Investigating the Intrinsic Geometry of the CelebA Dataset Using Diffusion Geometry

Ethan Arnold

September 2025

Undergraduate Summer Research Project

Durham University

Funded By The London Mathematical Society

Abstract

We investigate the intrinsic dimension and curvature of the CelebA dataset using recently developed diffusion geometry techniques. Recent results have suggested the CelebA dataset may have hyperbolic geometry. Using these techniques we discover that the dataset likely has both hyperbolic and elliptic regions.

Acknowledgements

I would like to thank Prof. Jeffrey Giansiracusa for his continued guidance and support during this project.

Contents

1 Introduction

1.1 Motivation

The CelebFaces Attributes Dataset, commonly known as CelebA, is a facial attributes dataset comprising 202,599 images of celebrities [1]. CelebA is a popular benchmark dataset in machine learning, often used for training and testing models for tasks such as facial recognition.

Julian Chang Jun Yeng discovered hyperbolic variational autoencoders outperform Euclidean variational autoencoders when reconstructing the CelebA images [2]. This suggests that the CelebA dataset may have intrinsic hyperbolic geometry.

To further investigate the intrinsic geometry of the CelebA dataset, we employ novel tools developed by Iolo Jones [3]. These tools utilise diffusion geometry techniques to estimate the dimension and curvature of data from manifolds.

1.2 Aims of the Project

In this summer research project, we first aim to understand relevent areas of differential geometry. We introduce surfaces and study the second fundamental form and Gaussian curvature.

We then study the theory of Diffusion Maps. We build a Python algorithm that utilises diffusion maps to reduce the dimension of the CelebA images to three, in order to produce a visual representation of the entire dataset.

Finally, we implement the Python tools developed by Iolo Jones [3], aiming to investigate the intrinsic dimension and curvature of the CelebA dataset.

2 Mathematical Background

In this section, we present the neccessary background required to understand Gaussian and scalar curvature. We also explain diffusion maps and why they are useful for dimensionality reduction.

2.1 Surfaces in Euclidean Space and Gaussian Curvature

The information presented in this chapter is mainly adapted from Curves and Surfaces, by Montiel and Ros [4].

2.1.1 Defining Surfaces and Curves

Informally, a set $S \in \mathbb{R}^3$ is a surface if, at every point in S, S locally looks like a slightly deformed patch of $\mathbb{R}^2$. We rigourously define a surface below, and provide a visual representation in Figure 2.1.

Definition 2.1. Consider a non-empty set $S \subset \mathbb{R}^3$. We say S is a **surface** if for all $p \in S$, there exists an open neighbourhood $V \subset S$ of p, an open set $U \subset \mathbb{R}^2$, and a map $X : U \to V$, where X has the following properties:

- **i)** $X : U \to V$ is a homeomorphism.
- **ii)** $(dX)_q : \mathbb{R}^2 \to \mathbb{R}^3$ is injective for all q in U.

We call each such $X : U \to S$ a **parametrisation of** S.

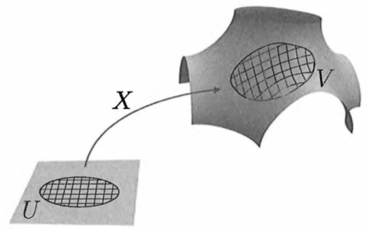

Figure 2.1: Diagram of a parametrisation of some surface in $\mathbb{R}^3$.

Remark 2.2. Property **(i)** means X is continuous, bijective, and has a continuous inverse.

Remark 2.3. In property **(i)**, $(dX)_q$ is called a differential, and, in this case, can be defined as follows. If we write

$$X(u,v) = (X_1(u,v), X_2(u,v), X_3(u,v))$$

then

$$(dX)_q(a,b) = \begin{bmatrix} \frac{\partial X_1}{\partial u}(q) & \frac{\partial X_1}{\partial v}(q) \\ \frac{\partial X_2}{\partial u}(q) & \frac{\partial X_2}{\partial v}(q) \\ \frac{\partial X_3}{\partial u}(q) & \frac{\partial X_3}{\partial v}(q) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}.$$

One can think of $(dX)_q$ here as the best linear approximation of the function X at q. We revisit differentials in Section 2.1.4.

Note 2.4. Surfaces can be embedded in higher dimensions than three, and we can also define hypersurfaces with dimensions greater than two. However, here we focus on 2-dimensional surfaces in $\mathbb{R}^3$ to aid visualisation.

Definition 2.5. We define a **curve in** $\mathbb{R}^3$ as a C^∞ map $\alpha : I \to \mathbb{R}^3$ for some open $I \subset \mathbb{R}$. If $\alpha(I) \subset S$ for some surface S, we say α is a **curve on** S.

2.1.2 Differentiable Functions on Surfaces

Consider a function defined on a surface. We must define what it means for such a function to be differentiable. To do this, we use parametrisations and inclusion maps to reduce the concept to our familiar notion of differentiability.

Definition 2.6. Given a surface $S \subset \mathbb{R}^3$, the **inclusion map** $i : S \to \mathbb{R}^3$ is the identity map, regarded as having a different domain and codomain. That is, $\forall\, p \in S$, $i(p) = p$ but we say $i(p) \in \mathbb{R}^3$.

Definition 2.7. Let S be a surface.

(A) A function $f : S \to \mathbb{R}^m$ is **differentiable** if, for every parametrisation $X : U \to S$, the composition $f \circ X : U \to \mathbb{R}^m$ is differentiable (in the usual Euclidean sense).

(B) Let $O \subset \mathbb{R}^n$ be open. A map $f : O \to S$ is **differentiable** if the composition $i \circ f : O \to \mathbb{R}^3$ is differentiable, where $i : S \to \mathbb{R}^3$ is the inclusion map.

(C) If S_1 is another surface and $i_1 : S_1 \to \mathbb{R}^3$ is its inclusion map, a map $f : S \to S_1$ is **differentiable** if $i_1 \circ f : S \to \mathbb{R}^3$ is differentiable in the sense of **(A)**.

2.1.3 Tangent Planes

Another important notion in differential geometry is that of the tangent plane to a surface at a given point.

Definition 2.8. Let S be a surface and $p \in S$. A vector $v \in \mathbb{R}^3$ is **tangent to S at p** if there exists a curve on S

$$\alpha : (-\epsilon, \epsilon) \to S$$

for some $\epsilon > 0$, such that $\alpha(0) = p$ and $\alpha'(0) = v$. The set of all vectors tangent to S at p is called the **tangent plane of S at p** and is denoted T_pS.

2.1.4 The Differential of a Differentiable Map

The notion of the differential of a map between Euclidean spaces, aforementioned in Remark 2.3, can be extended to surfaces.

Definition 2.9. Let S_1 and S_2 be surfaces and $f : S_1 \to S_2$ be a differentiable function. For a point $p \in S_1$ the **differential of f at p** is a map

$$(df)_p : T_pS_1 \to \mathbb{R}^3$$

defined as follows: Given $v \in T_pS_1$, choose a curve $\alpha : I \to S_1$ such that $\alpha(0) = p$ and $\alpha'(0) = v$. Then

$$(df)_p(v) = \frac{d}{dt}\bigg|_{t=0} (f \circ \alpha)(t) = (f \circ \alpha)'(0).$$

It can be shown that $(df)_p(v)$ does not depend on the chosen curve, α, hence is well defined. One can also prove that it is a linear map.

Remark 2.10. Since $f : S_1 \to S_2$ and $\alpha : I \to S_1$ are both differentiable functions, one can prove that $f \circ \alpha : I \to S_2$ is a differentiable function. Hence, it is a curve on S_2 with $(f \circ \alpha)(0) = f(p)$ and $(f \circ \alpha)'(0) = (df)_p(v)$. This means $(df)_p(v) \in T_{f(p)}S_2$. So we can think of the differential of f at p as being a map

$$(df)_p : T_pS_1 \to T_{f(p)}S_2.$$

2.1.5 The Gauss Map

Definition 2.11. Let $S \subset \mathbb{R}^3$ be a surface. A **unit normal field** on S is a differentiable map $N : S \to \mathbb{R}^3$ such that for all $p \in S$ we have $|N(p)| = 1$ and for all $v \in T_pS$ we have $N(p) \perp v$.

Definition 2.12. If one can define a single unit normal field on the entire surface, we call this unit normal field a **Gauss map**. In this case we call the surface **orientable**.

Remark 2.13. Let S be an orientable surface and $N : S \to \mathbb{R}^3$ be a Gauss map. Since $|N(p)| = 1$ for all $p \in S$ we have $N(S) \subset \mathbb{S}^2$, where $\mathbb{S}$ is the unit sphere centered at the origin. Hence, we can consider the Gauss map as $N : S \to \mathbb{S}^2$.

2.1.6 Gaussian Curvature

One can show that $\mathbb{S}^2$ is a surface. Hence a Gauss map is a differentiable function between two surfaces. As a result, we can calculate its differential at a point.

Let $S \in \mathbb{R}^3$ be an orientable surface, $N : S \to \mathbb{S}^2$ be a Gauss map, and $p \in S$. The differential can be written $(dN)_p : T_pS \to T_{N(p)}\mathbb{S}^2$. However, note that $T_{N(p)}\mathbb{S}^2$ is the set of all vectors orthogonal to $N(p)$. Hence, it can be proven, and intuitively seen, that $T_{N(p)}\mathbb{S}^2 = T_pS$. This means $(dN)_p$ is an endomorphism of T_pS.

Definition 2.14. Let $S \in \mathbb{R}^3$ be an orientable surface with Gauss map $N : S \to \mathbb{S}^2$. Let $p \in S$ and $(dN)_p : T_pS \to T_pS$ be the differential of N at p. Since T_pS is 2-dimensional, given a basis of T_pS one can represent the endomorphism $(dN)_p$ by a 2×2 matrix.
We define the **Gaussian curvature at** p to be the determinant of this matrix, and denote it by $K(p)$.
One can show the determinant does not depend on the chosen basis of T_pS, so the Gaussian curvature is well defined.

Note 2.15. To intuitively see why the Gaussian curvature relates to the shape of the surface, note that the Gauss map is a normal field on S. Hence, the differential of the Gauss map describes the variation of a normal field on the surface. It is not hard to see why this would give us information about the shape of the surface. The Gaussian curvature is just an associated invariant of this differential.

The sign of the Gaussian curvature at a point reveals important information about the geometry of the surface at that point. Positive Gaussian curvature corresponds to an elliptic point, for example a point in a bowl. Negative Gaussian curvature corresponds to a hyperbolic point, for example a saddle point. Zero Gaussian curvature corresponds to a point that is flat in at least one direction.

Note 2.16. In Section 3, we reference scalar curvature rather than Gaussian curvature. However, for 2-dimensional surfaces, the scalar curvature is just double the Gaussian curvature, so they have the same sign.

2.1.7 The Second Fundamental Form

Definition 2.17. The **second fundamental form of** S **at** p is the bilinear form $\sigma_p : T_pS \times T_pS \to \mathbb{R}$, where

$$\sigma_p(v, w) = -\langle\, (dN)_p(v)\, , w \,\rangle.$$

2.2 Diffusion Maps

2.2.1 Construction of a Random Walk on the Data

Suppose we have m data points, each of which is represented by a vector in $\mathbb{R}^n$. We will denote this dataset by X. We consider each data point as a node on a graph, and we aim to construct a random walk on this graph [5].

We define diffusion maps in general, however, it might be intuitively helpful to consider the following example given in Figure 2.2.

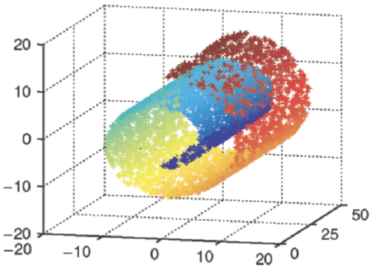

Figure 2.2: 5,000 points sampled from a Swiss roll in $\mathbb{R}^3$ [6]

We first define a weight function $k : X \times X \to \mathbb{R}$ by

$$k(x_i, x_j) = e^{\frac{-|x_i - x_j|}{\epsilon}},$$

where $\epsilon > 0$ is some constant we discuss in Section 2.2.4.

This weight function allocates a larger weight between data points that are closer together in the ambient space $\mathbb{R}^n$.

Next, we define the degree function

$$d(x_i) = \sum_{x \in X} k(x_i, x).$$

We use this to normalise the weight function and create transition probabilities.

We define the matrix P with entries

$$P_{ij} = \frac{k(x_i, x_j)}{d(x_i)}.$$

P is a stochastic matrix as its rows sum to one. Hence, we can define a Markov process on our graph by setting the transition probability of going from x_i to x_j in one discrete time step to be P_{ij}.

Note 2.18. Since this Markov chain is clearly irreducible and aperiodic, it has a unique stationary distribution $\pi \in [0,1]^m$. [7]

2.2.2 Diffusion Distance and Diffusion Maps

Definition 2.19. The **diffusion distance** between $x_i, x_j \in X$ is

$$D_t(x_i, x_j)^2 = \sum_{y \in X} \frac{(\, P^t_{x_i,y} - P^t_{x_j,y}\,)^2}{\pi(y)} \tag{1}$$

where π is the unique stationary distribution of the Markov chain.

Note 2.20. Intuitively, the diffusion distance is a measure of how 'connected' two points are in the intrinsic geometry of the dataset. In some sense it is a measure of the probabilility of diffusing from one point to another along this Markov chain we have created. Points may be close together in the Euclidean sense but have a diffusion distance that is large. For example, the blue and red sections in Figure 2.2.

Definition 2.21. We define the **diffusion map** $\Psi_t : X \to \mathbb{R}^n$ as

$$\Psi_t(x) = \begin{bmatrix} \lambda_1^t \psi_1(x) \\ \vdots \\ \lambda_n^t \psi_n(x) \end{bmatrix}$$

where $\{\psi_l\}$ are the left eigenvectors of P corresponding to eigenvalues $\{\lambda_l\}$.

Theorem 2.22. *Given $x_i, x_j \in X$ we have*

$$D_t(x_i, x_j)^2 = |\Psi_t(x_i) - \Psi_t(x_j)|^2 \tag{2}$$

That is, the diffusion distance is equal to the Euclidean distance between the diffusion maps. The proof revolves around the eigenvector expansion of P. [8]

Remark 2.23. This Theorem implies two points whose diffusion maps are close in the Euclidean sense have a strong 'connectivity'.

2.2.3 Dimension Reduction

Consider the coordinates of the diffusion map corresponding to the smaller eigenvalues. When caluclating the Euclidean distance between diffusion maps, i.e. the right hand side of equation (2), these coordinates have less influence due to their smaller magnitude. This allows us to prove an approximation of Equation (2).

Definition 2.24. Firstly, we reorder the eigenvalue and eigenvector pairs so that $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. We then define the **truncated diffusion map of order k**, $\Psi_t^{(k)} : X \to \mathbb{R}^k$, as

$$\Psi_t^{(k)}(x) = \begin{bmatrix} \lambda_1^t \psi_1(x) \\ \vdots \\ \lambda_k^t \psi_k(x) \end{bmatrix}.$$

Equation (2) now becomes

$$D_t(x_i, x_j)^2 \approx |\Psi_t^{(k)}(x_i) - \Psi_t^{(k)}(x_j)|^2.$$

This is formally proven in [8].

By creating these truncated diffusion maps we can represent the data points of X, originally embedded in $\mathbb{R}^n$, with points now embedded in $\mathbb{R}^k$ where $k < n$. The distance between the resulting points in $\mathbb{R}^k$ is a measure of how 'connected' the points of the original dataset were.

We now revisit the exmple given in Figure 2.2.

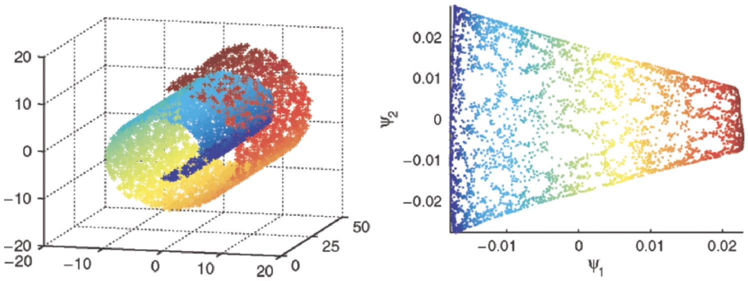

Figure 2.3: 5,000 points sampled from a Swiss roll in $\mathbb{R}^3$ (L). The truncated diffusion maps of order 2 (R). [6]

The data originally lived in $\mathbb{R}^3$, but when we plot the truncated diffusion maps of order 2 we obtain the plot given on the right of Figure 2.3. The diffusion maps have essentially uncovered the coordinates of the swiss roll and 'unravelled' it.

2.2.4 Revisiting the Parameter ϵ and Discussing t

When defining the random walk on our data in Section 2.2.1, we defined the weight function $k : X \times X \to \mathbb{R}$ by

$$k(x_i, x_j) = e^{\frac{-|x_i - x_j|}{\epsilon}}.$$

The parameter ϵ must be chosen when creating this kernel, and the value one picks is important.

A larger ϵ means larger transition probabilities will be allocated to points father away from each other in the ambient Euclidean space. Conversely, a smaller ϵ means points farther away will be allocated smaller transition probabilities.

The idea is that for ϵ too large, the graph will be too 'connected' so the diffusion process will provide little information about the intrinsic geomtry of the data. Conversely, for epsilon too small, very little diffusion takes place, which again provides little information about the geometry of the surface.

There is a technique for estimating a good ϵ to use, which we explore in Section 3.1.

When defining the diffusion distance and diffusion maps, we have also seen the parameter t arise. The value of t we use when creating the diffusion maps decides how many steps we wish to take along our Markov chain. Adjusting this can change whether our diffusion maps pick up large scale or small scale features of the data. For the purposes of this project we have used $t = 1$.

3 New Results

3.1 Estimating a Sensible ϵ for the CelebA Dataset

As previously mentioned in Section 2.2.4 there is a method for estimating an appropriate ϵ to use when forming our diffusion maps. [9]

For a given ϵ we calculate the weight matrix with entries $W_{ij} = e^{\frac{-|x_i - x_j|^2}{\epsilon}}$. We then sum over every entry of this matrix to obtain the sum of all the weights between our points, $\sum_{i,j} W_{ij}$.

The magnitude of this sum relates to how 'connected' the weighted graph we form is, and hence how much diffusion takes place in our final Markov chain. We plot the value of this sum against various ϵ values, shown in Figure 3.1.

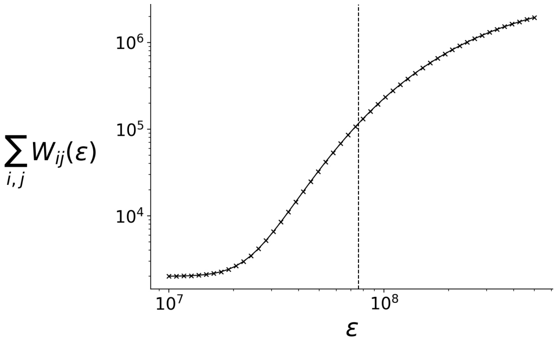

Figure 3.1: Logarithmic scale plot of $\sum_{i,j} W_{ij}$ against ϵ for 2,000 images of the CelebA dataset.

Coifman suggests that to ensure a meaningful amount of diffusion takes place, one should take an ϵ at the center of the linear section of this logarithmic scale plot.

In the case of the CelebA dataset, this is about $76,000,000$, so this is the value of ϵ we shall use henceforth.

3.2 Reducing the Dimension of CelebA Images Using Diffusion Maps

By following the steps in Section 2.2 we can construct a visual 3-dimensional representation of the CelebA dataset by creating truncated diffusion maps of order three.

The Python code used to do this is available at
https://github.com/ethanarnold45/CelebA_Project.git.

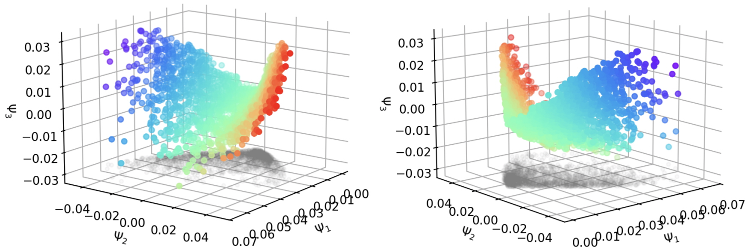

Figure 3.2: 2,000 images from the CelebA dataset reduced to 3-dimensions using diffusion maps.

In Figure 3.2 we obtain a curved 2-dimensional surface embedded in three dimensions. The fact that we obtain an obvious pattern rather than a random assortment of points suggests the diffusion maps have indeed uncovered some of the intrinsic geometry of the dataset.

3.3 Estimating the Dimension of the CelebA Dataset

Using Diffusion Geometry techniques one can estimate the intrinsic dimension of a dataset. For example, the intrinsic dimension of the Swiss roll in Figure 2.2 is two almost everywhere.

The techniques we will use in this section and Section 3.4 were developed by Iolo Jones [3], whose code is available at
https://github.com/Iolo-Jones/ManifoldDiffusionGeometry.git. The code used to apply Jones' tools to the CelebA dataset and produce the figures in these sections is available at
https://github.com/ethanarnold45/CelebA_Project.git.

We first use diffusion maps to reduce the dimension of the CelebA dataset to five. We then use Jones' tools to estimate the local intrinsic dimension of the dataset at each point. We then plot these dimension estimate on our 3-dimensional plot created in Section 3.2.

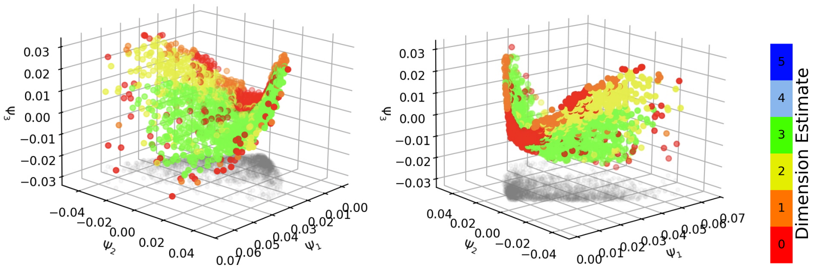

Figure 3.3: Local intrinsic dimension estimates of 2,000 CelebA images that have been reduced to 5 dimensions using diffusion maps.

Note 3.1. The dimension estimates shown in Figure 3.4 are calculated in five dimensions, despite the figure being represented in three dimensions.

Figure 3.3 suggests that when we reduce the dimension of the ambient space to five, our data primarily lives on a three dimensional space within this ambient space.

This could suggest that Jones' tools are recognising three significant parameters in this 5-dimensional space. More work is required to discover what these parameters may be.

We also see clusters of different estimations arise. This could perhaps mean that certain parameters are not relevant in certain groupings. An analogy could be that if one parameter we are picking up is the length of someones hair, this parameter could be irrelevent in the images of people with bald hair. Again, more research is required to determine what the actual reason for these clusters is.

3.4 Estimating the Curvature of the CelebA Dataset

In Section 3.3 we discovered that when we reduce the dimension of the ambient space to five, the majority of our data lives in a 3-dimensional subset of this. We use Jones' tools to estimate the scalar curvature of this 3-dimensional subset.

Note 3.2. We cannot estimate the curvature of the dataset in the original high dimensional space because using diffusion geometry for geomtric inference is only robust when the intrinsic dimension of the data is low [3].

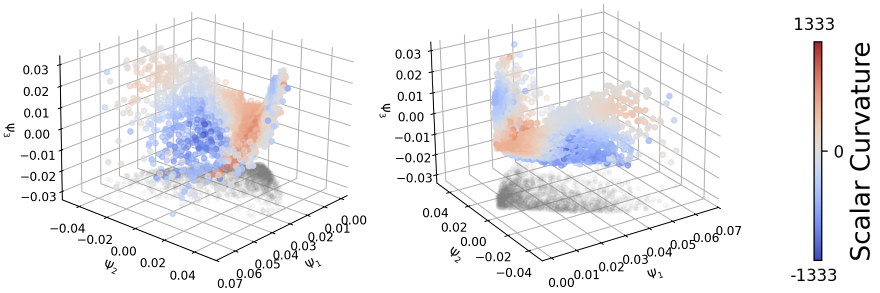

Figure 3.4: Local scalar curvature estimates of 2,000 CelebA images that have been reduced to 5 dimensions using diffusion maps. When estimating curvature we have assumed the data lives on a 3-dimensional subset of the ambient space.

Figure 3.4 shows a large cluster of points with negative curvature estimates, and a large cluster with positive scalar curvature estimates.

This may indicate the CelebA dataset has both hyperbolic regions and elliptic regions. More work is required to determine if this is the case, and if so whether we can categorise these regions.

4 Conclusions

We have employed diffusion maps to reduce the dimension of the CelebA dataset to five. Then, using recently developed diffusion geometry techniques, we determined that the dataset appears to live on a 3-dimensional subspace of this 5-dimensional ambient space.

Under the assumption this is true, we have estimated the local scalar curvature of this 3-dimensional subspace. We discovered that some clusters have negative scalar curvature and some clusters have positive scalar curvature. This suggests the CelebA dataset has both elliptic and hyperbolic regions.

Opportunities for further research include determining what the three parameters we are seeing in Section 3.3 are. Perhaps it could be glasses, facial expression, hats, etc. The fact that we are clearly picking up only three parameters may suggest that these parameters are being deemed more significant than others.

One could also attempt to categorise the hyperbolic regions and elliptic regions of CelebA. For example, perhaps smiling faces have elliptic geometry and others have hyperbolic geometry. This could potentially aid the development of geometric deep learning models as it might suggest certain models will perform better on given subsets of human photos.

References

[1] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[2] Julian Chan Jun Yen. Hyperbolic neural networks: Feed-forward networks and variational autoencoders for the poincare ball model of hyperbolic space. Master's thesis, Swansea University, 2021.

[3] Iolo Jones. Manifold diffusion geometry: Curvature, tangent spaces, and dimension. *arXiv preprint arXiv:2411.04100 [math.DG]*, 2024.

[4] Sebastian Montiel and Antonio Ros. *Curves and Surfaces*. American Mathematical Society, USA, 2009.

[5] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–11, 2006.

[6] Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis G. Kevrekidis. Diffusion maps – a probabilistic interpretation for spectral embedding and clustering algorithms. In Alexander N. Gorban, Balázs Kégl, Donald C. Wunsch, and Andrei Y. Zinovyev, editors, *Principal Manifolds for Data Visualization and Dimension Reduction*, volume 58 of *Lecture Notes in Computational Science and Engineering*, page 250. Springer Verlag, 2008.

[7] J. R. Norris. *Markov Chains*. Cambridge University Press, Cambridge, UK, 1997.

[8] Boaz Nadler, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of fokker–planck operators. In *Advances in Neural Information Processing Systems*, volume 18, 2005.

[9] Ronald R. Coifman, Yoel Shkolnisky, Fred J. Sigworth, and Amit Singer. Graph laplacian tomography from unknown random projections. *IEEE Transactions on Image Processing*, 17(10):1896–1897, 2008.