Reverse Diffusion on 2D Spirals

Atharva Kulkarni apkulkarni@ucsd.edu
Ester Tsai etsai@ucsd.edu
Karina Chen kac009@ucsd.edu
Zelong Wang zew013@ucsd.edu

Alex Cloninger acloninger@ucsd.edu
Rayan Saab rsaab@ucsd.edu

Abstract

Generative AI technologies like Dall-E, Imagen, Stable Diffusion, and Midjourney have revolutionized image creation, offering a wide range of results from text prompts. Utilizing diffusion models, they surpass traditional models like GANs, VAEs, and Flow-based models in stability, robustness against overfitting, and diversity of results. Diffusion models operate through a forward and reverse process, gradually reversing diffusion to predict added Gaussian noise. Sohl-Dickstein et al. (2015) introduced a method approximating conditioned probability distributions in reverse diffusion, while Ho, Jain and Abbeel (2020) implemented the U-Net architecture to learn denoising functions. This paper explores diffusion models using a simple 2D dataset, investigates how points move during the diffusion process, and examines the effects of various hyperparameters on model success. Our research demonstrates the efficacy of diffusion models on a Swiss roll dataset, suggesting their broader applicability to more complex data like images and videos.

Code: https://github.com/ester-tsai/diffusion_model_spiral_data

1 Introduction 2
2 Methods 2
3 Results 4
4 Discussion 6
5 Conclusion 7
References 8

1 Introduction

Diffusion models are a relatively new invention but have proved immensely popular and effective. They satisfy creative needs, reduce manual labor, lower the threshold for skill level, and allow users to experiment with unique designs. The most popular uses of diffusion models include text-to-image generation, layout-to-image generation, inpainting masked images, and enhancing image resolution, which are all useful in real-world applications. Common commercial applications of diffusion models include OpenAI's Dall-E and Adobe's Firefly. Dall-E is purely a text-prompt-based image generation tool, whereas Firefly integrates with Adobe's existing products; for example, Firefly improves Photoshop's ability to generate a larger background.

Many image generation tools currently use variations of diffusion models, and several papers exist in the field, but mostly in the scope of the applications for generating images. Our project simplifies the data distribution to a 2D space (instead of, for example, 65,536 dimensions for a 256 by 256 image). Specifically, we are applying a diffusion model to the common Swiss roll dataset. This targets the problem of how a basic diffusion model works and aims to make the concepts clear mathematically and visually.

The Swiss roll problem was solved using a radial basis function network with a single hidden layer and 16 hidden units with 40 timesteps (Sohl-Dickstein et al. 2015). On the other hand, our current solution utilizes a neural network with linear layers and 50 timesteps, and it seems to work similarly based on the reversed output. In this paper, we will go into our experiments with other values for these parameters.

A new method of producing samples from probability density $p(x)$ uses only the gradients $\nabla x log p(x)$ in a Markov chain of updates (Welling and Teh 2011). The benefit of this physics-inspired approach is that it avoids collapsing into local minima by injecting Gaussian noise into parameter updates, compared to stochastic gradient descent.

The training objective can be simplified by ignoring weighted terms while training the neural network and is called DDPM (Ho, Jain and Abbeel 2020). However, DDPMs are very slow to generate samples in the reverse diffusion process. As a result, others have tried running a strided sampling schedule (Nichol and Dhariwal 2021), and DDIMs were proposed as well (Song, Meng and Ermon 2020), which derived an implicit generative model with the same marginal noise distributions as DDPMs while still deterministically mapping noise to images.

2 Methods

In our study, we reproduced the Denoising Diffusion Probabilistic Model (DDPM) framework to train and generate 2D data following the Swiss roll distribution. We experimented with different parameters within this framework, including the number of denoising steps and the level of randomness introduced at each step. We briefly review the formulation of DDPMs from Ho, Jain and Abbeel (2020). For a more general derivation, see Sohl-Dickstein et al. (2015).

Given a data distribution $x_0 \sim q(x_0)$, we define a forward noising process q which produces latents x_1 through x_T by adding Gaussian noise at time t with variance β_t as follows:

$$q(x_1, ..., x_T|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{1}$$

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}) \tag{2}$$

Given sufficiently large T and a well behaved schedule of β_t, the latent x_T is nearly an isotropic Gaussian distribution. Thus, if we know the exact reverse distribution $q(x_{t-1}|x_t)$, we can sample $x_T \sim \mathcal{N}(0, \mathbf{I})$ and run the process in reverse to get a sample from $q(x_0)$. However, since $q(x_t|x_{t-1})$ depends on the entire data distribution, we approximate it using a neural network as follows:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{3}$$

As noted in (Ho, Jain and Abbeel (2020)), the noising process defined in Equation 2 allows us to sample an arbitrary step of the noised latents directly conditioned on the input x_0. With $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$, we can write the marginal

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{4}$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{5}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Here, $1 - \bar{\alpha}_t$ tells us the variance of the noise for an arbitrary timestep, and we could equivalently use this to define the noise schedule instead of β_t.

Using Bayes theorem, one can calculate the posterior $q(x_{t-1}|x_t, x_0)$ in terms of β_t and $\mu_t(x_t, x_0)$ which are defined as follows:

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{6}$$

$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}\beta_t}}{1-\bar{\alpha}_t}x_0 + \sqrt{\frac{\bar{\alpha}_t(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}x_t \tag{7}$$

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t\mathbf{I}) \tag{8}$$

There are many different ways to parameterize $\mu_\theta(x_t, t)$ in the prior. The most obvious option is to predict $\mu_\theta(x_t, t)$ directly with a neural network. Alternatively, the network could predict x_0, and this output could be used in Equation 7 to produce $\mu_\theta(x_t, t)$. The network could also predict the noise ϵ and use Equations 5 and 7 to derive

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t, t)\right) \tag{9}$$

Ho, Jain and Abbeel (2020) found that predicting ϵ worked best with loss function:

$$L_{\text{simple}} = E_{t,x_0,\epsilon}\left[||\epsilon - \epsilon_\theta(x_t, t)||^2\right] \tag{10}$$

Our experiments were designed to observe the denoising process across various time steps, examining how individual data points and clusters evolve to reconstruct the final Swiss roll distribution (Sohl-Dickstein et al. 2015). This analysis allowed us to gain insights into the probabilistic modeling of data and the effectiveness of the diffusion process in capturing complex data distributions.

3 Results

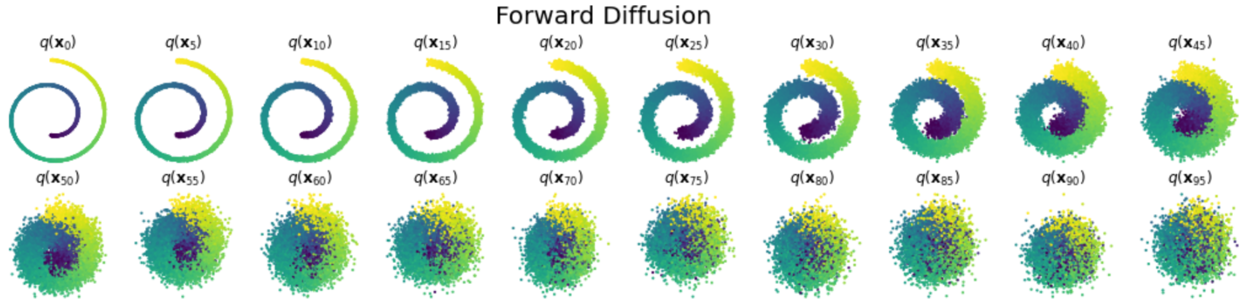

Figure 1: The forward diffusion process for swiss roll data, with n_steps=100

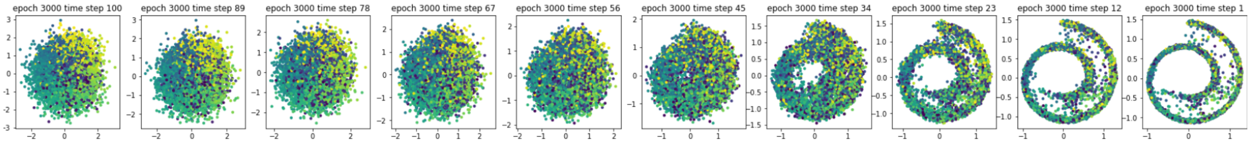

Figure 2: The reverse diffusion process for swiss roll data

Figures 1 and 2 show the forward and reverse diffusion processes, respectively. Although the Swiss roll started with a perfect gradient color along the manifold line, the reverse process was unable to reconstruct the exact color arrangement. Each individual point appears to end up at the nearest location on the Swiss roll distribution, as opposed to ending up at its exact original location. We can also see that there is noise where the points do not all go back to a location on the manifold. This is the same as the output obtained by Sohl-Dickstein et al. (2015), and is to be expected.

Figure 3 shows a clear separation of the two clusters by color. As the forward diffusion progresses, the points become increasingly mixed by adding random Gaussian noise.

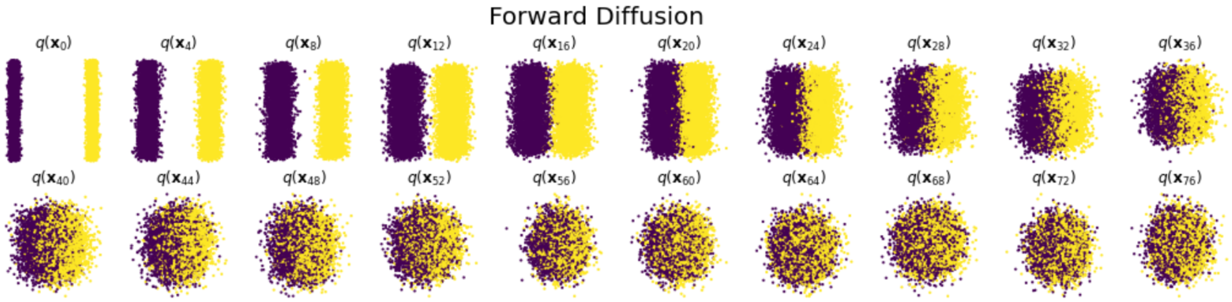

Figure 3: Forward diffusion process for distribution of two clusters

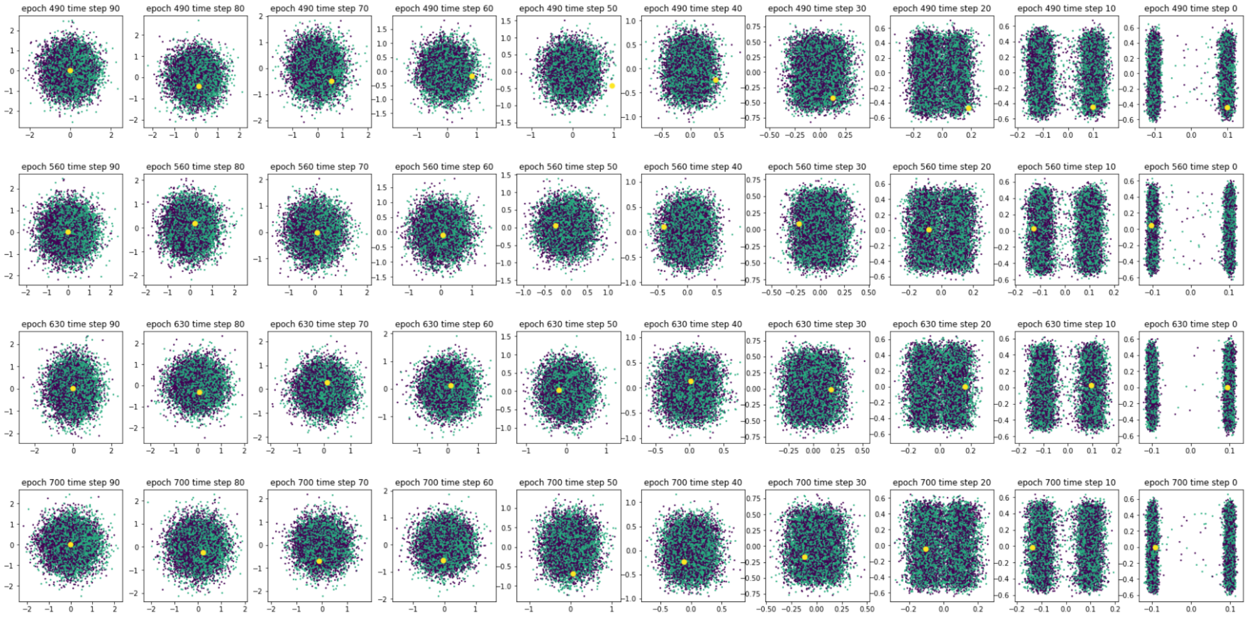

Figure 4: Movement of a single point in the reverse diffusion process with distribution of two clusters

Figure 4 tracks the movement of a single point as well as the two clusters in the reverse diffusion process. This point always starts at (0, 0), which is at the very center of the two clusters, and it can end up in either cluster. We can see that there is no set path that the point follows, and it ends up at either cluster with equal probability. Furthermore, the original two clusters do not separate cleanly. We can infer that the points go to their nearest cluster, rather than to the cluster they originated in before noise was added.

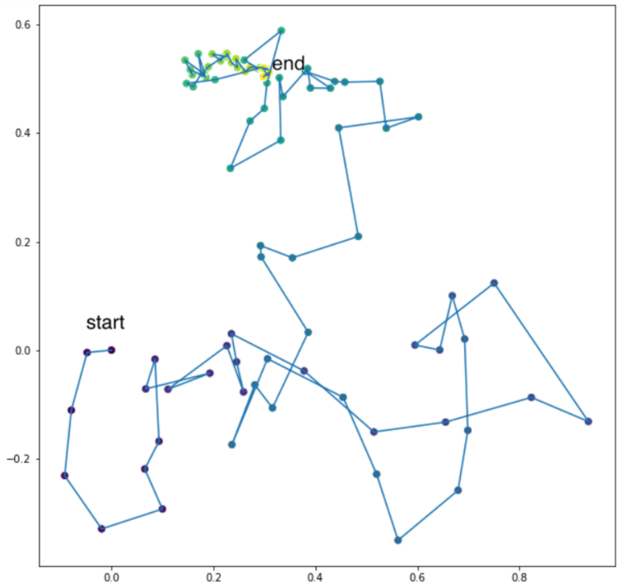

Figure 5: Movement of a single point in the reverse diffusion process with the two cluster distribution

Figure 5 tracks the movement of a single point in the reverse diffusion process for the distribution of two clusters. We can see that while the steps taken are very random and unpredictable towards the beginning, as the timestep increases, the movement is smaller and seemingly more concentrated.

4 Discussion

4.1 Inside the Diffusion Process

The results of our replication and extension of the Denoising Diffusion Probabilistic Model (DDPM) for 2D data provide a compelling visualization of the diffusion process. Figures 1 and 2 illustrate the forward and reverse diffusion processes for Swiss roll data with 100 steps, offering a clear representation of the model's behavior over time. The visual degradation of the Swiss roll into a noisy distribution and its subsequent reconstruction provide a intuitive understanding of the model's generating process.

The forward diffusion process effectively obfuscates the original Swiss roll structure, while the reverse diffusion process, although unable to perfectly reconstruct the original color position arrangement, shows a clear convergence towards the original data structure.

The initial unpredictability and the eventual stabilization of the trajectory mirror the inherent uncertainty in the diffusion process and follows our assumption since the noise added in early stage is smaller than noise added in later stage.

4.2 Applications and Future Work

While our experiments focus on 2D Swiss roll data, there is potential for DDPM to reconstruct complex patterns in fields such as image generation and noise reduction. As it has been difficult to probe into the details of high-dimensional data distribution, our work provides a easy way to gain comprehensive understandings of the diffusion process.

4.3 Limitations

While our model performed well in reconstructing the Swiss roll distribution, it is important to note the limitations. The model's performance on other complex distributions and in real-world scenarios remains to be thoroughly evaluated. Additionally, the computational intensity of the model, especially for high-dimensional data, poses a challenge for practical applications.

5 Conclusion

We have seen DDPMs applied on a 2D swiss roll distribution and on a two cluster distribution. During the forward diffusion process, we can see how as more random Gaussian noise is added, the distribution also tends towards a random Gaussian distribution of $N(0, I)$. For both distributions, we can conclude that points during the reverse diffusion process go back to the closest point on the manifold, with some randomness. From these experiments, we have gained a better understanding of how points move during both the forward and reverse diffusion processes in addition to a better grasp of how diffusion models are implemented as a whole.

References

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. 2020. “Denoising Diffusion Probabilistic Models.” *CoRR* abs/2006.11239. [Link]

Nichol, Alex, and Prafulla Dhariwal. 2021. “Improved Denoising Diffusion Probabilistic Models.” *CoRR* abs/2102.09672. [Link]

Sohl-Dickstein, Jascha, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics.” *CoRR* abs/1503.03585. [Link]

Song, Jiaming, Chenlin Meng, and Stefano Ermon. 2020. “Denoising Diffusion Implicit Models.” *CoRR* abs/2010.02502. [Link]

Welling, Max, and Yee Whye Teh. 2011. “Bayesian Learning via Stochastic Gradient Langevin Dynamics.” In *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Madison, WI, USA Omnipress