AMATH 563: GENERATING A TRANSPORT MODEL FOR COMPLEX TARGET DISTRIBUTIONS IN 2D

WINNIE LAU

Department of Applied Mathematics, University of Washington, Seattle, WA
wwlau@uw.edu

1. Introduction

The goal of this report is to design and implement a minimum MMD transport map to transport $\eta = N(0, I)$ onto some target distribution ν, which is defined as the Moons, Swissroll, and Pinwheel dataset. The empirical MMD distance between η and ν for each target was calculated given the RBF, Laplacian, and polynomial kernel of degree 2, where we found the RBF kernel had the lowest consistent MMD. A transport model was then developed with the transport map trained on the RBF kernel, Laplace kernel, and polynomial kernels of degree 1, 2, 4. The RBF and Laplace kernel had the best visual quality, while all the polynomial kernels did not converge to the target distributions. Additionally, decreasing N results in poorer visual quality of the samples for all kernels.

2. Methods

The target distributions used in this report are benchmarks taken from the paper by Grathwohl et al., "FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models".

Let the reference distribution be defined by $\eta = N(0, I)$, which is the standard normal distribution in $\mathbb{R}^2$. Let the target distribution be defined as ν, containing samples from three target distributions: Swiss Roll, Moons, or Pinwheel (Figure 1). The number of empirical samples within each distribution is defined by $N \in \mathbb{N}$, and each dataset is normalized.

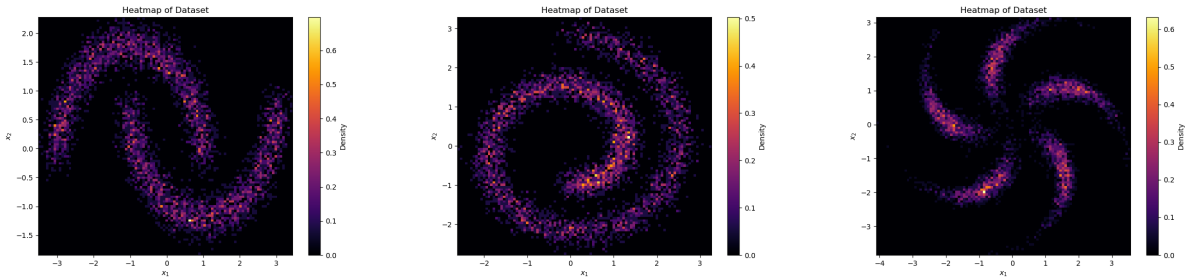

Figure 1. Heatmap of samples from target distributions Moons, Swissroll, and Pinwheel from left to right.

Date: June 10, 2025.

2.1. **Part 1.** Consider two sets of samples X and X' where each $x_i \in X$ and $x'_j \in X'$ are identically and independently sampled from η and ν, respectively, for $i, j = 1, \dots N$. The maximum mean discrepancy (MMD) between the two samples is defined as

$$MMD(X, X') = \frac{1}{N^2}\sum_{i,j=1}^{N} G(x_i, x_j) + \frac{1}{N^2}\sum_{i,j=1}^{N} G(x'_i, x'_j) - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G(x_i, x'_j).$$

The kernel G for MMD is defined as

- RBF Kernel: $G(x, x') = \exp(-\frac{1}{2\sigma^2}\|x - x'\|)$
- Laplacian Kernel: $G(x, x') = \exp(-\frac{1}{\sigma}\|x - x'\|)$
- Polynomial Kernel: $G(x, x') = (x^T x' + c)^d$ for $c \in \mathbb{R}$ and $d \in \mathbb{N}$.

The lengthscale σ for the RBF and Laplacian Kernel is calculated as

$$d_{ij} = \text{dist}(x_i, x'_j) = \|x_i - x'_j\|$$
$$\sigma = \beta \cdot \text{median}(d_{ij})$$

using the median heuristic, given some $\beta \in \mathbb{R}$. To start, we use $\beta = 1$, $c = 1$, and $d = 2$ for the polynomial kernel.

2.2. **Part 2.** Let $\mathcal{T}$ be the RKHS of a matrix valued kernel where we use the kernel K, originally defined as the RBF kernel from Part 1. Let Z and X represent our reference and target samples, respectively. Our optimization function is defined as

$$T^* = \underset{T \in \mathcal{T}}{\arg\min} \quad MMD_G^2(T\#\eta^N, \nu^N) + \frac{\lambda}{2}\|T\|_{\mathcal{T}}^2$$

which has some solution

$$T^* = (T_1^*, \dots, T_n^*), \quad T_j^* = K_j(\cdot, Z)(K_j(Z, Z) + \sigma_j^2 I)^{-1} w_j^*$$
$$w_j^* = \underset{V}{\arg\min} \quad MMD_G^2(Z + V, X) + \frac{\lambda}{2}\sum_{j=1}^{N} w_j^T (K_j(Z, Z) + \sigma_j^2 I)^{-1} w_j$$

where W has rows w_j, and V denotes its columns. The regularization parameter λ is tuned with the length scale such that the two terms $\lambda\|T\|_{\mathcal{T}}$ and $MMD^2(Z + V, X)$ are on the same order.

The optimization problem was solved using L-BFGS, a quasi-Newton method that uses H_k, an approximation of the Hessian inverse, with the update formula given by

$$x_{k+1} = x_k - \alpha_k \cdot H_k \cdot \nabla f(x_k)$$

where ∇f is the gradient of the loss function and α_k is the step size at a given step k. The gradient was calculated using JAX.

To update each step, we define variables

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k), \quad \rho_k = \frac{1}{y_k^\top s_k}$$

where H_k is updated using

$$H_{k+1} = \left(I - \rho_k s_k y_k^\top\right) H_k \left(I - \rho_k y_k s_k^\top\right) + \rho_k s_k s_k^\top.$$

Now, let $q = \nabla f(x_k)$. The algorithm is then run with two loops, one backward loop defined with

$$\rho_i = \frac{1}{y_i^\top s_i}, \quad \alpha_i = \rho_i s_i^\top q, \quad q = q - \alpha_i y_i$$

in order to calculate

$$H_k^0 = \frac{s_{k-1}^\top y_{k-1}}{y_{k-1}^\top y_{k-1}} I, \quad r = H_k^0 q,$$

and one forward loop defining

$$\beta_i = \rho_i y_i^\top r, \quad r = r + s_i(\alpha_i - \beta_i)$$

This yields the search direction $p_k = -r$, and the update is:

$$x_{k+1} = x_k + \alpha_k p_k.$$

We define our variables `max_iter = 100, tol = 1e-5` where we want $\|\nabla f\| < tol$ or $k \leq max\_iter$, along with checking whether our MMD decreases at each iteration.

2.3. **Part 3.** Once the transport model is developed, the transport map was trained using the RBF kernel, Laplace kernel, and polynomial kernel of degree $d = 1, 2, 4$. using the definitions from Part 1. A heatmap of the transported samples will be displayed. In testing the effects of parameters on the model, varying values for the numbers of training samples N will be explored, with $N = 500, 1000, 2000$.

3. Results

3.1. **Part 1.** The computed MMD distance between η and ν for each of the three target distributions are shown in Table 1. From this, we see that RBF has the smallest consistent MMD distance for all three targets, on the order of 10^{-3}, while the MMD distance for the Laplacian is larger, on the order of 10^{-2}. For the polynomial, the MMD varied wildly, with the largest error for moons on the order of 10^{-1} and the smallest error with the pinwheel, on the order of 10^{-5}. As such, RBF has the better all-around kernel to use for the various distribution, while polynomial has inconsistent performance depending on what target distribution it is applied to.

Target	rbf	laplacian	poly
moons	0.003421	0.014184	0.387632
swissroll	0.005956	0.010922	0.022059
pinwheel	0.002376	0.012630	0.000077

Table 1. Empirical MMD distances between η and ν for the three target distributions: RBF, Laplacian, and polynomial with degree 2.

3.2. **Part 2.** We analyze the effectiveness of L-BFGS by visualizing $\|\nabla f\|$ and the MMD loss at each step (Figure 2). Our plot indicates a decrease in both $\|\nabla f\|$ and the MMD loss as the algorithm progresses, indicating our transport map does indeed transport our reference distribution closer to our target distributions.

3.3. **Part 3.** The empirical samples generated by the transport map using the RBF and Laplacian kernel were plotted with a heat map given $N = 5000$ and an initial $\lambda = 10^{-4}$. For the RBF kernel, we see the samples match the target distributions, as evidenced by the three discernible targets (Figure 3). The same can be said for the Laplace kernel, with brighter points (i.e. higher density) where our target distribution should be (Figure 4). Upon further inspection, we observe the RBF kernel had more points scattered throughout the plot, while the Laplace kernel had a cleaner plot, with points that were transported closer onto the target distribution.

We used $N = 1000$ for the polynomial kernels because the code was unable to run and converge for $N = 5000$, and this can be seen in the heatmaps (Figure 5). We observe linear behavior in the transported samples generated by the degree 1 polynomial, with an increasing slope for the moons, and a decreasing slope for the other two targets. For the degree 2 and 3 polynomial, the points

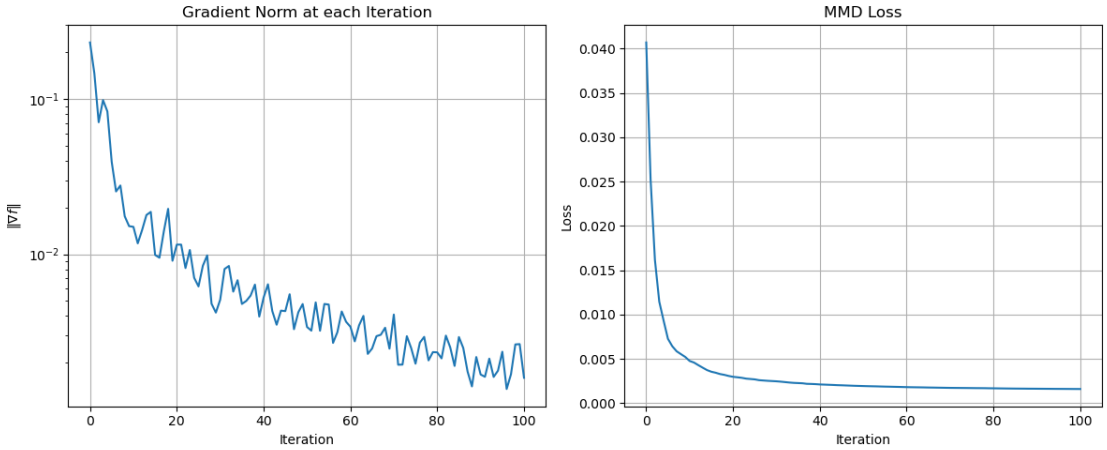

FIGURE 2. Plot of gradient norm (left) and MMD loss (right) at each iteration when running L-BFGS with the transport map using the RBF kernel and target Pinwheel.

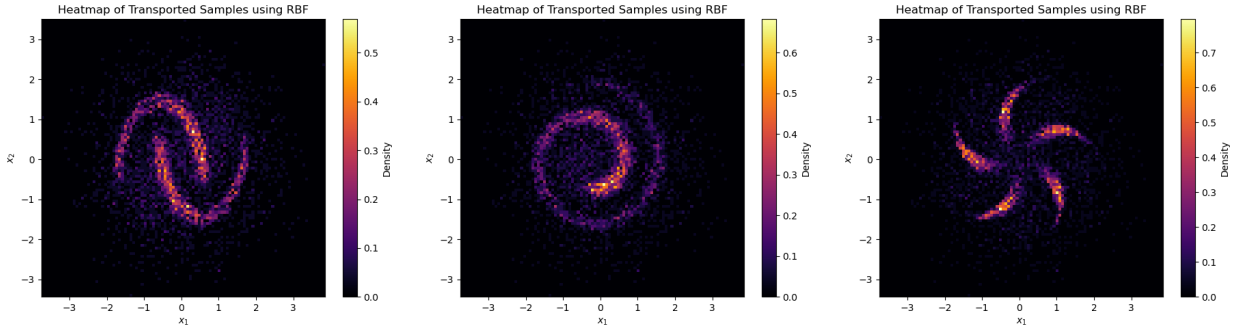

FIGURE 3. Heatmap of samples generated by the transport map using the RBF kernel, with $N = 5000$. The distributions displayed from left to right are moons, swissroll, and pinwheel.

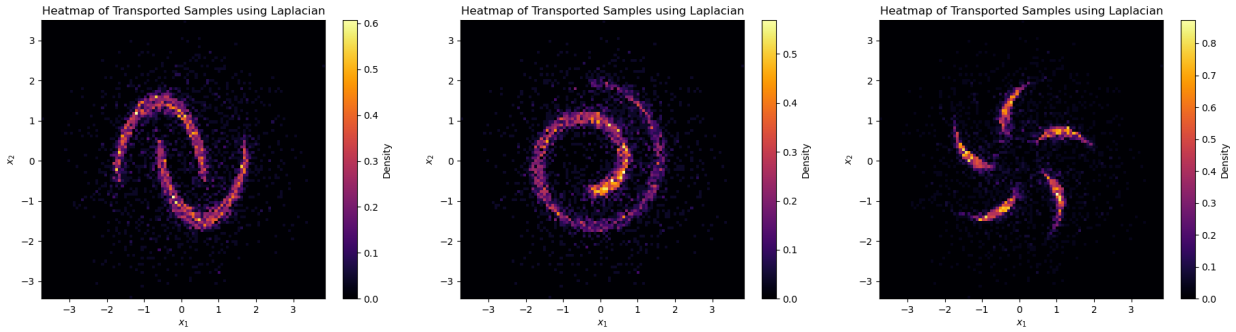

FIGURE 4. Heatmap of samples generated by the transport map using the Laplacian kernel, with $N = 5000$. The distributions displayed from left to right are moons, swissroll, and pinwheel.

remain dispersed in a cloud around 0. The brightest cloud that appears is with the polynomial of degree 2 for moons where it appeared stretched out towards the top left and bottom right in the shape of an ellipse.

Next, we observe the effect of the value of $N = 500$ and $N = 5000$ on the transport map. For RBF we can note the outline of the shapes starting to form, though it still remains blurry as the points are still scattered among the cloud (Figure 7). This is particularly prevalent with $N = 500$ where the samples remain dispersed. A clearer shape forms when $N = 2000$, but still not as clear as when $N = 5000$.

Next, we consider the Laplacian kernel for $N = 500, 2000$ where we observe similar behavior to the RBF kernel (Figure 8). For $N = 500$, the points remain scattered and the shape of the target distributions can be barely seen, especially with the moons. When we raise the number to $N = 2000$, we achieve far more distinct shapes, even compared to the RBF kernel where we can see the shapes.

Finally, we consider the heatmap of the samples generated from polynomial kernel of degree 1 for $N = 500, 2000$ (Figure 9). We see that for the lower value of $N = 500$, the samples remain within the point cloud around 0. We only see a slight positive linear correlation between x_1 and x_2 for the pinwheel data set. When we increase to $N = 2000$, we observe similar to behavior as when $N = 1000$, though with the point cloud appearing denser here due to the increased number of points.

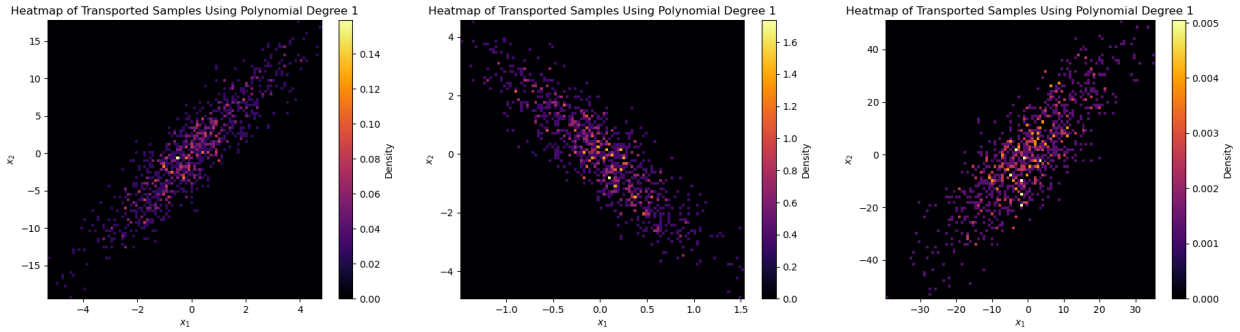

FIGURE 5. Heatmap of samples generated by the transport map using the polynomial kernel with degree 1 and $N = 1000$. The distributions displayed from left to right are moons, swissroll, and pinwheel.

4. Summary and Conclusions

When calculating the MMD between the standard distribution and the three target distributions, the RBF kernel had the most consistent low values across all the target distribution, followed by the Laplace kernel. The polynomial kernel varied the most in MMD between the three targets, where it had the highest and lowest MMD overall with the moons and pinwheel, respectively.

From observing the heatmaps of the samples generated by the transport maps, the RBF and Laplace Kernels were able to clearly reconstruct the target distributions, with the Laplace kernel having the clearest distribution, i.e. highest visual quality of the samples.

Meanwhile, all of the polynomial kernels were unable to capture the shape of the target distributions. The degree 1 polynomial captured a linear relationship in the samples, while a point cloud was observed for the degree 2 and 4 polynomial with no clear distinction for their original targets.

In altering the values of N, we observe that as N decreases, there is less convergence to the target distributions for all kernels. For the RBF and Laplace kernels, we see the shape of the target

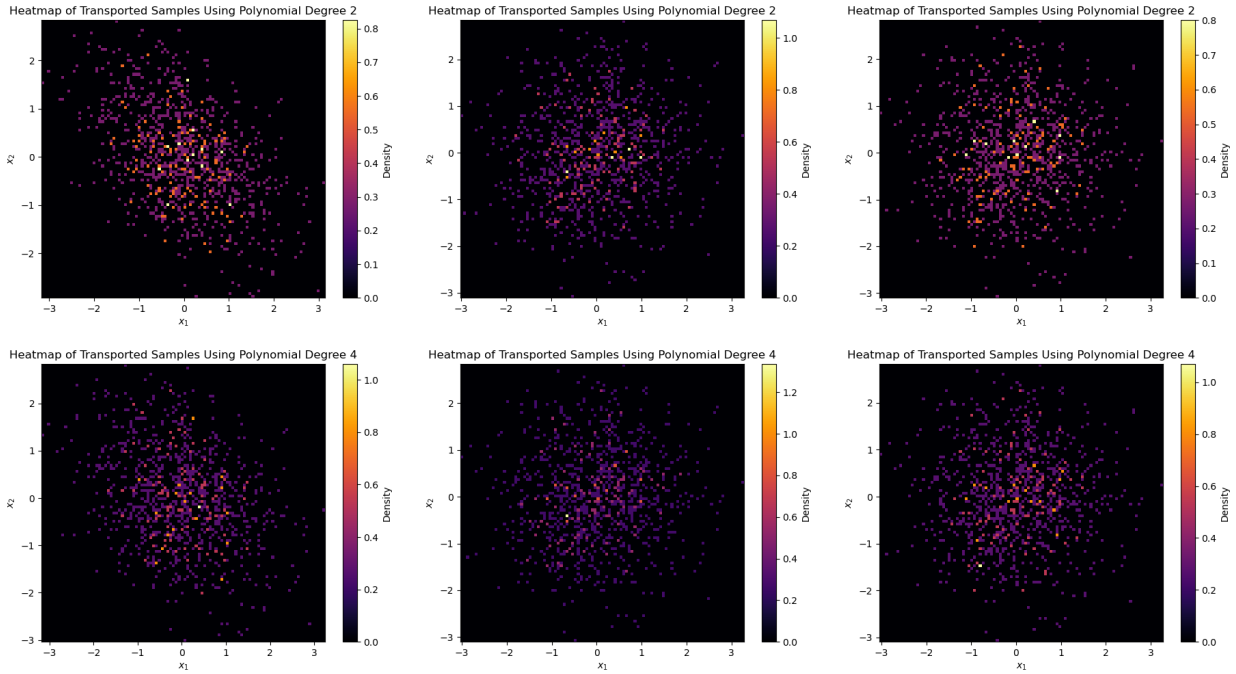

FIGURE 6. Heatmap of samples generated by the transport map using the polynomial kernel with degree 2 (top row) and degree 4 (bottom row), with $N = 1000$. The distributions displayed from left to right are moons, swissroll, and pinwheel.

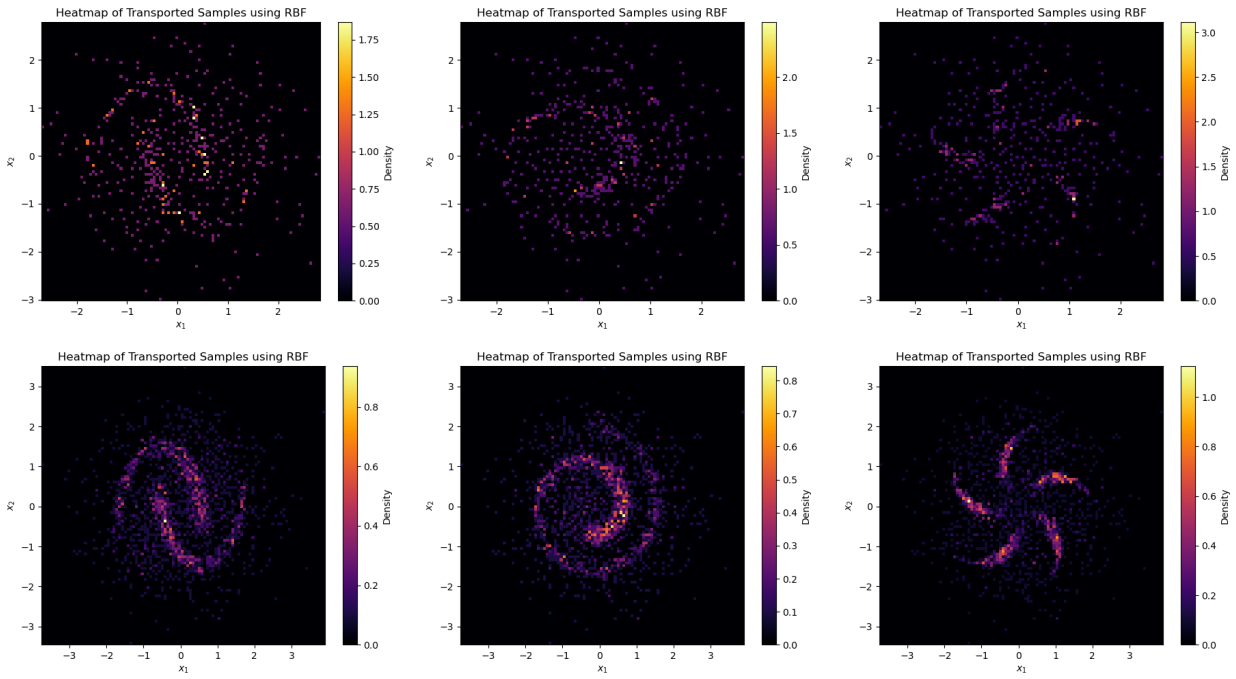

FIGURE 7. Heatmap of samples generated by the transport map using the RBF kernel, with $N = 500$ (top row) and $N = 2000$ (bottom row). The distributions displayed from left to right are moons, swissroll, and pinwheel.

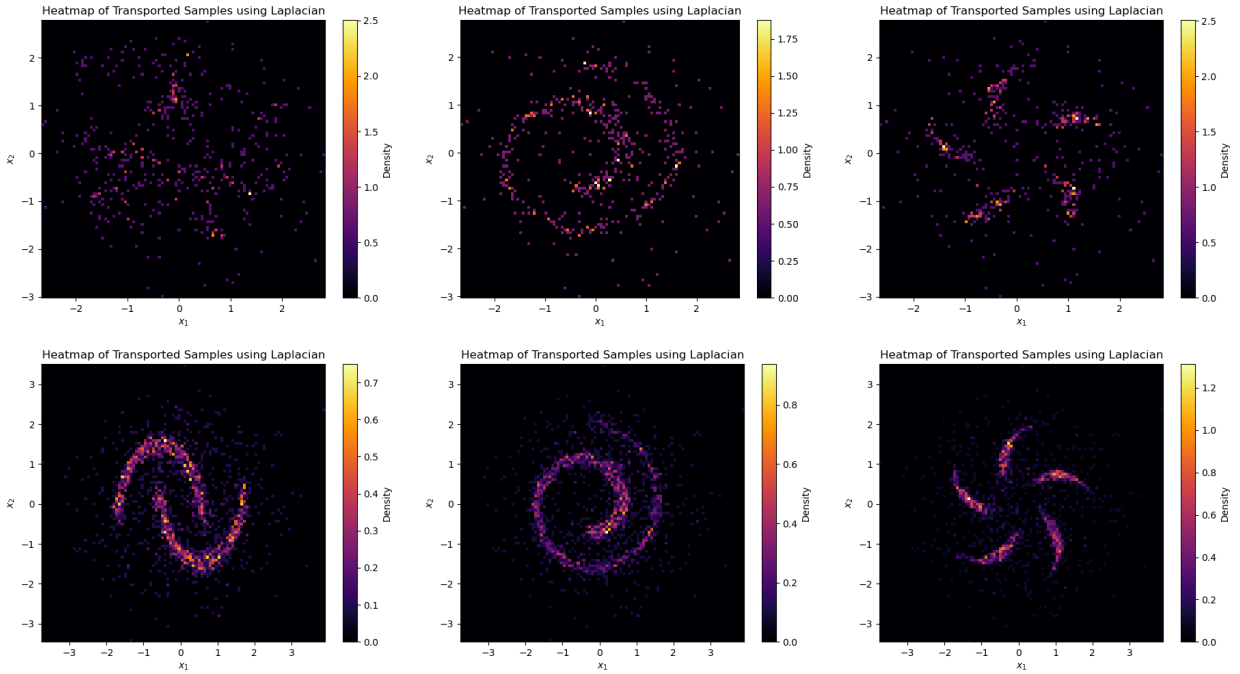

FIGURE 8. Heatmap of samples generated by the transport map using the Laplacian kernel, with $N = 500$ (top row) and $N = 2000$ (bottom row). The distributions displayed from left to right are moons, swissroll, and pinwheel.

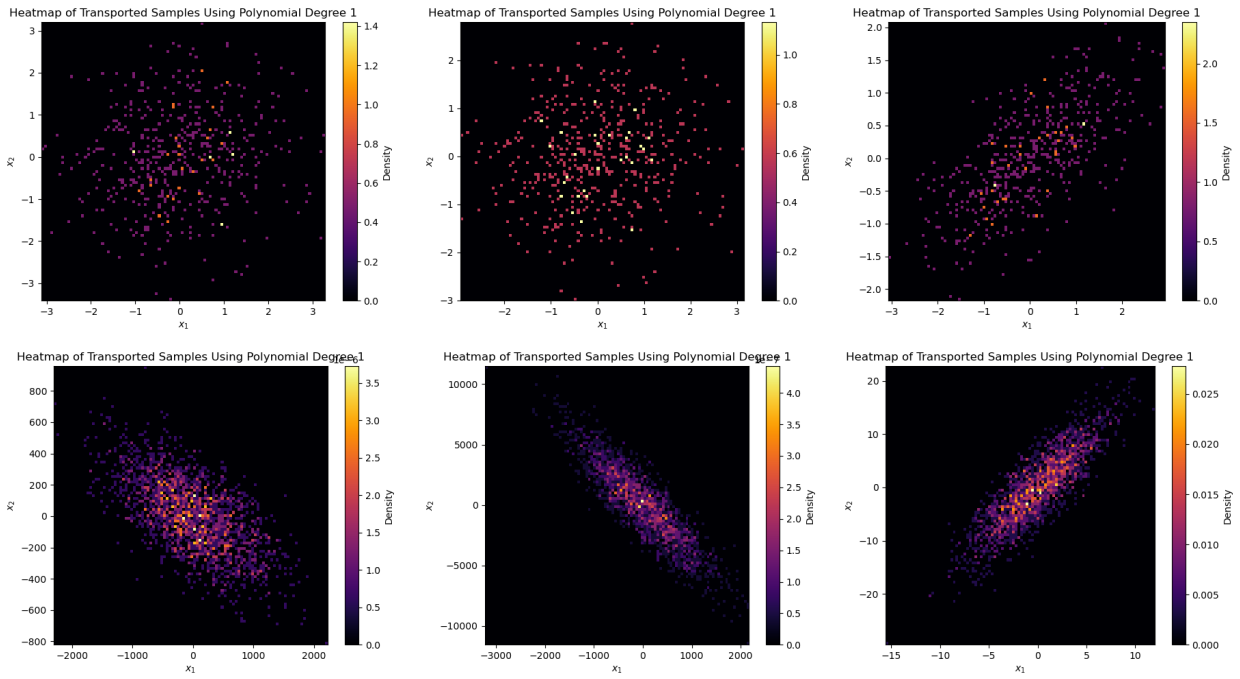

FIGURE 9. Heatmap of samples generated by the transport map using the polynomial kernel of degree 1, with $N = 500$ (top row) and $N = 2000$ (bottom row). The distributions displayed from left to right are moons, swissroll, and pinwheel.

distributions forming, but still many samples scattered around. For the degree 1 polynomial, we lose the linear correlation of the points with fewer N.

Possible future directions includes attempting more hyper-parameter tuning to see if that makes further improvements in our transport problem. Moreover, it will be interesting to see how well these kernels perform with the other distributions defined from the set-up code, such as the "rings" or "checkboard," and compare which kernels have better performance.

Acknowledgements

The author is thankful to Prof. Bamdad Hosseini for useful discussions about MMD and the transport models. We are also thankful to Christina, Nghi, Emily, David, Tom, Howard, and others for productive discussions about the theory and implementation of the transport model in Python.