DISTRIBUTION MAPPING USING DIFFUSION PROCESSES : SDE'S AND ODE'S

Swati Swati

Victoria University of Wellington, NZ

1. INTRODUCTION

Diffusion models, based on Stochastic Differential Equations (SDEs) and Ordinary Differential Equations (ODEs), are powerful generative tools for transforming data distributions. These models are especially useful in tasks like image generation, where mapping one distribution to another is essential.

In this study, we apply both SDE and ODE methods to map between two-pixel images from different distributions: Gaussian data to a uniform distribution of "dogs" and a "cat" distribution to the same target. Using the first-order Euler method for the SDE and linear interpolation for the ODE, we compare their performance and visualize the transformations and velocity fields. This work explores the effectiveness of these approaches for distribution mapping in generative modeling [3].

2. THEORY

In this section, we focus on the theoretical background relevant to the implemented system.

2.1. Diffusion Models

Diffusion models use stochastic processes to transform data distributions over time. The basic idea is to gradually add noise to data, pushing it towards a known simple distribution (e.g., Gaussian), and then learn the reverse process to generate data from noise. This reverse process is what allows diffusion models to map distributions from one form to another, making them powerful tools for generative modeling [1].

2.2. Stochastic Differential Equations (SDEs)

An SDE describes the evolution of a system with both deterministic and random components. The score-based SDE approach seeks to learn the reverse diffusion process by estimating the score function. The general form of the SDE for diffusion models can be written as:

$$d\mathbf{x} = f(\mathbf{x}, t)dt + g(t)d\mathbf{w}, \tag{1}$$

where $f(\mathbf{x}, t)$ is the drift term, $g(t)$ is the diffusion term, and $d\mathbf{w}$ represents the Wiener process (Brownian motion). When $f(\mathbf{x}, t) = 0$, the equation simplifies to a purely stochastic process, ideal for fast transformations in generative tasks.

2.3. Ordinary Differential Equations (ODEs)

Unlike SDEs, ODEs involve only deterministic components. The evolution of the system is described solely by a function $f(\mathbf{x}, t)$, which governs how the data transforms over time [3]. The ODE for generative models is given by:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, t), \tag{2}$$

where $f(\mathbf{x}, t)$ is learned to map the input data (e.g., Gaussians) into the target distribution (e.g., dogs). The linear interpolation is used in ODE, to smoothly map from one distribution to another. For two distributions, P_0 (e.g., Gaussians) and P_1 (e.g., dogs), the interpolant can be expressed as:

$$P_t = (1 - t)P_0 + tP_1, \tag{3}$$

where $t \in [0, 1]$ controls the interpolation between the distributions over time.

2.4. Euler and Euler-Maruyama Methods

The Euler method provides a simple numerical approximation for solving ODEs. For a differential equation $\frac{dx}{dt} = f(x, t)$, the Euler update rule is:

$$x_{n+1} = x_n + f(x_n, t_n)\Delta t, \tag{4}$$

where Δt is the step size. For SDEs, the Euler-Maruyama method adds a stochastic component $g(t_n)\Delta w_n$ to the above equation, where Δw_n represents the Wiener process increment. This method is crucial for approximating the stochastic paths of the SDEs used in diffusion models [3].

2.5. Velocity Fields

Velocity fields visualize the rate of change of data points as they are mapped from one distribution to another. For an ODE, the velocity field is given by $f(x, t)$, describing the direction and speed of each point's transformation. In an SDE, the velocity field incorporates stochasticity, combining the drift $f(x, t)$ and diffusion $g(t)$ [3].

2.6. Kullback-Leibler Divergence

Kullback-Leibler (KL) divergence is a measure of how one probability distribution diverges from a second, expected probability distribution. It quantifies the information lost when approximating one distribution with another.

2.7. Wasserstein Distance

The Wasserstein distance, measures the distance between two probability distributions and compares the overall structure of distributions.

2.8. Histogram Intersection

Histogram Intersection is a measure of similarity between two distributions based on their histograms and also evaluates the overlap between two pixel distributions. It calculates the sum of the minimum values for each bin in the histogram [5].

3. EXPERIMENTS

This section evaluates Stochastic Differential Equations (SDEs) and Ordinary Differential Equations (ODEs) for mapping distributions from Gaussians to Dogs and Cats to Dogs using two-pixel images. Dogs are uniformly distributed within the square defined by corners (2,2) and (3,3), cats between (2,-3) and (3,-2), while Gaussians have an identity covariance matrix scaled by s. Neural networks with inputs $x \in R^2$ and time t are used to predict the mappings [2].

3.1. Experimental Setup

The task involved mapping Gaussians to Dogs using both SDE and ODE approaches, followed by training the ODE to map Cats to Dogs. For both methods, we implemented a neural network to model velocity fields. The SDE model utilized a three-layer fully connected architecture with an input dimension of 3 (for $x \in R^2$ and time t), with 128 neurons in the first hidden layer, 64 in the second, and ReLU activations, producing 2 output neurons for the dog distribution. The ODE model featured an additional hidden layer with 256 neurons and employed the SiLU activation function throughout, allowing for direct output of predicted velocities.

3.2. Training Process

Both the SDE and ODE models were trained using the Adam optimizer, with the SDE model trained over 2500 epochs and the ODE model for 5000 epochs, both using a batch size of 64. The SDE model employed the Euler-Maruyama method to simulate sample paths, incorporating the score function to capture the drift term in the stochastic differential equation [4]. In contrast, the ODE model applied linear interpolation between the source and target distributions to define its trajectory. The plots comparing the original distribution with the Gaussian to Dog mapping for both SDE and ODE models, along with the Cat to Dog ODE mapping, are presented in Fig. 1, Fig. 2, and Fig. 3, respectively.

The loss function used was the Mean Squared Error (MSE) between the final state of the trajectory and the target distribution. The loss curves are plotted in Fig. 4. For both mappings (Gaussian to dog and cat to dog), the predicted velocities were optimized over the training iterations, and can be seen in the Fig. 5, Fig. 6 and Fig. 7.

To evaluate the models, several performance metrics were used. Mean Squared Error was used to measure reconstruction quality of the final mapped distribution against the target distribution. Wasserstein Distance was used to quantify the distributional difference between the mapped and target distributions. Histogram Intersection and KL Divergence were used to compare the overlap and shape similarity between the mapped and target distributions.

3.3. Results and Analysis

The evaluation results for the SDE and ODE approaches, visualized in Fig. 8, are summarized in Table 1. The ODE approach consistently outperformed the SDE across key metrics, particularly in Wasserstein Distance and Histogram Intersection. Notably, the ODE mapping from Cats to Dogs, detailed in Table 2, demonstrated the best overall performance.

Processing Time: The ODE approach demonstrated significantly lower processing time than the SDE approach, making it more efficient for distribution mapping.

Wasserstein Distance: The ODE's Wasserstein Distance of 0.0917 was notably lower than the SDE's 0.2670, indicating less work for transformation and better preservation of the target distribution's structure during mapping.

Histogram Intersection: The ODE achieved a Histogram Intersection of 0.4897, surpassing the SDE's 0.3106, further demonstrating its effectiveness in capturing target distribution characteristics, as shown in Fig 9

MSE Comparison: Despite the SDE's lower MSE of 0.1036 compared to 0.1663 for the ODE, this does not accurately reflect performance, as lower MSE in SDE may stem from a lack of capturing broader distributional properties. The difference in metrics is better visualized in Fig 10

KL Divergence: The ODE generally exhibited lower KL Divergence values, especially in the latter part of the output, indicating better alignment with the target distribution, as per Fig 11

Furthermore, the superior performance of the ODE in mapping Cats to Dogs is due to the structural similarity between the species, which enabled effective learning. The model's architecture and direct velocity field representation captured complex relationships, while the well-defined cat distribution facilitated smoother mappings and better alignment with the target.

4. CONCLUSION

In summary, while the SDE approach demonstrated competitive reconstruction quality, the ODE approach showed superior performance in capturing the characteristics of the target distribution through lower Wasserstein Distance and better Histogram Intersection values. Notably, the ODE model for mapping Cats to Dogs had the best performance across all metrics. Additionally, the significantly reduced processing time for the ODE approach enhances its practical applicability, making it a more efficient and effective choice for the task of mapping distributions in this experimental setup.

5. STATEMENT OF TOOLS USED

The tools used in this study include PyTorch for ODE and SDE implementation, Plotly for visualization, and Google Colab for executing the code. The code was written by me, and I verify that it is my original work. The code can be accessed at the following link: *Final-Code*.

6. REFERENCES

[1] B. D. Anderson. Reverse-time diffusion equation models. Stochastic Processes and their Applications, 1982.

[2] N. Maheswaranathan J. Sohl-Dickstein, E. Weiss and S. Ganguli. “deep unsupervised learning using nonequilibrium thermodynamic. International Conference on Machine Learning, 2015.

[3] Prof Bastiaan Kleijn. Diffusion: Learning process. Victoria University Lecture in L11_diffusion425.pdf, 2024.

[4] T. Aila T. Karras, M. Aittala and S. Laine. Elucidating the design space of diffusion-based generative models. arXiv preprint arXiv, 2022.

[5] P. Vincent. A connection between score matching and denoising autoencoders. Neural computation, vol. 23, 2011.

7. APPENDIX

Fig. 1: Distribution Comparison: Original vs Gaussian SDE.

Table 1: Performance Metric for SDE vs ODE Gaussian to Dog.

Metric	SDE	ODE
Processing Time	4 min	3 min
Wasserstein Distance	0.27	0.11
Histogram Intersection	0.325	0.4638
KL Divergence	0.55 to 1.12	0.17 to 0.36
MSE	0.103	0.171

Table 2: Performance Metric for ODE: Cat to Dog.

Metric	ODE
Processing Time	3 min
Wasserstein Distance	0.061
Histogram Intersection	0.673
KL Divergence	0.009 to 0.15
MSE	0.163

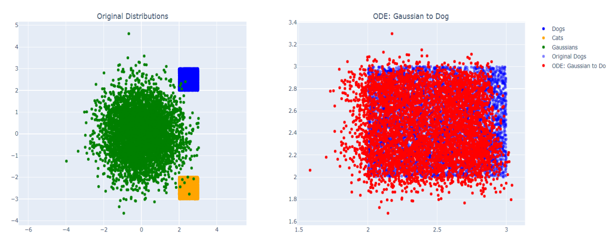

Fig. 2: Distribution Comparison: Original vs Gaussian ODE.

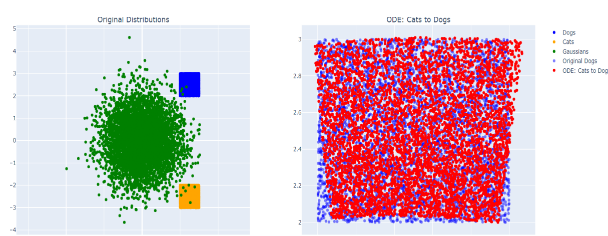

Fig. 3: Distribution Comparison: Original vs Cat ODE.

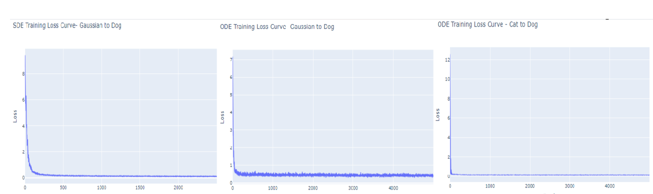

Fig. 4: Loss Curves: Gaussian SDE, Gaussian ODE, Cats ODE.

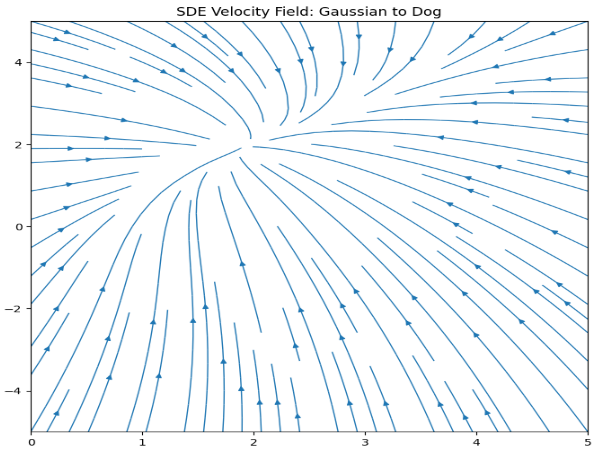

Fig. 5: SDE Velocity Field: Gaussian to Dog.

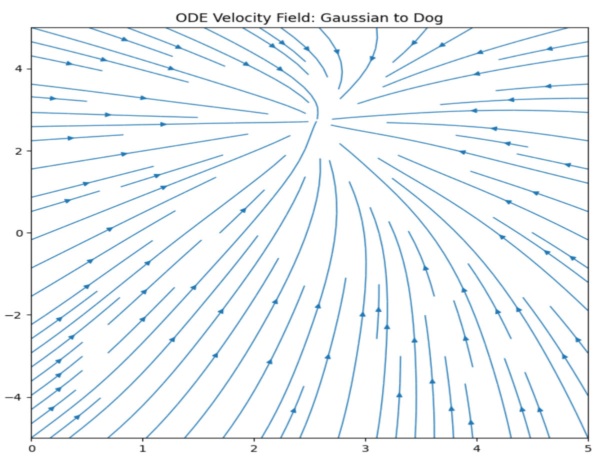

Fig. 6: ODE Velocity Field: Gaussian to Dog.

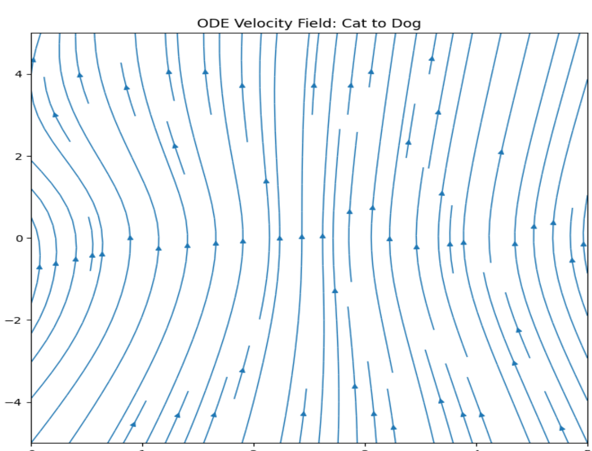

Fig. 7: ODE Velocity Field: Cat to Dog.

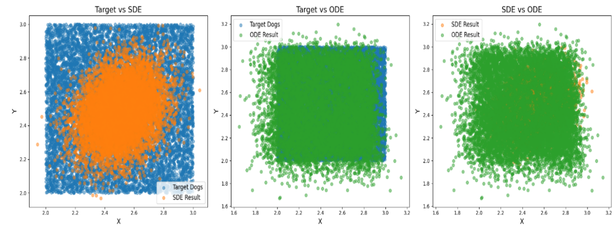

Fig. 8: Comparison: Target vs SDE, ODE and SDE vs ODE.

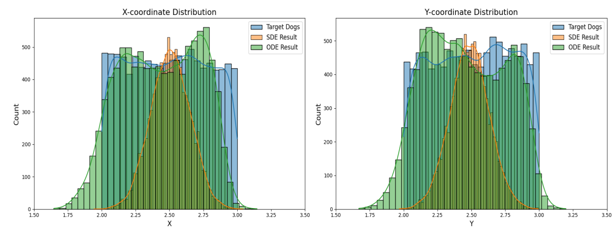

Fig. 9: Histogram Comparison: Gaussian SDE vs ODE.

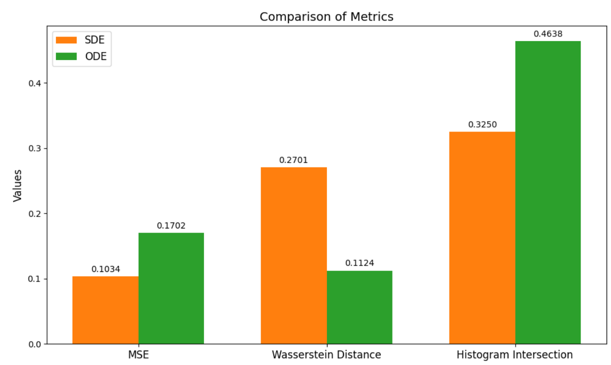

Fig. 10: Metric Comparison: Gaussian SDE vs ODE.

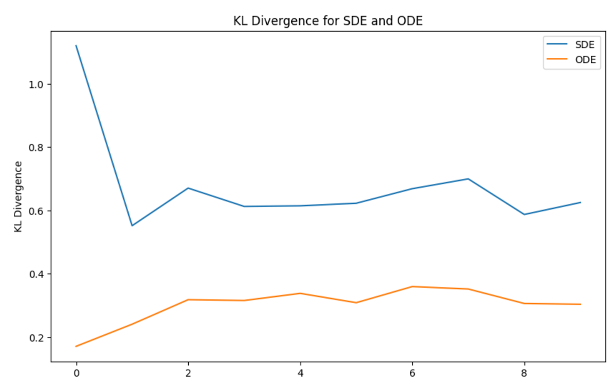

Fig. 11: KL Divergence for SDE vs ODE.