusion Model Principle

Analysis of Diffusion Model Principle [9]

The diffusion model includes two processes:

- Forward process: Referred to as the diffusion process, it involves progressively adding Gaussian noise to an image until it transforms into random noise.
- Reverse process: A parameterized Markov chain that serves as a denoising process, capable of generating data.

Both the forward process and the reverse process are parameterized Markov chains. We will model and solve the diffusion model through variational inference [12].

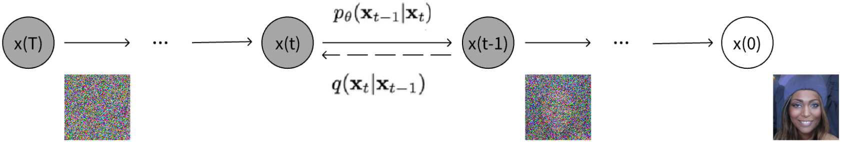
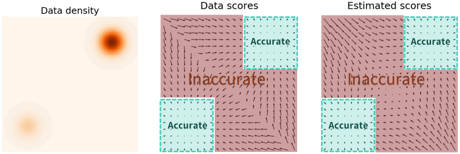


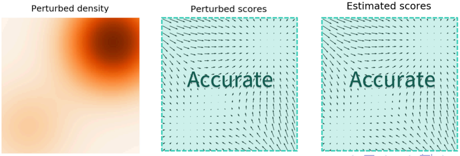
Figure: Diagram of the forward and reverse processes of the DDPM model.

Enhancing Estimation Level with Noise Perturbations in Low Density Regions [13]

- Initial samples often in low-density areas.
- Leads to model inaccuracies.



- Perturb data points with noise.
- Fills in low-density areas.
- Improves model accuracy.



Diffusion Models Distribution

- Take DDPM as an example.
- Diffusion models are latent variable models characterized by:

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

  where $x_1, \ldots, x_T$ are latent variables with the same dimensionality as the data x_0, sampled from $q(x_0)$ [14].
- The joint distribution $p_\theta(x_{0:T})$, referred to as the reverse process, follows a Markov chain with Gaussian transitions:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t),$$
$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)).$$

Diffusion Process

- The approximate posterior $q(x_{1:T}|x_0)$, also known as the forward process or diffusion process, is modeled as a Markov chain that incrementally introduces Gaussian noise to the data:

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

- We can simplify this by setting $\alpha_t = 1 - \beta_t$:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{\alpha_t}x_{t-1}, \beta_t I\right).$$

Diffusion Process

- An essential aspect of the diffusion process is its ability to directly sample x_t at any time step t solely based on the original data x_0.
- To simplify the representation of the diffusion process, we introduce the notation $\bar{\alpha}_t = \sqrt{\alpha_1} \dots \sqrt{\alpha_t}$.

$$
\begin{aligned}
x_t &= \sqrt{\alpha_t} x_{t-1} + \sqrt{\beta_t} \epsilon_t \\
&= \sqrt{\alpha_t} \left(\sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{\beta_{t-1}} \epsilon_{t-1} \right) + \sqrt{\beta_t} \epsilon_t \\
&= \cdots \\
&= \left(\sqrt{\alpha_t} \cdots \sqrt{\alpha_1} \right) x_0 + \sum_{i=1}^{t} \underbrace{\left(\sqrt{\alpha_t} \cdots \sqrt{\alpha_{i+1}} \sqrt{\beta_i} \epsilon_i \right)}_{\sim \mathcal{N}\left(0, \left(1 - \bar{\alpha}_t^2\right) \mathbf{I}\right)} \\
&= \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \qquad \quad for \quad \epsilon \sim \mathcal{N}(0, I).
\end{aligned}
$$

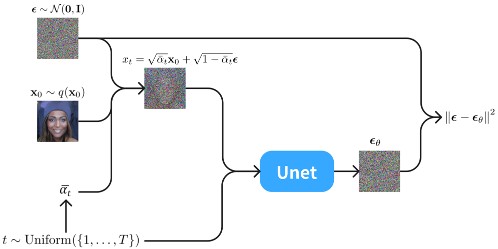
- After anti-reparameterization, we get:

$$
q(x_t | x_0) = \mathcal{N} \left(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) \mathbf{I} \right).
$$

Diffusion Process

Algorithm Diffusion Algorithm [9]

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



Generation Process

- The reverse process functions as a denoising procedure. Given knowledge of the genuine distribution $p(x_{t-1}|x_t)$ at each step of the reverse process, starting with a random noise $x_T \sim \mathcal{N}(0, I)$, gradual denoising can produce an authentic sample. Therefore, the reverse process also serves as the generation process.
- Applying Bayes' theorem, we have:

$$p(x_{t-1}|x_t) = \frac{p(x_t|x_{t-1}) \cdot p(x_{t-1})}{p(x_t)}.$$

- The issue arises as $p(x_{t-1}), p(x_t)$ are unknown, as x_{t-1} cannot be directly inferred from x_t. However, inference can be conducted under the condition of x_0.
- Therefore, by applying Bayes' theorem, we can compute the conditional probability $p(x_{t-1}|x_t, x_0)$.

Generation Process

The Formula shows how we revise our understanding of x_{t-1} given both the prior information x_0 and the observation x_t.

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

Based on the previous derivation, we can express the three distributions:

1. $p(x_{t-1}|x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}x_0, \sqrt{1-\bar{\alpha}_{t-1}}\epsilon\right)$
2. $p(x_t|x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0, \sqrt{1-\bar{\alpha}_t}\epsilon\right)$
3. $p(x_t|x_{t-1}) = \mathcal{N}\left(\sqrt{\alpha_t}x_{t-1}, \sqrt{1-\bar{\alpha}_t}\epsilon\right)$

Generation Process

- Then, we obtain:

$$
\begin{aligned}
P(x_{t-1}|x_t, x_0) &\propto \exp\left(-\frac{1}{2}\left(\frac{(x_t - \sqrt{\alpha_t}x_{t-1})^2}{\beta_t}\right) + \frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t}\right) \\
&= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)x_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)x_{t-1} + C(x_t, x_0)\right)\right).
\end{aligned}
$$

- Hence, the mean and variance of the posterior distribution $p(x_{t-1}|x_t, x_0)$ are:

$$
\begin{aligned}
\mu &= \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0, \\
\sigma^2 &= \frac{\beta_t\left(1-\bar{\alpha}_{t-1}\right)}{\alpha_t\left(1-\bar{\alpha}_{t-1}\right)+\beta_t}.
\end{aligned}
$$

Generation Process

As x_0 is unknown, we can choose an unknown function such that $x_0 = f(x_t)$. However, observing the previous equations, we can infer that the formula for deriving x_t from x_0 is the inverse of a satisfying formula:

$$x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_t \right).$$

Next, substitute into the expression for the mean:

$$\mu = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_t \right).$$

Since ϵ represents noise of unknown origin, we employ a neural network, denoted as $\epsilon_\theta(x_t, t)$, to model it. This enables us to define our optimization objective as minimizing the discrepancy between the true noise and the noise predicted by the neural network at each step, expressed as $\|\epsilon_t - \epsilon_\theta(x_t, t)\|^2$.

Generation Process

The final formula is then given by:

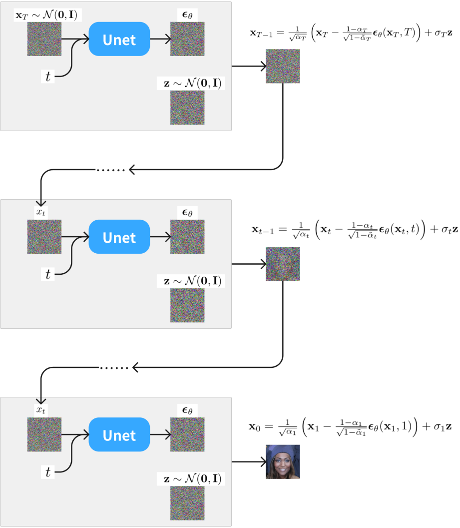
$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha} t}} \epsilon_\theta(x_t, t) \right) + \sigma_t \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, I)$, and $\beta_t = 1 - \alpha_t$.

Algorithm 2 Sampling Algorithm

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T$ to 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** $\mathbf{x}_0$

Generation Process



Score-based generative modeling with SDEs

From Variational Diffusion Models to Score-based Generative Models

- By analyzing the derivation of the Variational Diffusion Model (VDM), we find an equivalence with Score-based Generative Models [15].
- This enables us to employ neural networks to estimate the score function, allowing flexible modeling of data distributions. Flexible probability distributions are expressed in the form:

$$p_\theta(x) = \frac{1}{Z_\theta} e^{-f_\theta(x)},$$

  where $f_\theta(x)$ is a flexible, parameterizable function (often modeled by a neural network), and Z_θ is a normalizing constant.

Score Function Estimation

- Computing the normalizing constant Z_θ is challenging in maximum likelihood estimation.
- To avoid this, we utilize a neural network $s_\theta(x)$ to estimate the score function $\nabla_x \log p(x)$ of distribution $p(x)$:

$$\begin{aligned}\nabla_x \log p_\theta(x) &= \nabla_x \log\left(\frac{1}{Z_\theta}e^{-f_\theta(x)}\right)\\ &= \nabla_x \log\frac{1}{Z_\theta} + \nabla_x \log e^{-f_\theta(x)}\\ &= -\nabla_x f_\theta(x)\\ &\approx s_\theta(x).\end{aligned}$$

- Optimize the score model by minimizing the Fisher Divergence with the ground truth score function:

$$\mathbb{E}_{p(x)}\left[\|s_\theta(x) - \nabla \log p(x)\|_2^2\right].$$

Score-based generative modeling with SDEs

- Denoising Diffusion Probabilistic Models
  - Rely on normalization techniques like denoising diffusion
  - Normalization is necessary to generate high-quality images
- Score-based generative modeling with Stochastic Differential Equations
  - Do not require normalization steps
  - Offer a different approach to modeling and generating data compared to DDPMs

Stochastic Denoising Score Matching with Langevin Dynamics (SMLD) [6]

- Utilizes a Noise Conditional Score Network (NCSN) $s_{\boldsymbol{\theta}}(\mathbf{x}, \sigma)$ to approximate $\log p_\sigma(\mathbf{x})$.
- Optimizes a sum of denoising score matching objectives weighted by the variance σ_i^2 across different noise scales σ_i^2:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \sigma_i^2 \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{p_{\sigma_i}(\tilde{\mathbf{x}}|\mathbf{x})} \left[\| s_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}, \sigma_i) - \nabla_{\tilde{\mathbf{x}}} \log p_{\sigma_i}(\tilde{\mathbf{x}}|\mathbf{x}) \|_2^2 \right].$$

- Generates samples using Langevin MCMC iteratively:

$$\mathbf{x}_i^m = \mathbf{x}_i^{m-1} + \epsilon_i s_{\boldsymbol{\theta}^*}(\mathbf{x}_i^{m-1}, \sigma_i) + \sqrt{2\epsilon_i}\mathbf{z}_i^m, \quad m = 1, 2, \ldots, M.$$

- As $M \to \infty$ and $\epsilon_i \to 0$ for all i, $\mathbf{x}_1^M$ approximates an exact sample from $p_{\sigma_{\min}}(\mathbf{x})$ under certain conditions, ensuring an accurate representation of the data distribution.

Score Matching and Score-based Model [6]

- The score of a distribution can be estimated by training a score-based model on samples using score matching.
- We can train a time-dependent score-based model $s_\theta(x, t)$ by extending the optimization objective to a continuous setting.
- This involves minimizing the following objective function:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_t \Big\{ \lambda(t) \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)} \Big[\big\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0)) \big\|_2^2 \Big] \Big\}.$$

- Next, we will introduce in detail how to use SDE for data perturbation and sample generation.

Stochastic Differential Equations for Data Perturbation

- In accordance with the exposition provided by Song [7], our objective is to formulate a diffusion process $x(t)_{t=0}^{T}$ governed by an Itô SDE:

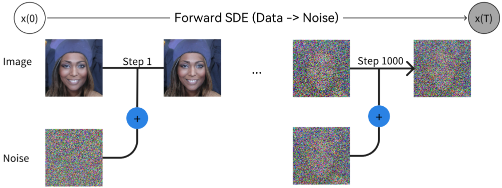
$$\mathrm{d}x = f(x,t)\mathrm{d}t + g(t)\mathrm{d}w,$$

- where $x(0)$ follows the distribution p_0, inferred from an i.i.d. dataset, and $x(T)$ follows a tractable prior distribution p_T, amenable to efficient sampling. Here, $f(x,t)$ denotes the *drift* coefficient, $g(t)$ represents the *diffusion* coefficient, and $\mathrm{d}w$ stands for the Wiener increment, representing infinitesimal changes in the Wiener process (a.k.a., Brownian motion).
- The transition from x_t to $x_{t+\Delta t}$ can be expressed as:

$$\begin{aligned} x_{t+\Delta t} - x_t &= f\left(x_t, t\right)\Delta t + g(t)\sqrt{\Delta t}\epsilon, \\ x_{t+\Delta t} &= x_t + f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\epsilon. \end{aligned}$$

- The conditional probability $p(x_{t+\Delta t}|x_t)$ is given by:

$$p(x_{t+\Delta t}|x_t) \sim \mathcal{N}(x_t + f(x_t, t)\Delta t, g^2(t)\Delta t I).$$



Reversing Stochastic Differential Equations for Sample Generation

- Transition back from $x_{t+\Delta t}$ to x_t:

$$
\begin{aligned}
p\left(x_t \mid x_{t+\Delta t}\right) &= \frac{p\left(x_{t+\Delta t} \mid x_t\right) p\left(x_t\right)}{p\left(x_{t+\Delta t}\right)} \\
&= p\left(x_{t+\Delta t} \mid x_t\right) \exp\left\{\log p\left(x_t\right) - \log p\left(x_{t+\Delta t}\right)\right\}.
\end{aligned}
$$

- Using Bayes' rule and Taylor expansion:

$$
\begin{aligned}
p\left(x_t \mid x_{t+\Delta t}\right) \approx \exp\Bigg\{-\frac{1}{2g^2(t+\Delta t)\Delta t} \| &\left(x_{t+\Delta t} - x_t\right) - \left(f\left(x_{t+\Delta t}, t+\Delta t\right)\right. \\
&\left. - g^2(t+\Delta t)\ \nabla_{x_{t+\Delta t}} \log p\left(x_{t+\Delta t}\right) \Delta t \right\|_2^2 \Bigg\}.
\end{aligned}
$$

- The mean and variance of the conditional distribution $p(x_t \mid x_{t+\Delta t})$:

$$
\begin{aligned}
\mu &= x_{t+\Delta t} - \left(f\left(x_{t+\Delta t}, t+\Delta t\right) - g^2(t+\Delta t) \nabla_{t+\Delta t} \log p\left(x_{t+\Delta t}\right)\right) \Delta t, \\
\sigma^2 &= g^2(t+\Delta t)\Delta t.
\end{aligned}
$$

Reversing Stochastic Differential Equations for Sample Generation

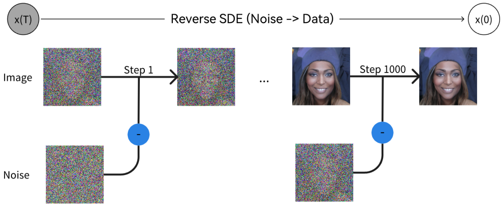
- Hence, we have

$$x_{t+\Delta t} - x_t = \left[f\left(x_{t+\Delta t}, t+\Delta t\right) - g^2(t+\Delta t)\nabla_{x_{t+\Delta t}} \log p\left(x_{t+\Delta t}\right)\right]\Delta t \\ + g(t+\Delta t)\sqrt{\Delta t}\epsilon.$$

- Finally, the reverse-time SDE for the generation process is:

$$dx = \left[f(x,t) - g^2(t)\nabla_{x_t} \log p(x_t)\right] dt + g(t) d\omega,$$

- Once the score $\nabla_{x_t} \log p(x_t)$ is known for all t, the reverse diffusion process can be simulated to sample from p_0.



Exploring VE and VP SDEs

- VE-SDE (SMLD):
  - Variance-exploding process
  - $x_t = x_0 + \sigma_t \epsilon$, $x_{t+1} = x_t + \sqrt{\sigma_{t+1}^2 - \sigma_t^2}\epsilon$.
  - $d\mathbf{x} = \sqrt{\frac{d(\sigma^2(t))}{dt}} d\mathbf{w}$, with $f(x_t, t) = 0$ and $g(t) = \frac{d}{dt}\sigma_t^2$.
- VP-SDE (DDPM):
  - Variance-preserving process ($\sqrt{1-\bar{\alpha}_T} \approx 1$)
  - $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, x_{t+1} = \sqrt{1-\beta_{t+1}} x_t + \sqrt{\beta_{t+1}}\epsilon$.
  - $d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}$, with $f(x_t, t) = -\frac{1}{2}\beta(t)x_t$ and $g(t) = \sqrt{\beta(t)}$.

Equivalence between VE-SDE and VP-SDE

- **VE:**

$$x_t = x_0 + \sigma_t \epsilon,$$
$$\frac{x_t}{\sqrt{1+\sigma_t^2}} = \frac{x_0}{\sqrt{1+\sigma_t^2}} + \frac{\sigma_t}{\sqrt{1+\sigma_t^2}}\epsilon.$$

- **VP:**

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon.$$

- Correspondence: $\sqrt{\bar{\alpha}t} \equiv \frac{1}{\sqrt{1+\sigma_t^2}}$ and $\sqrt{1-\bar{\alpha}t} \equiv \frac{\sigma t}{\sqrt{1+\sigma_t^2}}$

Experimental Methodology

Datasets Used

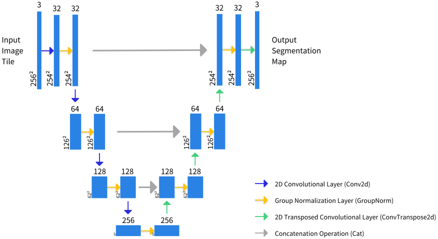
- MNIST [16]: A widely used handwritten digit dataset with 60,000 training and 10,000 test examples
- Swiss Roll [17]: A synthetic dataset with a twisted two-dimensional manifold structure, used for evaluating dimensionality reduction algorithms
- FFHQ (Flickr-Faces-HQ) [18]: A high-quality facial image dataset with 70,000 images at 1024$\times$1024 resolution, featuring diverse attributes

Implementation Details

- Utilized TensorFlow 2.4.0 and PyTorch 1.7.0 on Tesla V100 GPUs
- Conducted initial experiments on MNIST and Swiss Roll, then extensively explored the FFHQ dataset
- Preprocessed FFHQ images by resizing to 256×256 pixels and applying augmentations (random horizontal flips, rotations)
- Set batch size to 32, used Adam optimizer, and applied linear learning rate warm-up for the first 5000 iterations
- Systematically saved model checkpoints and logged progress/performance every 100 iterations
- Leveraged the Probability Counting sampling method for robust sampling strategies
- Employed Group Normalization and Swish activation function to ensure training stability

Network Architecture

- Integrated 2D Convolutional Layers, Group Normalization Layers, and 2D Transposed Convolutional Layers
- Utilized tensor concatenation operations for feature map fusion
- Encoded time information using a GaussianFourierProjection layer and a Linear layer
- Implemented a U-Net-like architecture with encoding and decoding paths, preserving spatial information through skip connections



Evaluation Metrics

- Employed loss functions tailored for training score-based generative models [19], including:
  - Loss for training with arbitrary SDEs

$$\mathrm{loss}_{\mathrm{SDE}} = \mathrm{mean}\left(\sum_i \left((\mathrm{score}_i \cdot \sigma_i + \mathbf{z}_i)^2\right)\right).$$

  - Loss for training with VE-SDEs

$$\mathrm{loss}_{\mathrm{VESDE}} = \mathrm{mean}\left(\sum_i \left((\mathrm{score}_i - \mathrm{target}_i)^2\right) \cdot \sigma_i^2\right).$$

  - Loss for training with VP-SDEs

$$\mathrm{loss}_{\mathrm{VPSDE}} = \mathrm{mean}\left(\sum_i \left((\mathrm{score}_i - \mathrm{noise}_i)^2\right)\right).$$

Evaluation Metrics

Utilized the Fréchet Inception Distance (FID) metric to evaluate the quality of generated images [20]. FID measures the discrepancy between two multivariate normal distributions estimated from features extracted from real and generated images based on the U-Net architecture. It is calculated using the formula:

$$\mathsf{FID} = \|\mu_r - \mu_g\|^2 + \mathsf{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}).$$

Here, μ_r and μ_g are the means, and Σ_r and Σ_g are the covariance matrices of the real and generated distributions, respectively.

Results Analysis

Generation Process: Swissroll

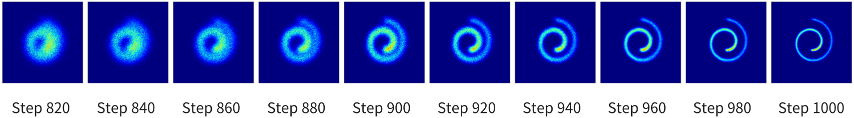


Figure: Denoising Process of the Swissroll Dataset.

- A preliminary experiment showcases the denoising process using the Swissroll dataset.
- From step 820 to step 1000, the denoising process gradually refines the generated images.
- This enhancement in clarity and fidelity illustrates the effectiveness of the Swissroll denoising process.

Generation Process: FFHQ

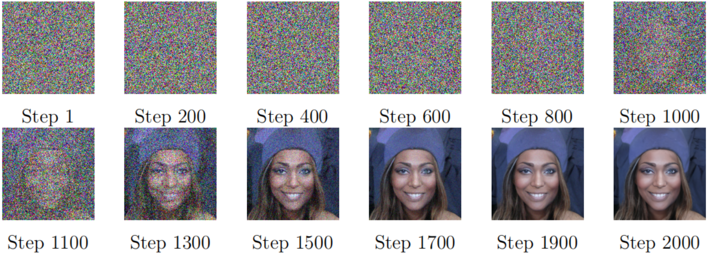


Figure: Denoising Process of FFHQ Dataset with NCSNPP (Continuous).

- Further experiments demonstrate the denoising process using the FFHQ dataset.
- From step 1 (random noise) to step 2000, the generated images become sharper and more detailed, closely approximating the true data distribution.
- The gradual improvement in image quality illustrates the effectiveness of the DDPM denoising process.

Model Performance: Training and Validation Loss

- Training and validation loss curves exhibited a consistent trend
- Rapid initial decrease, indicating significant improvement in learning
- Gradual stabilization and convergence to a relatively steady state

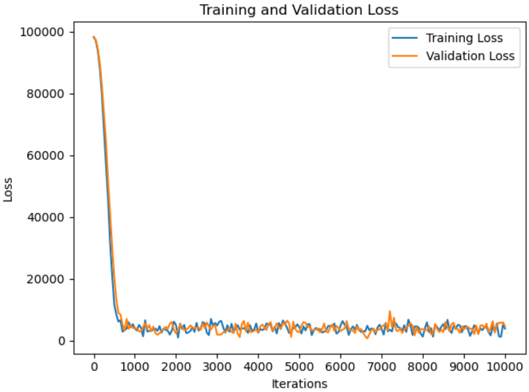


Figure: Training and Validation Loss of the Training Process.

Model Performance: FID Evaluation

- Final FID score on FFHQ dataset: 4.962341
- Indicates a high degree of similarity between generated and real image distributions
- FID value is higher than that obtained by Song et al. on CIFAR-10, but FFHQ is a more complex dataset
- Suggests the samples generated by our model are of high quality

Dataset	Method	FID
FFHQ (Ours)	DDPM	4.96
CIFAR-10 (Song Yang's)	DDPM	3.37

Sample Generation Results

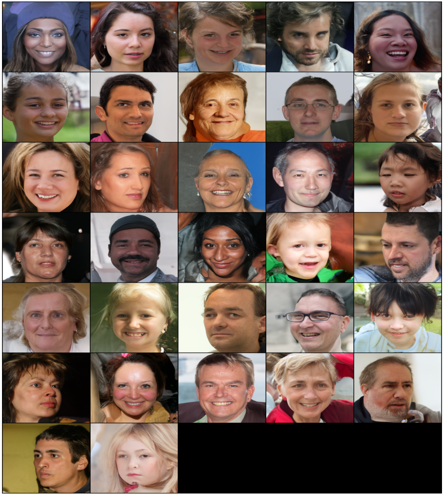


Figure: Sample Generation Results.

Conclusion

Conclusion

- In summary, this project provides an analysis of Denoising Diffusion Probabilistic Models (DDPMs) and Score-based models utilizing Stochastic Differential Equations (SDEs). We have derived precisely the principles underlying DDPM and SDE, elucidating their significance in image generation. Furthermore, our experiments demonstrate the stability of score-based models in sample generation, shedding light on their practical implementation.
- Looking ahead, while our proposed methodologies have demonstrated promising results, there exist opportunities for further exploration, particularly in balancing the stability of score-based models with the efficient sampling capabilities of implicit models like GANs. Enhancing the speed of sampling processes is a crucial aspect that warrants attention in future research endeavors.

References I

[1] Ian J. Goodfellow et al. "Generative Adversarial Networks". In: *arXiv preprint arXiv:1406.2661* (2014). Focus to learn more. DOI: 10.48550/arXiv.1406.2661. URL: https://arxiv.org/abs/1406.2661.

[2] Tero Karras et al. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=Hk99zCeAb.

[3] Diederik P. Kingma and Prafulla Dhariwal. "Glow: Generative Flow with Invertible 1x1 Convolutions". In: *arXiv preprint arXiv:1807.03039* (2018). DOI: 10.48550/arXiv.1807.03039. URL: https://arxiv.org/abs/1807.03039.

[4] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. "Generating Diverse High-Fidelity Images with VQ-VAE-2". In: *arXiv preprint arXiv:1906.00446* (2019). DOI: 10.48550/arXiv.1906.00446. URL: https://arxiv.org/abs/1906.00446.

References II

[5] Yilun Du and Igor Mordatch. "Implicit Generation and Modeling with Energy Based Models". In: *Neural Information Processing Systems*. 2019. URL: https://api.semanticscholar.org/CorpusID:202765664.

[6] Yang Song and Stefano Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution". In: *Advances in Neural Information Processing Systems*. Modified: May 6, 2023. 2019. URL: https://github.com/ermongroup/ncsn.

[7] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=PxTIG12RRHS.

[8] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *ICML*. 2015, pp. 2256–2265. URL: http://proceedings.mlr.press/v37/sohl-dickstein15.html.

References III

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *arXiv preprint arXiv:2006.11239* (2020). [Submitted on 19 Jun 2020 (v1), last revised 16 Dec 2020 (this version, v2)]. DOI: 10.48550/arXiv.2006.11239. arXiv: 2006.11239 [cs.LG]. URL: https://arxiv.org/abs/2006.11239.

[10] Nanxin Chen et al. "WaveGrad: Estimating Gradients for Waveform Generation". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=NsMLjcFaO8O.

[11] Zhifeng Kong et al. "DiffWave: A Versatile Diffusion Model for Audio Synthesis". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=a-xFK8Ymz5J.

[12] Ankush Ganguly and Samuel W. F. Earp. *An Introduction to Variational Inference*. 2021. arXiv: 2108.13083 [cs.LG].

[13] Yang Song. *Generative Modeling by Estimating Gradients of the Data Distribution | Yang Song*. 2021. URL: https://yang-song.net/blog/2021/score/.

References IV

[14] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 2256–2265. URL: https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[15] Calvin Luo. "Understanding Diffusion Models: A Unified Perspective". In: *arXiv preprint arXiv:2208.11970* (2022). DOI: 10.48550/arXiv.2208.11970. URL: https://arxiv.org/abs/2208.11970.

[16] Yann LeCun, Corinna Cortes, and CJ Burges. "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (2010).

[17] Stephen Marsland. *Machine Learning: An Algorithmic Perspective*. https://homepages.ecs.vuw.ac.nz/~marslast/Code/Ch6/lle.py. 2014.

References V

[18] Tero Karras and Janne Hellsten. *NVlabs/Ffhq-Dataset*. https://github.com/NVlabs/ffhq-dataset. 2021.

[19] Yang Song. *Score_sde_pytorch/Losses.py at Main · Yang-Song/Score_sde_pytorch*. github.com/yang-song/score_sde_pytorch/blob/main/losses.py. 2021.

[20] Wikipedia Contributors. *Fréchet Inception Distance*. 2021. URL: https://en.wikipedia.org/wiki/Fr%C3%A9chet_inception_distance.

[21] Brian D.O. Anderson. "Reverse-time diffusion equation models". In: *Journal of Mathematical Analysis and Applications* 87.2 (1982). Received November 13, 1979; Revised June 12, 1980; Available online April 2, 2002., pp. 592–602. DOI: 10.1016/0304-4149(82)90051-5. URL: https://doi.org/10.1016/0304-4149(82)90051-5.

References VI

[22] Chin-Wei Huang, Jae Hyun Lim, and Aaron C. Courville. "A Variational Perspective on Diffusion-Based Generative Models and Score Matching". In: *CoRR* abs/2106.02808 (2021). arXiv: 2106.02808. URL: https://arxiv.org/abs/2106.02808.

[23] Yann LeCun et al. "A Tutorial on Energy-Based Learning". In: 2006. URL: https://api.semanticscholar.org/CorpusID:8531544.

[24] Tero Karras et al. *Elucidating the Design Space of Diffusion-Based Generative Models*. 2022. arXiv: 2206.00364 [cs.CV].

Thanks for watching!
Q&A