# Deep Learning in Practice 4th Coding Assignment

Zhongke Sun
zhongke.sun@student-cs.fr

March 19, 2025

## 1 Experimental Data

This lab aims to implement a small-scale diffusion model and evaluate on a synthetic dataset.

- 1D Dataset: The 1D dataset consists of samples that originate from two distinct regions along the x-axis. A binary random variable $\gamma$ determines whether a point is drawn from the positive or negative range, with additional random perturbations applied to introduce slight variability.
- 2D Dataset: The 2D dataset extends the 1D approach by generating points in a two-dimensional space

## 2 Forward Process

The forward process in diffusion models describes the stepwise transformation of an initial data point $x_0$ into pure noise through a series of incremental perturbations. This is achieved by gradually adding Gaussian noise to the data across $T$ timesteps.

At each step $t$, the transition follows the update rule:

$$x_t = \sqrt{1-\beta_t} \cdot x_{t-1} + \sqrt{\beta_t} \cdot \epsilon \tag{1}$$

where:

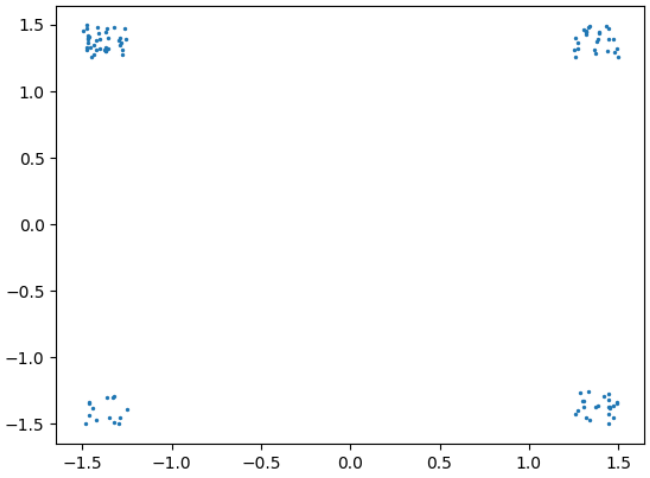



Figure 1: Data Visualization - 2D Data

- $\beta_t$ is a predefined noise schedule, controlling the variance of the added noise.
- $\epsilon \sim N(0, 1)$ represents a standard Gaussian noise term.

Instead of iterating through every step sequentially, we can derive a closed-form solution to compute $x_t$ directly from the original data $x_0$. By recursively applying the transition equation, we obtain:

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

where: $\bar{\alpha}t = \prod s = 1^t \alpha_s, \quad \alpha_t = 1 - \beta_t$

## 2.1 1D Visualization

The following left plot shows the distribution of values through time, and the right plot visualizes the distribution of values computed directly from $x_0$.

From the figures, we can see that the two methods share a similar trend. We will calculate means and variances to show whether they have a similar distribution.

According to the sanity check, we can verify that the two diffusion processes have a similar behavior.

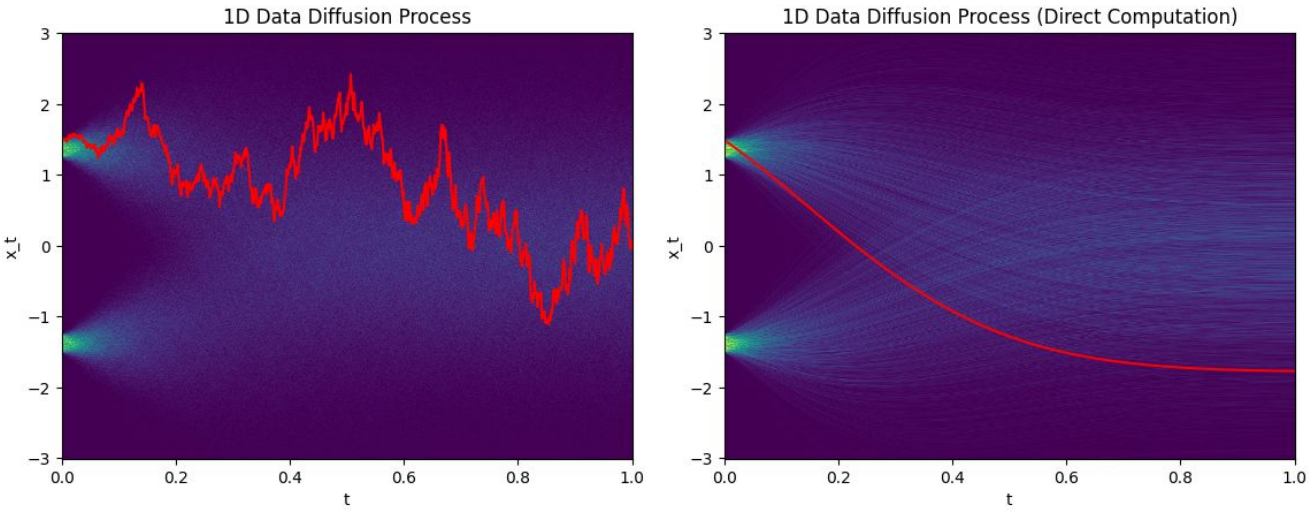



Figure 2: Visualizations of 1D Data Diffusion Process (Iterative and Direct computation)

| Timestep | Stepwise Mean | Direct Mean | Stepwise Variance | Direct Variance |
|---|---|---|---|---|
| **0** | 0.015819 | 0.015803 | 1.893485 | 1.893675 |
| **200** | 0.013098 | 0.010966 | 1.600442 | 1.617933 |
| **400** | -0.010352 | 0.004033 | 1.177794 | 1.185542 |
| **600** | 0.001836 | -0.000759 | 1.039445 | 1.012917 |
| **800** | -0.009923 | -0.002751 | 0.976233 | 0.982628 |

Table 1: Comparison of means and variances computed with iterative and direct methods (For each timestep)

| Method | Mean | Variance |
|---|---|---|
| **Iterative Diffusion** | -0.001150 | 1.247865 |
| **Direct Diffusion** | 0.003526 | 1.247960 |

Table 2: Comparison of means and variances computed with iterative and direct methods (Across all timestep)

## 2.2 2D Visualization

To visualize the forward diffusion process in 2D, we perform the following steps:

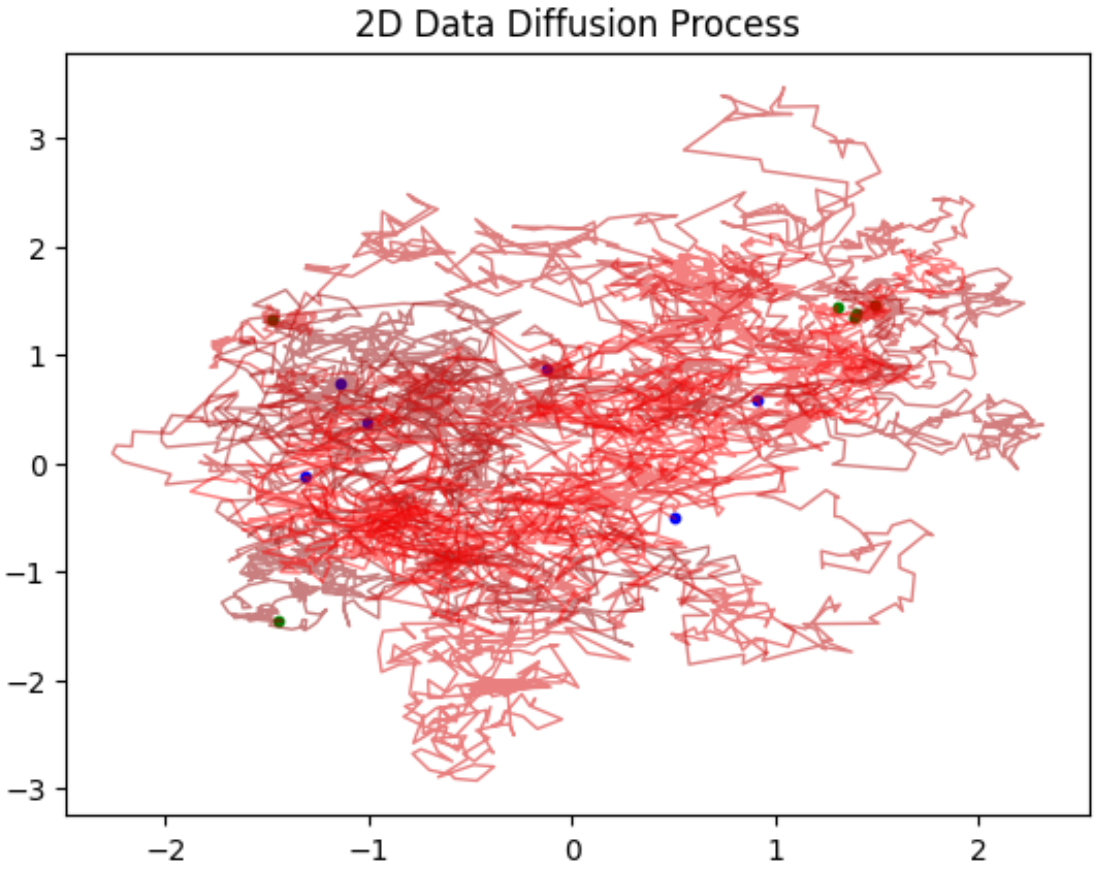



Figure 3: 2D Data Diffusion Process

1. Initialize a batch of 2D data points drawn from the structured distribution.
2. Iteratively add noise to each sample using the stepwise forward diffusion function.
3. Store the intermediate trajectories for visualization.
4. Plot the particle trajectories, showing the progressive diffusion of points through 2D space.

At the Figure.3, **the red lines** represent the movement of individual particles over time, showing how they diffuse throughout the 2D space. **The green dots** indicate the initial positions of the particles, while **the blue dots** represents their final positions after the diffusion process has fully evolved.

Unlike the initial structured state, the final positions are randomly scattered throughout the space, indicating that the data has successfully transitioned to an approximately **Gaussian distribution**.

# 3 Reverse Process (Denoising)

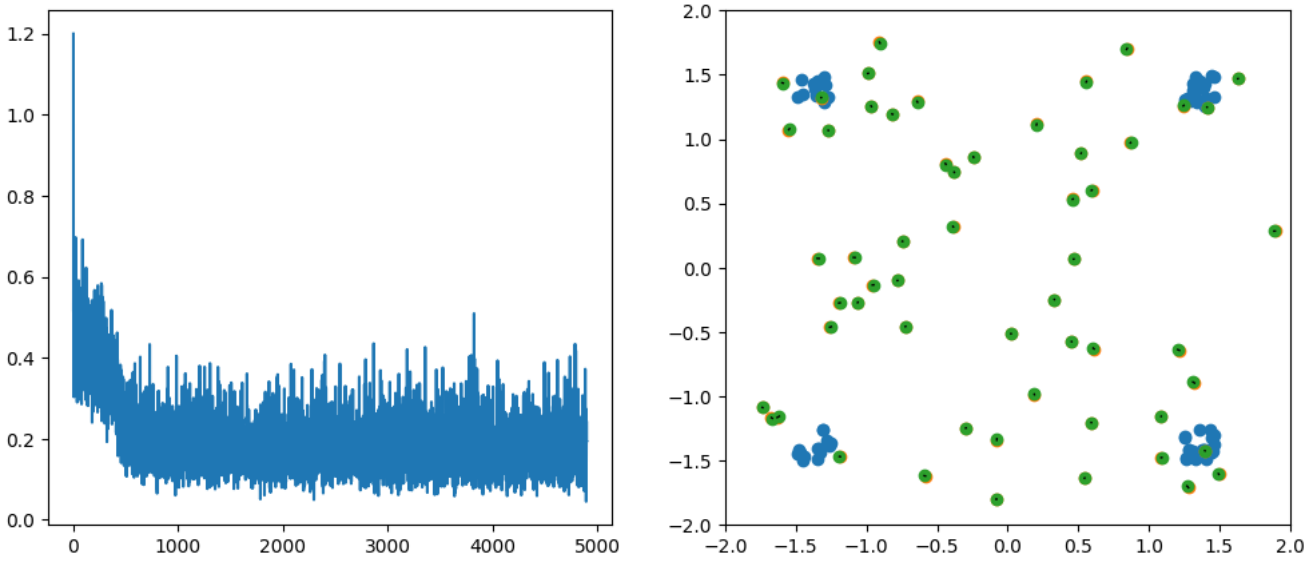



Figure 4: Reverse Process (Denoising)

Having implemented and visualized the forward diffusion process, we now aim to reverse the process by training a neural network to denoise the data step by step. This approach follows the Denoising Diffusion Probabilistic Model (DDPM) framework, which learns to gradually remove noise and reconstruct structured data.

The reverse process is formulated as an iterative sampling procedure:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left[ x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}t}} \epsilon_\theta(x_t, t) \right] + z\sigma_z \tag{3}$$

where:

- $\epsilon_\theta(x_t, t)$ is the learned noise prediction function.
- $z \sim \mathcal{N}(0, I)$ is additional noise, optionally introduced for stochastic sampling.
- $\sigma_z$ is the standard deviation of the noise at timestep $t$.

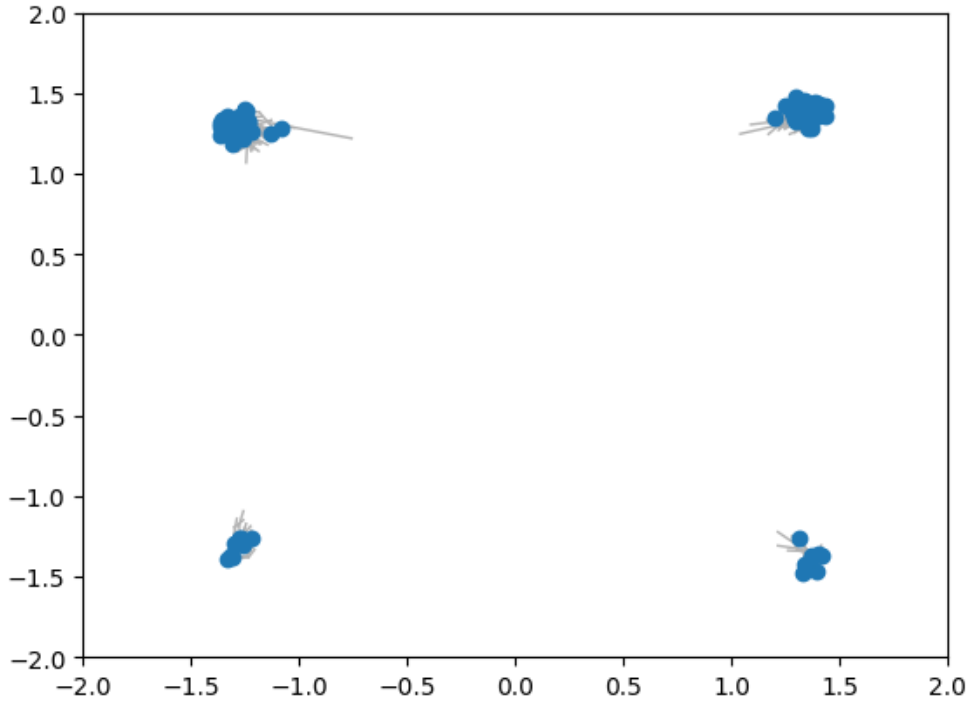
The training loop runs for 50,000 iterations, updating the model and periodically visualizing the loss and generated samples.(See Figure.4)

1. **Loss Curve**: The loss steadily decreases, indicating that the model is learning to predict noise effectively.

2. **Reconstructed Samples**: The model successfully denoised data, progressively recovering the structured 2D data distribution.

# 4 Inference Visualization

After training the diffusion model to reverse the forward process and reconstruct structured data from noise, we now evaluate its performance by visualizing the inference process. The goal is to observe how well the model can progressively denoise data, recovering the original structured distribution from random noise.




From the figure, we are able to see the denoising process starts with pure Gaussian noise, with points randomly scattered in the 2D space. As the process unfolds, data points gradually move towards their structured locations. The gray trajectories represent the stepwise evolution of each point over time. The final denoised samples accurately recover the original structured distribution.