Exercise 2 - Flow Models

Ido Pinto
67912 - Advanced Course in Machine Learning

August 14, 2024

1 Normalizing Flows

1.1 Q1: Loss.

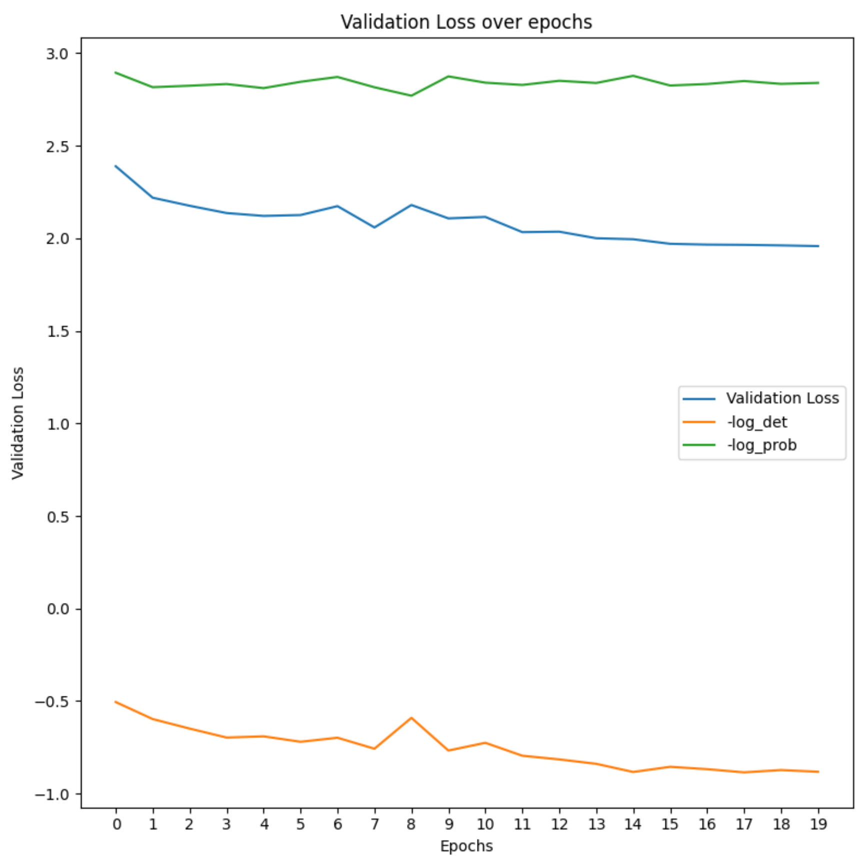


Figure 1: Validation Loss over the training epochs w/ log-determinant and $\log(P_z(x))$

1.2 Q2: Sampling.

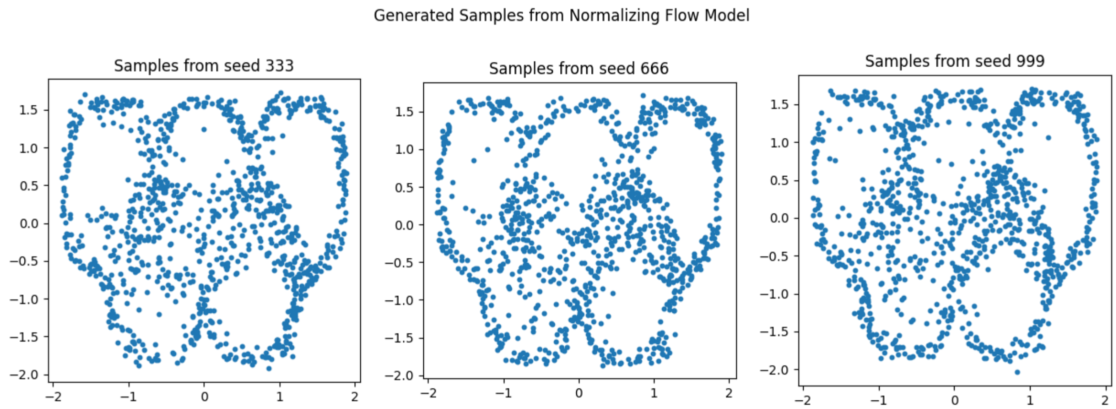


Figure 2: 1000 Generated samples from different seeds

1.3 Q3: Sampling over time.

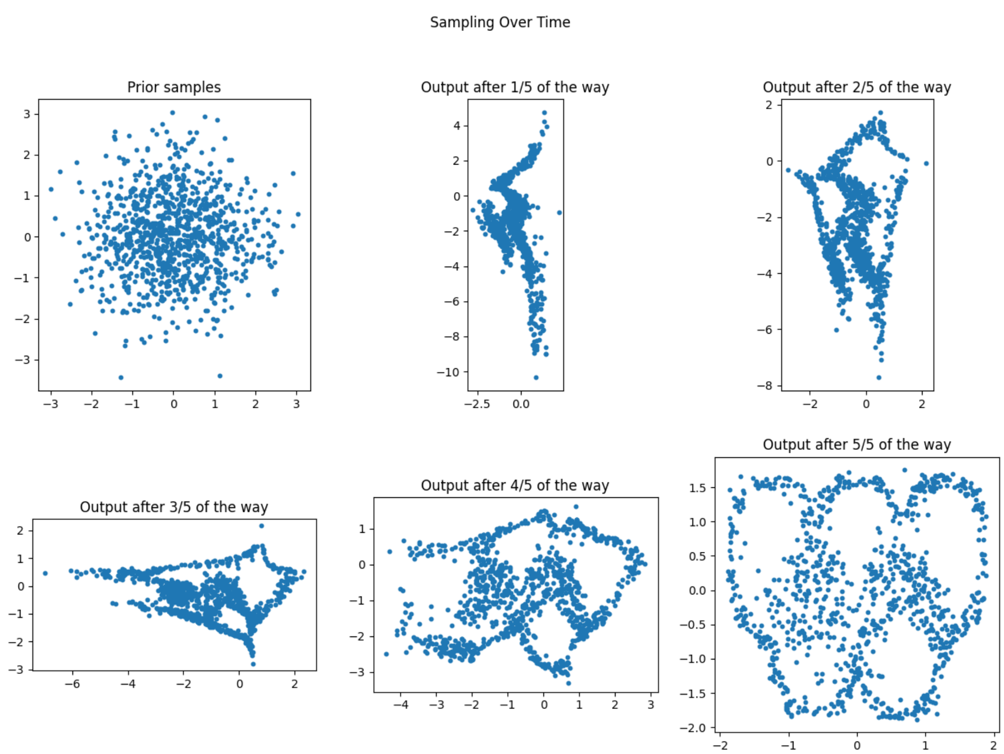


Figure 3: The movement of 1000 noise samples in 2D space during the model's forward pass.

1.4 Q4: Sampling trajectories

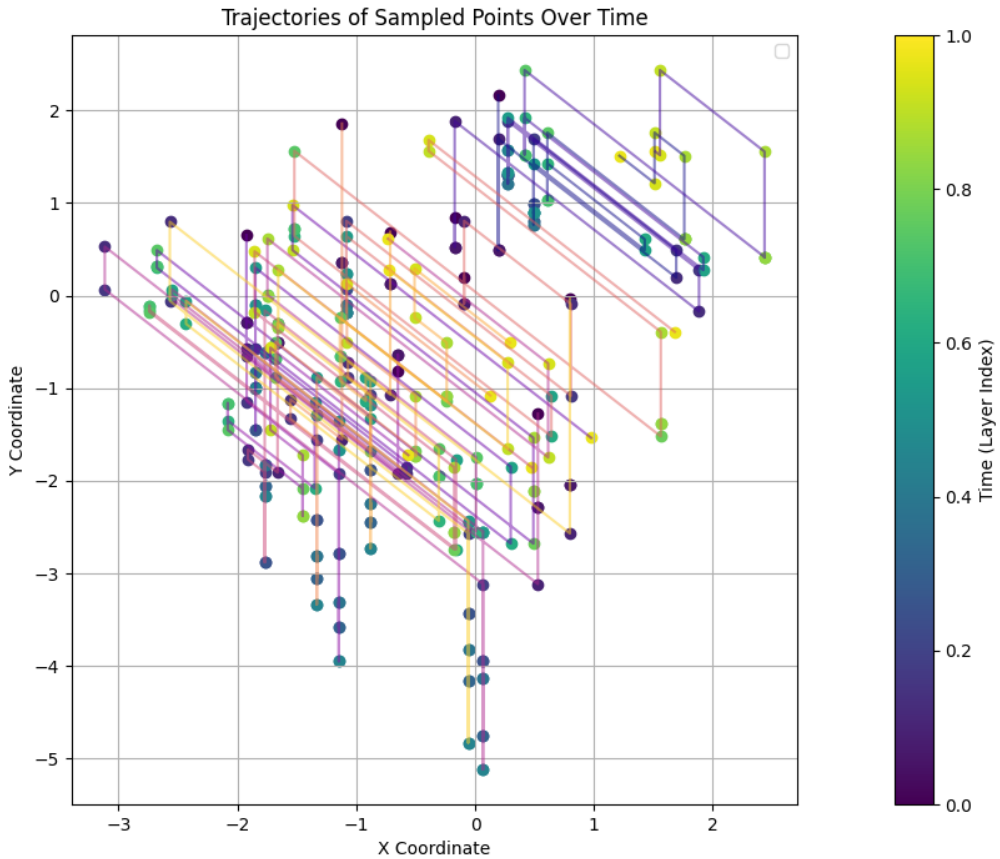


Figure 4: Trajectories of 10 noise samples in 2D space during the model's forward pass. Purple points are the initial noise samples and the yellow points are points that supposed to be on the Olympic Rings.

1.5 Q5: Probability estimation

Figure 5 illustrates the trajectory of 5 data-points through the inverse process, layer by layer. Three of these points are from within the Olympic rings and the other two are from outside of the rings. The points are colored according to their time t. We can observe that the inliers inverse trajectory ends at points that are close to (0,0), the mean of the prior distribution, whereas outliers trajectory ends at points that are further from that mean. Let's try to explain this behavior using the prior and posterior log probabilities of each points. The inliers are:

$$y_1 = (-0.68536588, 1.04613213)$$
$$y_2 = (-0.83722729, 1.48307046)$$
$$y_3 = (0.71348099, -0.03766172)$$

and the outliers are:

$$y_4 = (-1.44, -1.44)$$
$$y_5 = (1.8, 1.8)$$

The log-probabilities of the inliers according to the model:

$$\log(P_y(y_1)) = -1.7171$$
$$\log(P_y(y_2)) = -1.81$$
$$\log(P_y(y_3)) = -1.78$$

and the log-probabilities of the outliers according to the model:

$$\log(P_y(y_4)) = -19.98$$
$$\log(P_y(y_5)) = -13.4$$

we can observe that these outliers values are much lower than the inliers values. These values are consistent with the corresponding prior log-probabilities of the points.

$$\log(P_z(\phi^-1(y_1))) = -2.43$$
$$\log(P_z(\phi^-1(y_2))) = -3.033$$
$$\log(P_z(\phi^-1(y_3))) = -2.035$$
$$\log(P_z(\phi^-1(y_4))) = -21.33$$
$$\log(P_z(\phi^-1(y_5))) = -15.35$$

Since the log-probabilities and the prior log probabilites of the inliers are much closer to 0 than the outliers, we can understand that the points inside the Olympic logo are more likely from outside of it. The model is trained using MLE (maximum likelihood estimation) to map real data-points to points from the prior distribution $\mathcal{N}(0, I)$ This training strategy enables the model to link high-probability regions in the latent space with significant regions in the data space, enhancing both probability estimation and mapping accuracy.

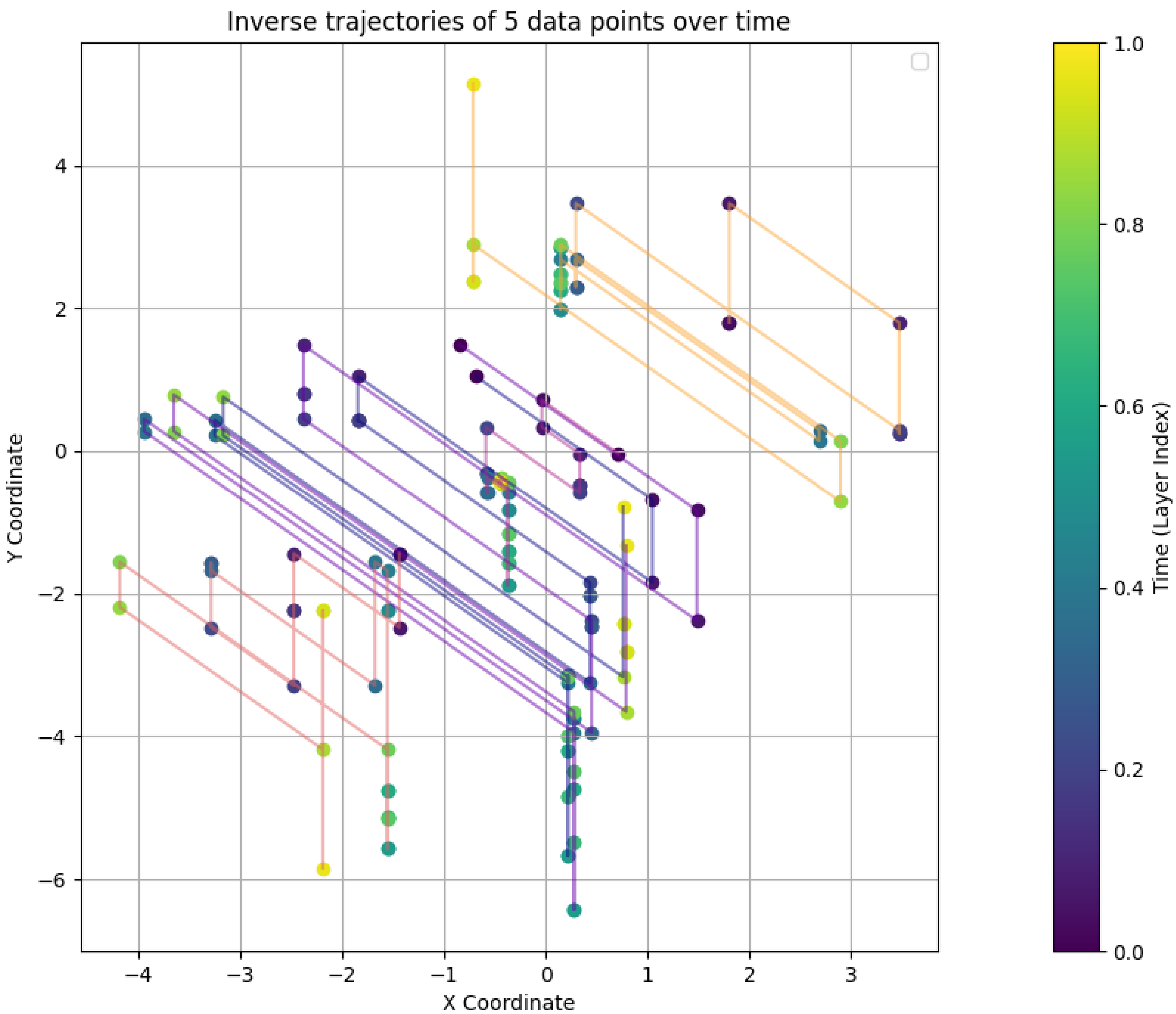


Figure 5: Inverse trajectories of 3 points on the Olympic rings and 2 points from outside of the rings.

2 Flow Matching

2.1 Unconditional Flow Matching

2.1.1 Q1: Loss.

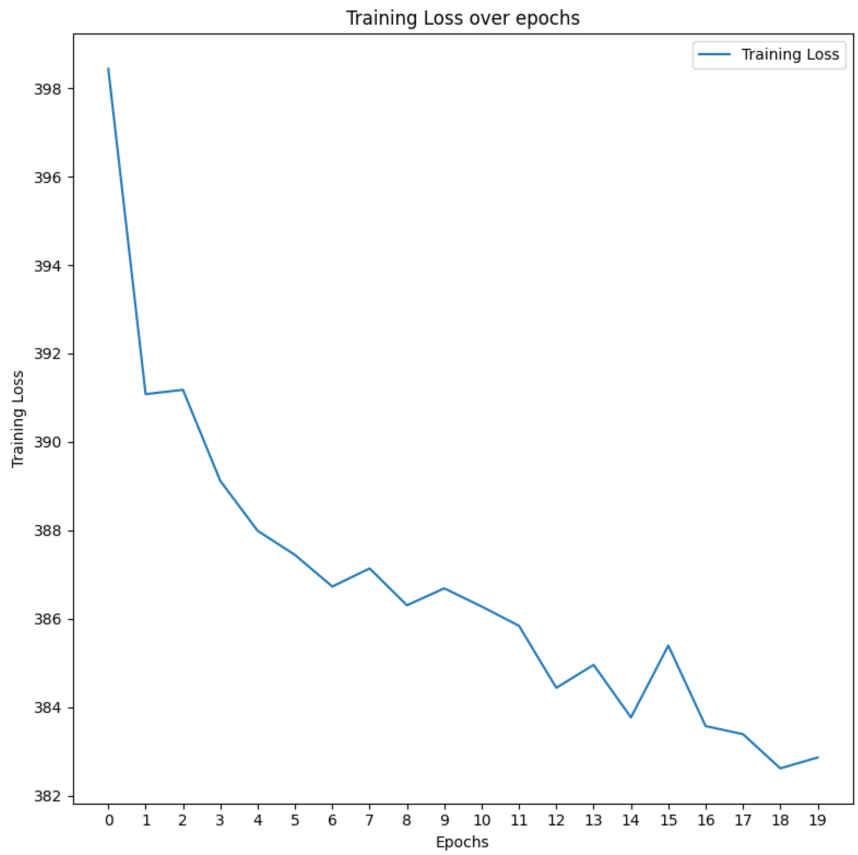


Figure 6: Training loss over epochs

2.2 Q2: Flow Progression.

In Figure 7 we can see the flow of $\{z_i\}_{i=1}^{1000}$ sampled from $\mathcal{N}(0, I)$ over time using the forward pass of our flow matching model for each plot until time $t \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$.
The distribution progression of the flow matching model seems to be smooth and less noisy, from inside-out compared to the distribution progression in normalizing flow presented in 3.

As we know, the flow matching model wishes to learn a continuous and smooth vector field resulting in a linear flow between the standard normal distribution and the distribution of the Olympic rings data. In contrast, Normalizing flow tries to find an invertible function between the data space and the latent space, using interleaving affine coupling layers and permutation layers, which breaks linearity and resulting in a messy distribution progression that doesn't necessarily follow any linear flow as seen in Figure 7. Overall, we can conclude from both figures that flow matching is more consistent and stable over time compared to normalizing flow.

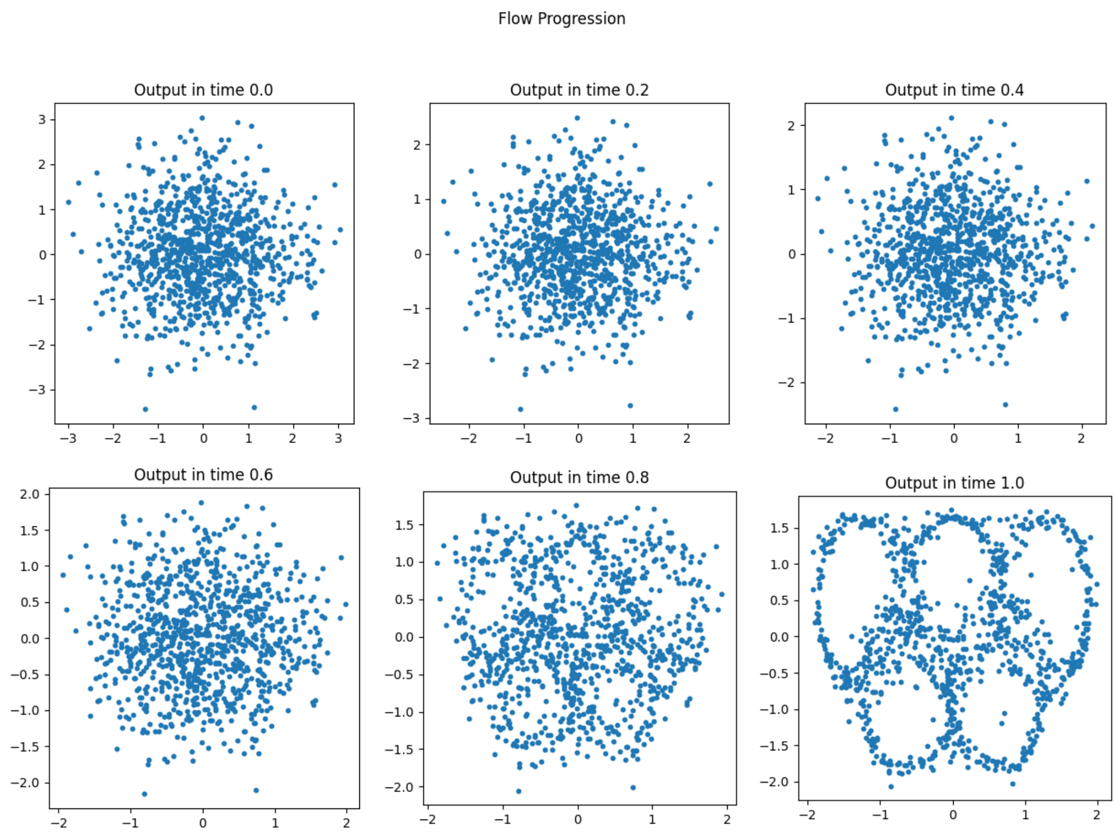


Figure 7: Unconditional flow Matching generated outputs over time

2.3 Q3: Point Trajectory.

Further to what was said in section 2.2, we can get a strong confirmation to our conclusion. In Figure 9 we can observe the flow smoothness of 10 points from the latent space over time. In contrast, normalizing flow samples trajectories presented in Figure 4 are messy, noisy and less smooth although the points end in the data space.

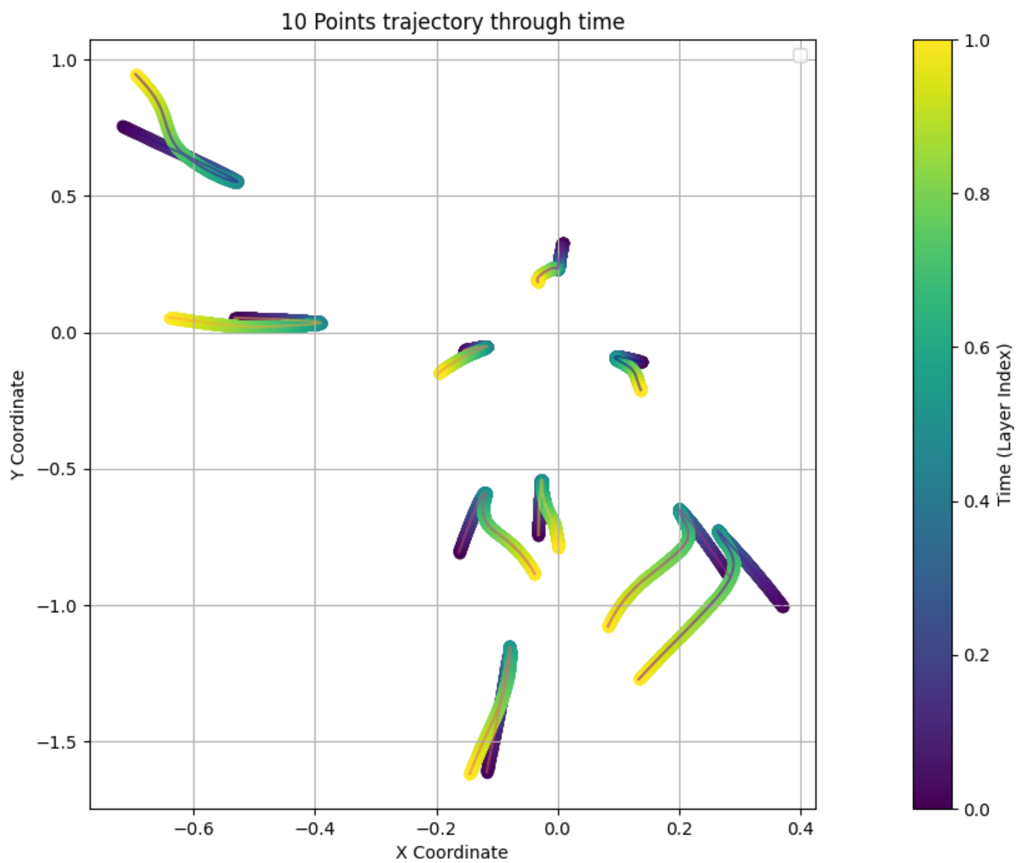


Figure 8: Unconditional Flow Matching 10 generated samples trajectories over time.

2.4 Q4: Time Quantization

In Figure 9, we can observe that the quantization of the flow indeed affect the resulted distribution. As Δt gradually increased the resulted distribution is more noisy and centered around the prior zero mean while loosing the shape of the rings and as Δt decreased on the other hand we get much more precise capture of the true distribution we desire.

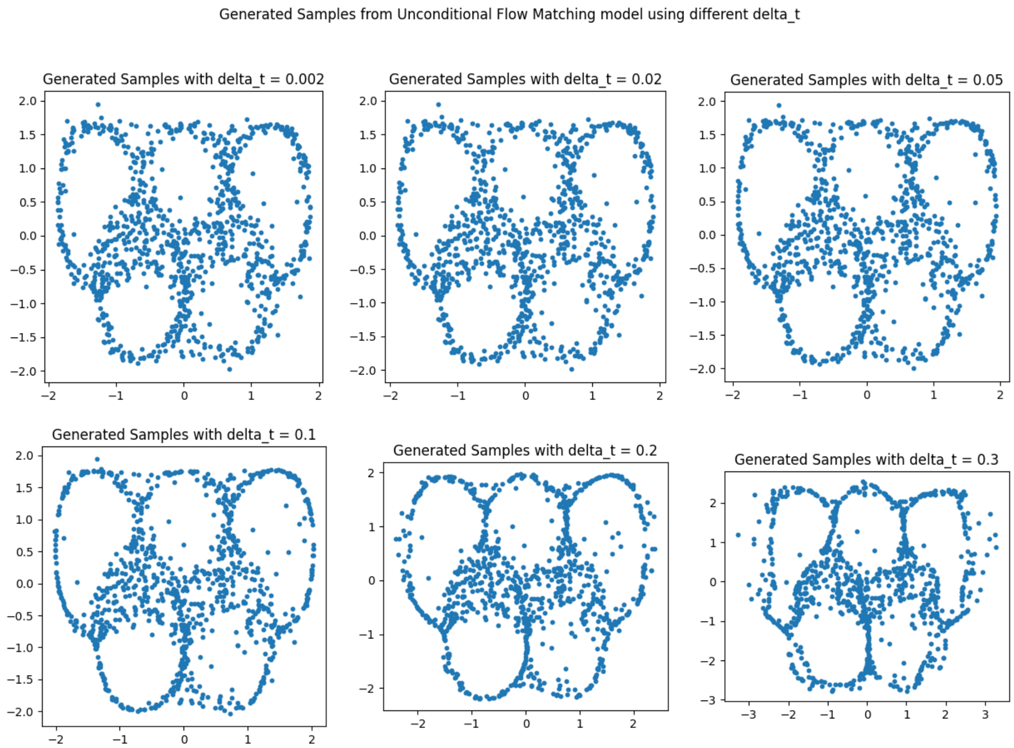
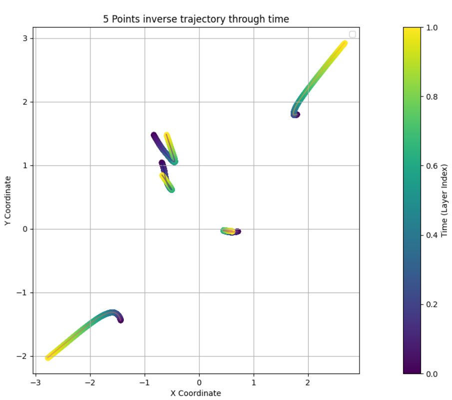


Figure 9: Unconditional Flow Matching generated points over time with different Δt

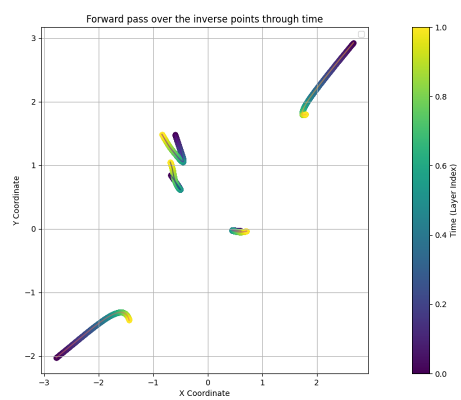
2.5 Q5: Reversing The Flow

In Figure 10(a), we can see the inverse flow trajectory of the same 5 points from section 1.5. The outputs from normalizing flow presented in 5 and the outputs from flow matching are clearly not the same. Both models moved the true inliers closer to the prior zero mean while moving outliers further from the prior zero mean. However the models differ in the mapping function they learn. In normalizing flow, it learns a composition of discrete mappings from data space to latent space and flow matching learns which direction and velocity to continue in time t and does it every Δt. As there are an infinite number of valid mapping functions that result in different outputs, it is very unlikely that both models will have the same outputs.

After re-entering the inverted points back into the forward model, we can observe in Figure 10(b) that we get the nearly the same points back, but not exactly the same. (up to precision errors, which can be solved by decreasing Δt) However if we re-enter the normalizing flow inversion back to the normalizing flow we will indeed get back the same data points, this is because this model is defined by an invertible function so clearly if the model is well-defined we should get the same points back.



(a) inverted trajectory of 5 points through time



(b) forward pass over the inverted points

Figure 10: Comparison between the inverse trajectory and the forward pass on it

2.6 Conditional Flow Matching

2.6.1 Q1: Plotting the Input.

I trained the conditional flow matching model by inheriting the unconditional flow matching class, and concatenating the embedding of each class with the input before the original flow matching model forward pass. In Figure 11 we can see the training dataset.

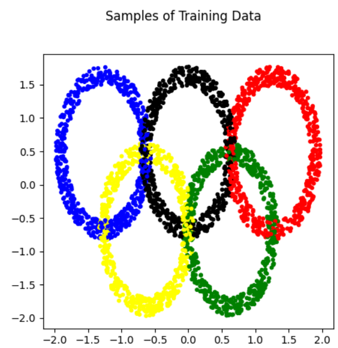


Figure 11: Training data points colored by their class

2.7 Q2: A Point from each Class.

Figure 12 represents the trajectory of points (one from each class, colored by their class's color) through time. Consider the black point with purple scope. If we followed it's trajectory, we will end at the black ring class as we expected. The same can be done for each class.

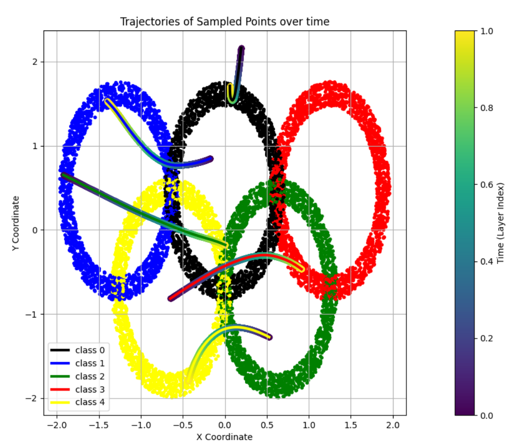


Figure 12: Trajectory of one point from each class

2.8 Q3: Sampling.

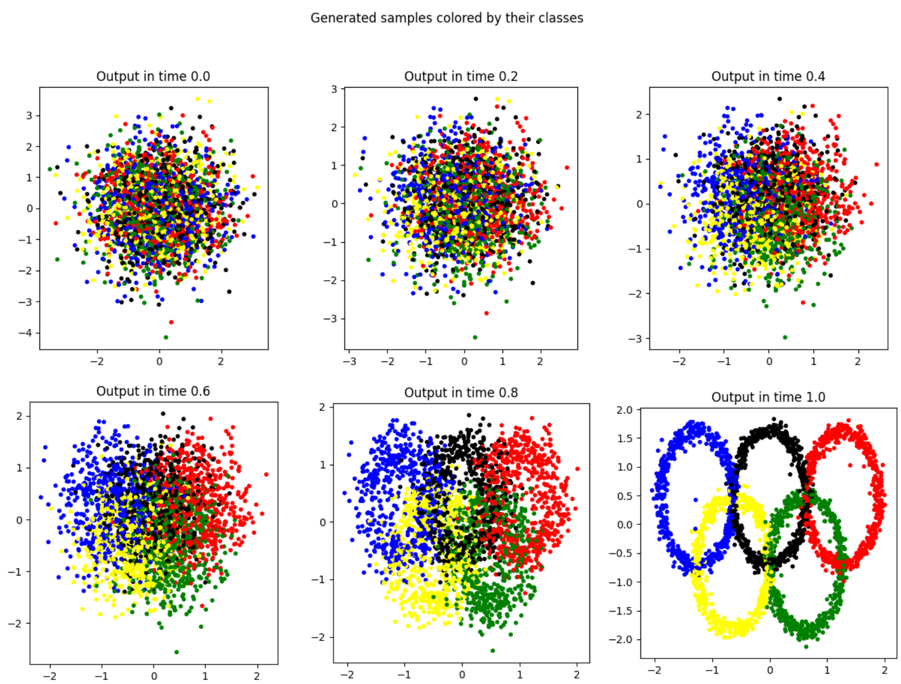


Figure 13: Generated points from our Conditional Flow Matching model colored by their classes's color.