EE660 Final Project

Yuming Li

December 2024

Generative Models

Denoising score matching

An enhanced version of DSM loss is introduced by employing multiple noise realizations for each data point x_i. Let k be a fixed positive integer, and define $\tilde{x}_i^j = x_i + \sigma w_i^j$, where $\{w_i^j\}_{i,j=1}^{n,k}$ are i.i.d $N(0, I)$ noise vectors. This results in a variance-reduced loss, as shown in the following.

$$\hat{L}_n(\theta) = \frac{1}{nk\sigma^4} \sum_{i=1}^{n} \sum_{j=1}^{k} \|\tilde{x}_i^j + \sigma^2 f_0(\tilde{x}_i^j) - x_i\|^2. \tag{1}$$

Denoising Diffusion Probabilistic Models

The algorithms for denoising diffusion probabilistic models [1] are shown below:

Algorithm 1 Training

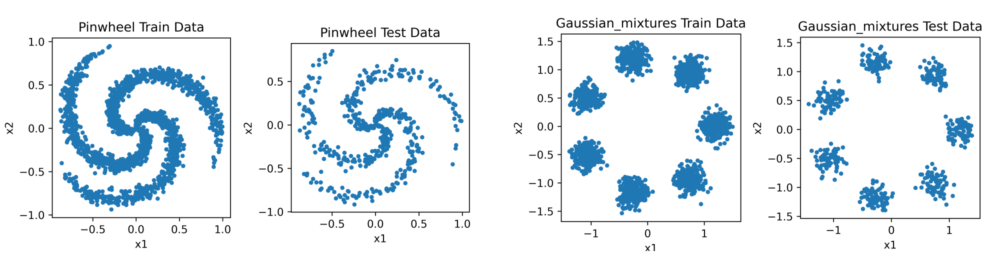
repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
    $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0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

Algorithm 2 Sampling

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

Dataset Usage

In this project, as depicted in Figure 1, datasets consisting of two-dimensional Pinwheel and Gaussian mixtures were developed, each partitioned into 2000 training samples and 500 test samples.

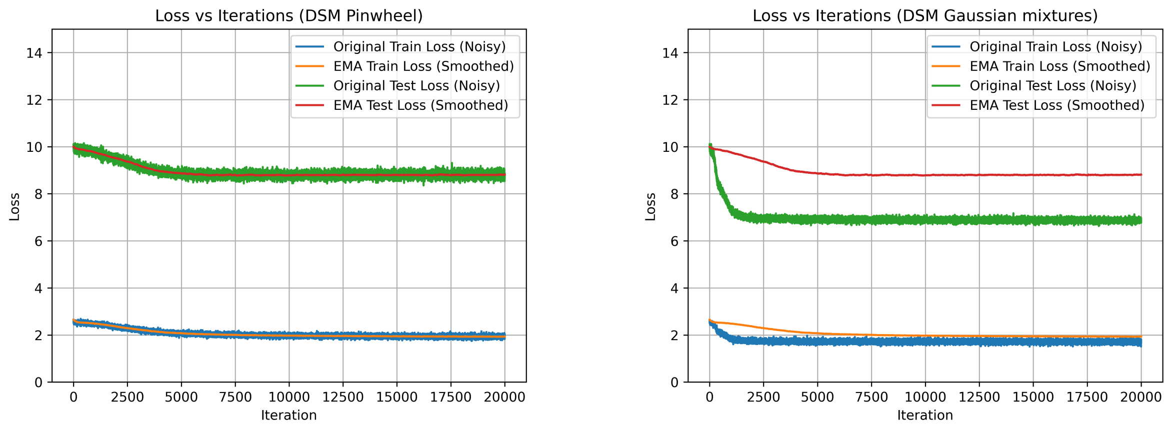


(a) Pinwheel Dataset (b) Gaussian mixtures Dataset

Figure 1: Dataset Visualization

Training/Test Curves

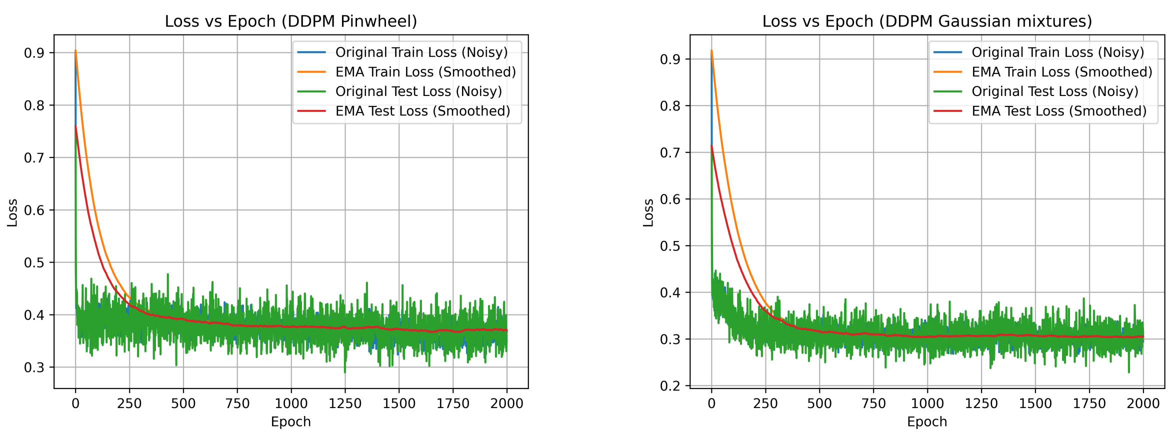
The DSM is trained with $\sigma = 0.1$, $k = 20$, using a 4-layer MLP (hidden size 64), over 20000 iterations with a batch size of 128 and a learning rate of 1×10^{-3}.



(a) Pinwheel Dataset (b) Gaussian mixtures Dataset

Figure 2: DSM Training/Test Loss

The DDPM is trained over $T = 500$ steps with linear noise scheduling (β from 0.0001 to 0.02), a 4-layer MLP (hidden size 64), time embeddings (dimension 32), and a batch size of 64, across 2000 epochs at a learning rate of 1×10^{-3}.



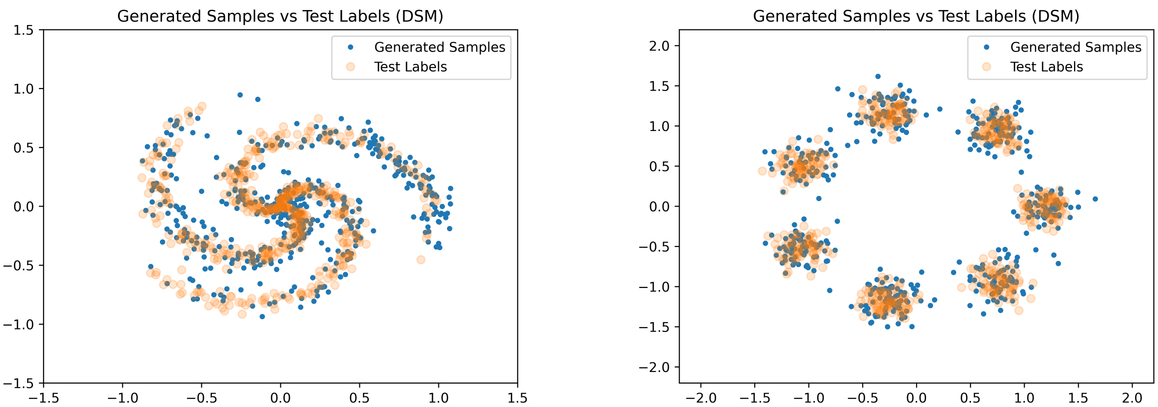
(a) Pinwheel Dataset (b) Gaussian mixtures Dataset

Figure 3: DDPM Training/Test Loss

Figures 2 and 3 show decreasing training and test losses for both DSM and DDPM, indicating stable training. DSM exhibits a higher test than training loss, indicative of inadequate generalization. Conversely, DDPM shows a narrower gap between its training and test losses, reflecting superior generalization. An exponential moving average (EMA) with $\alpha = 0.01$ smooths the loss curves, particularly useful for DSM's fluctuating losses.

Qualitative comparison

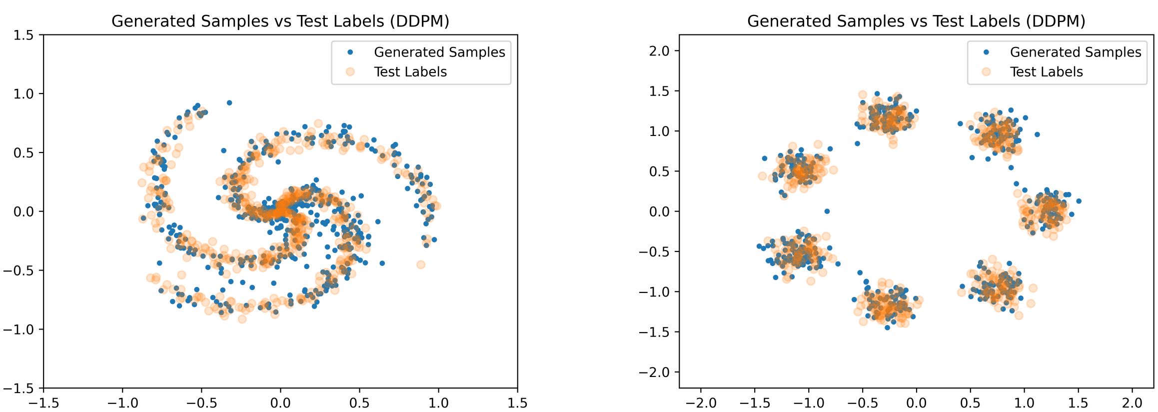
Figures 4 and 5 display the synthesis of 500 data points using DSM and DDPM, compared against test data for the Pinwheel and Gaussian mixtures datasets. The generated samples closely match the test distributions, with left and right panels for the Pinwheel and Gaussian mixtures, respectively. Crucially, normalizing data before training and denormalizing after generation is essential for accurately replicating the Pinwheel dataset scale. These results highlight the effectiveness of both methods in capturing the true data distributions.



(a) Pinwheel Dataset

(b) Gaussian mixtures Dataset

Figure 4: DSM Samples Generated



(a) Pinwheel Dataset

(b) Gaussian mixtures Dataset

Figure 5: DDPM Samples Generated

Although both models perform well in fitting the data distributions, a closer comparison reveals that DDPM generates data points that are more tightly clustered around the test labels. This indicates that DDPM achieves a higher degree of fidelity and better overall alignment with the target distribution. The sharper clustering of DDPM's samples highlights its superior modeling capacity compared to DSM.

Quantitative comparison

To quantitatively assess the ability of models to accurately fit data distributions, we employed Gaussian Kernel Density Estimation (KDE) to transform the discrete data points of both real and generated datasets into continuous probability density estimates. This transformation facilitates the robust calculation of log-likelihoods. The KDE formula utilizing a Gaussian kernel is expressed as follows:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{2}$$

where $K(\cdot)$ represents the Gaussian kernel, h denotes the bandwidth, and x_i are the individual data points.

Subsequently, we computed the log-likelihoods for both datasets in a bidirectional manner:

- M1 and M2: Log-likelihoods of real data evaluated on the generated data model.
- M3 and M4: Log-likelihoods of generated data evaluated on the real data model.

This bidirectional evaluation not only assesses the accuracy with which the models represent the true data distributions but also examines the plausibility of the generated data under these distributions. The outcomes are summarized in Table 1.

	M1	M2	M3	M4
DSM (Pinwheel)	-436.94	-0.87388	-469.65	-0.93929
DSM (Gaussian mixtures)	-868.36	-1.7367	-870.79	-1.7416
DDPM (Pinwheel)	-396.20	-0.79240	-393.58	-0.78715
DDPM (Gaussian mixtures)	-854.65	-1.7093	-844.44	-1.6889

Table 1: Log-likelihood comparison between DSM and DDPM on Pinwheel and Gaussian mixtures datasets. M1 and M2 refer to log-likelihoods of real data under generated data (sum and mean, respectively), and M3 and M4 refer to log-likelihoods of generated data under the real data (sum and mean, respectively).

For the Pinwheel dataset, the DDPM model consistently achieves higher (less negative) log-likelihoods compared to the DSM model across all metrics (M1, M2, M3, and M4), indicating a superior ability to fit and generate data distributions.

In the case of the Gaussian mixtures dataset, although both models show close performance, DDPM still slightly surpasses DSM, demonstrating a more robust capability in capturing complex data distributions, as evidenced by the consistently higher log-likelihood values across all evaluated metrics.

References

[1] Jonathan Ho, Ajay Jain, Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. Neural Information Processing Systems, 2020.