

Analyzing product manifolds using diffusion maps

Sharon Zhang

Advised by:

Dr. Amit Moscovich and Professor Nicolas Boumal

Junior Paper
Department of Mathematics
Princeton University



*This paper represents my own work in accordance
with University regulations. /s/ Sharon Zhang*

May 15, 2020

Abstract

Diffusion maps are a spectral method for dimensionality reduction which exploit the intrinsic geometry of data. Given a set of points sampled from a manifold, the eigenvectors of the diffusion map formed by those samples are discrete approximations of the Laplace eigenfunctions of the manifold. Using this relationship, we propose an algorithm for decomposing product manifolds into independent submanifolds and test it on simulated data sampled from submanifolds of \mathbb{R}^D . Our results show that the algorithm performs well on these manifolds, and can potentially be adapted for applications such as single-particle reconstruction in cryo-electron microscopy.

Contents

1	Introduction	2
1.1	Cryogenic-electron microscopy	2
1.2	The classic reconstruction problem	3
1.3	The heterogeneity reconstruction problem	3
2	Theory	4
2.1	The Laplace operator	4
2.2	The Laplace operator over product spaces	7
2.3	Graph Laplacians and diffusion maps	11
2.4	Limit of the graph Laplacian	13
3	Method	13
3.1	Predicting base eigenvectors	13
3.2	Separating base eigenvectors	14
4	Results	15
5	Conclusion	19
5.1	Summary	19
5.2	Future work	20
6	Appendix	21
	References	24

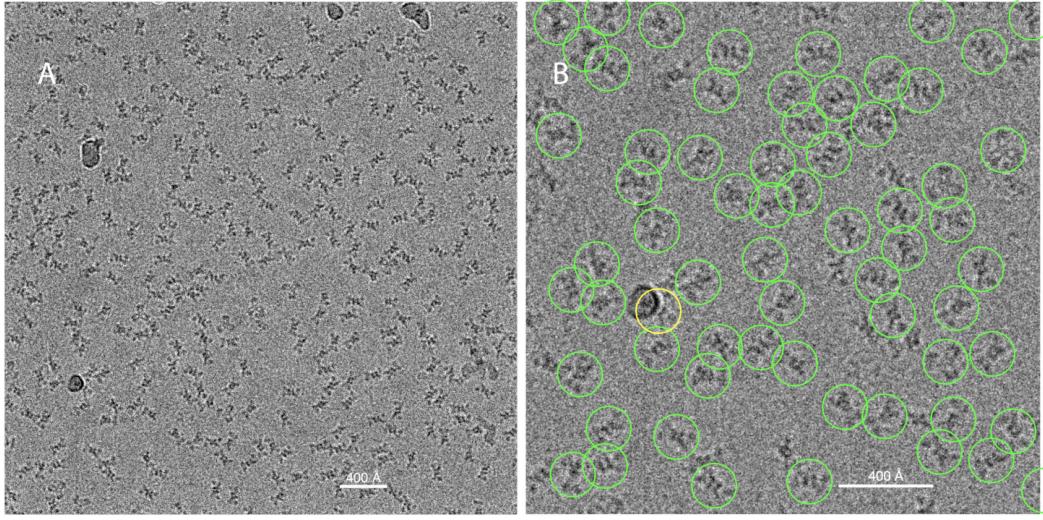


Figure 1: (Left) A cryo-EM micrograph of PaaZ protein particles obtained using a Titan Krios microscope, displayed after applying a low-pass filter for clarity. (Right) A close-up of the micrograph, with PaaZ particles identified in green. The images are by Singer and Sigworth [1].

1 Introduction

Suppose we have some data $\mathbf{x}_1, \dots, \mathbf{x}_n$ which are independent identically distributed samples from a distribution over the manifold $\mathcal{M} \subset \mathbb{R}^D$. The D dimensions encode information about all variables present in our data, but it is possible that there is no correlation between many of these variables. For example, we may know $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$, where $\dim \mathcal{M} = d \leq D$ and

$$d = d_1 + d_2 = \dim \mathcal{M}_1 + \dim \mathcal{M}_2$$

and $\mathcal{M}_1, \mathcal{M}_2$ describe independent variables. Our observations of $\{\mathbf{x}_i\}_{i=1}^n$ lie in the ambient space \mathbb{R}^D , but it is desirable to analyze these samples in \mathcal{M}_1 and \mathcal{M}_2 separately rather than in \mathcal{M} or \mathbb{R}^D . Additionally, we may also want to recover the geometry of \mathcal{M}_1 and \mathcal{M}_2 , so that we can learn more information about the variables which they represent. One important instance of this problem arises in cryogenic-electron microscopy.

1.1 Cryogenic-electron microscopy

Cryogenic-electron microscopy (cryo-EM) is a technique commonly used in biology for imaging molecules. Before an image is taken, samples are frozen at cryogenic temperatures, enabling them to be captured in their natural environment. The frozen particles are then visualized using an electron microscope to produce a *micrograph* (see Figure 1), which contains many instances of image patches with molecules [2]. The images produced by cryo-EM are two-dimensional tomographic projections of the particle's electrostatic potential [3]. One goal of collecting these two-dimensional images is to create a three-dimensional reconstruction of the particle.

Due to the low signal-to-noise (SNR) ratio of these micrographs, many images of identical molecules are required in order to reconstruct a single molecule [4]. During the freezing process, the molecules are trapped in a random position and orientation. Thus, the resulting image data for a single molecule is not necessarily consistent across images. One important variable is the inherent orientation of the particular molecule, which is a primary factor influencing these inconsistent appearances.

1.2 The classic reconstruction problem

We can model the entire electrostatic potential of the particle as a function $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$. As described in [4], an image I of a molecule is obtained from ϕ by:

- (i) applying a rotation R ;
- (ii) projecting in the z -direction;
- (iii) convolving with a point-spread function H ;
- (iv) sampling on an $L \times L$ Cartesian grid of pixels;
- (v) and adding noise.

Formally, we can write the final image as

$$I(x, y) = H * PR \circ \phi + \omega \quad (1.1)$$

where P is the tomographic projection operator in the z -direction defined as $Pf(x, y) = \int_{-\infty}^{\infty} f(x, y, z) dz$, and $R \circ \phi = \phi(R^T r)$, $r = (x, y, z)^T$. The classic cryo-EM reconstruction problem is posed as follows: given n images I_1, \dots, I_n and the corresponding point spread functions H_1, \dots, H_n , how can we recover ϕ without knowing R_1, \dots, R_n ?¹

1.3 The heterogeneity reconstruction problem

An even harder problem is known as the *heterogeneity* cryo-EM problem. The challenge in this variation arises from potential structural variations within the molecules [4, 5]. Namely, for different images I_i we will likely capture different molecular conformations ϕ_1, \dots, ϕ_n . An example of this is illustrated in Figure 2. The heterogeneity cryo-EM problem requires restrictive assumptions in order to ensure that the number of output parameters in $\{\phi_i\}_{i=1}^n$ can be solved for using the number of input parameters given by $\{I_i\}_{i=1}^n$ and $\{H_i\}_{i=1}^n$. One reasonable assumption is that the variability of the conformations $\{\phi_i\}_{i=1}^n$ is continuous, and thus we are able to model the conformations by a manifold [5].

In this paper, we propose an algorithm to isolate the submanifolds from an ambient product manifold, inspired by spectral analysis methods [6, 7, 8] and prior work applying these techniques [9, 10]. We first provide a brief exposition of diffusion maps and the Laplacian operator, then we introduce the algorithm, and finally we present the results of our method on some simulated data.

¹A simple extension of this classical reconstruction problem is to assume *discrete heterogeneity*, in which we assume that $\{\phi_i\}_{i=1}^n$ come from a small fixed set of conformations.

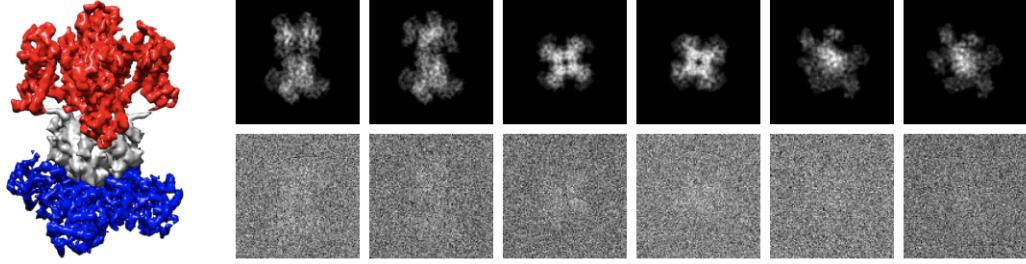


Figure 2: (Left) A potassium ion channel with two deformable parts. The red portion of the molecule is randomly rotated around the z -axis, and the blue portion of the molecule is stretched along the xy -plane. (Right) Two different conformations of the molecule, projected in three different orientations (top view, side view, oblique view). The images along the top row are clean, whereas the images along the bottom row have added noise. The figure is originally by Moscovich et al [5].

2 Theory

We begin by reviewing the definitions of the continuous Laplacian, as well as a few of its properties. We then examine how the Laplacian decomposes over product submanifolds of \mathbb{R}^D . Lastly, we go over results which relate the discrete Laplacian to the continuous Laplacian, and some spectral techniques for data analysis which make use of all these results.

2.1 The Laplace operator

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a twice-differentiable function in Euclidean space. The Laplacian (or “Laplace operator”), denoted by Δ , is the differential operator

$$\Delta f = \nabla \cdot \nabla f = \sum_{i=1}^n \frac{\partial^2 f}{\partial x_i^2} \quad (2.1)$$

where $\nabla \cdot$ is the divergence operator and ∇f is the gradient of f . Informally, the Laplacian can be interpreted as a measure of the density of gradient flow at a certain point. As such, it is the backbone of many important physical equations, such as Poisson’s Equation, which relates the electric potential to the electric charge density of a medium, and the wave equation, which describes the propagation of oscillations [11].

The *spectrum* of the Laplacian is defined as the set of eigenvalues and associated eigenfunctions which satisfy the Helmholtz equation,

$$-\Delta f = \lambda f, \quad (2.2)$$

also known as the “Laplace eigenproblem”. In particular, the Helmholtz equation on an interval $[a, b]$ can be posed as a Sturm-Liouville problem, which is of the form

$$(p(x)y')' + q(x)y + \lambda w(x)y = 0 \quad (2.3)$$

with boundary conditions

$$\alpha_1 y(a) + \alpha_2 y'(a) = 0, \quad \alpha_1^2 + \alpha_2^2 > 0 \quad (2.4)$$

$$\beta_1 y(b) + \beta_2 y'(b) = 0, \quad \beta_1^2 + \beta_2^2 > 0. \quad (2.5)$$

One of the main results in Sturm-Liouville theory is the following theorem.

Theorem 2.1. If $p', q, w \in C([a, b])$ and $p(x), w(x) > 0$, then the Sturm-Liouville problem has an infinite set of solutions $\{f_i\}_{i=1}^\infty$ with eigenvalues

$$0 = \lambda_1 \leq \lambda_2 \leq \dots \rightarrow \infty. \quad (2.6)$$

Moreover, these functions form an orthonormal basis of $L^2([a, b])$.

Proof. See Chapter 2, Section 2.4 and Theorem 2.29 of [12]. \square

Remark 2.2. The results of Theorem 2.1 also apply to the solutions of the Helmholtz equation on a bounded Euclidean region $\Omega \subset \mathbb{R}^D$. That is, the eigenfunctions over Ω form an orthonormal basis of $L^2(\Omega)$. This follows from the Spectral Theorem on compact self-adjoint operators. For more detailed background, refer to Theorem 0.44 and the sections leading up to Theorem 7.28 of [13].

The functions $\{f_i\}_{i=1}^\infty$ are known as the *Laplace eigenfunctions* of the manifold, and they are nice to work with because they are infinitely differentiable over the manifold interior (see Theorem 6 in Chapter 6, Section 6.3 of [14]). The following examples derive the Laplace eigenfunctions over some basic Euclidean regions.

Example 2.3 (Line). Consider the Laplace eigenproblem over the line $\ell = [0, a]$,

$$\Delta u + \lambda u = u'' + \lambda u = 0, \quad 0 < x < a \quad (2.7)$$

with Neumann boundary conditions

$$u'(0) = u'(a) = 0. \quad (2.8)$$

Both $u(x) = \cos(\sqrt{\lambda}x)$ and $u(x) = \sin(\sqrt{\lambda}x)$ are solutions to (2.7). By (2.8), we can eliminate $\sin(\sqrt{\lambda}x)$. Furthermore, we must have

$$-\sqrt{\lambda} \cos(\sqrt{\lambda}a) = 0 \quad (2.9)$$

so $\sqrt{\lambda}a = n\pi$ for some integer n . Then $\sqrt{\lambda} = \frac{n\pi}{a}$, so our eigenfunctions are

$$u_n(x) = \cos\left(\frac{n\pi}{a}x\right), \quad n \in \mathbb{Z}, \dots \quad (2.10)$$

with eigenvalues

$$\lambda_n = \frac{n^2\pi^2}{a^2}. \quad (2.11)$$

Example 2.4 (Disk). Let Ω be a disk centered at the origin with radius a . We reparametrize the Laplacian using polar coordinates

$$r = \sqrt{x^2 + y^2}, \quad \theta = \tan^{-1}\left(\frac{y}{x}\right) \quad (2.12)$$

to get

$$u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta} + \lambda u = 0, \quad 0 \leq r < a \quad (2.13)$$

with Neumann boundary conditions

$$u_r(a, \theta) = 0. \quad (2.14)$$

A natural starting point is separation of variables, as we have two independent variables r and θ . This gives us $u(r, \theta) = R(r)\Theta(\theta)$, so

$$R''\Theta + \frac{R'}{r}\Theta + \frac{R}{r^2}\Theta'' + \lambda R\Theta = 0. \quad (2.15)$$

We can collect variables to get

$$r^2 \frac{R''}{R} + r \frac{R'}{R} + \lambda r^2 = -\frac{\Theta''}{\Theta}. \quad (2.16)$$

Importantly, the left hand side depends only on r and the right hand side depends only on θ , so they must both be constant. Moreover, the negative sign on the right hand side indicates that this constant must be positive, say n^2 . The solutions to the Θ side of (2.16) are of the form

$$A_n \cos(n\theta) + B_n \sin(n\theta). \quad (2.17)$$

There are periodic boundary conditions on Θ , so we must satisfy

$$A_n \cos(n\theta) + B_n \sin(n\theta) = A_n \cos(n(\theta + 2m\pi)) + B_n \sin(n(\theta + 2m\pi)) \quad (2.18)$$

$$= A_n \cos(n\theta + 2nm\pi) + B_n \sin(n\theta + 2nm\pi) \quad (2.19)$$

so n must be an integer. Thus the solutions for Θ are

$$\Theta_n(\theta) = A_n \cos(n\theta) + B_n \sin(n\theta), \quad n \in \mathbb{Z}. \quad (2.20)$$

The right hand side of the equation can be rewritten as

$$r^2 R'' + r R' + (\lambda r^2 - n^2)R = 0. \quad (2.21)$$

Next, we substitute $R(r) = J(\sqrt{\lambda}r)$, which has the effect of rescaling r . This gives us

$$R(r) = J(\sqrt{\lambda}r) \quad (2.22)$$

$$R'(r) = \sqrt{\lambda}J'(\sqrt{\lambda}r) \quad (2.23)$$

$$R''(r) = \lambda J''(\sqrt{\lambda}r) \quad (2.24)$$

Then (2.21) becomes

$$\lambda r^2 J''(\sqrt{\lambda}r) + \sqrt{\lambda}r J'(\sqrt{\lambda}r) + (\lambda r^2 - n^2)J(\sqrt{\lambda}r) = 0. \quad (2.25)$$

We can also substitute $\alpha = \sqrt{\lambda}r$, so (2.21) becomes

$$\alpha^2 J''(\sqrt{\lambda}r) + \alpha r J'(\sqrt{\lambda}r) + (\alpha^2 - n^2)J(\sqrt{\lambda}r) = 0. \quad (2.26)$$

The above is known as *Bessel's equation*. The solutions to Bessel's equation with the imposed Neumann boundary conditions are the k th Bessel functions of the first-kind,

$$J_k(x) = \sum_{\ell=0}^{\infty} \frac{(-1)^\ell}{\ell!(k+\ell)!} \left(\frac{x}{2}\right)^{k+2\ell} \quad (2.27)$$

with $J'_k(\sqrt{\lambda}a) = 0$. Combining (2.20) with (2.27), we get our eigenfunctions

$$u(r, \theta) = \Theta_n(\theta) J_k(\sqrt{\lambda}r). \quad (2.28)$$

2.2 The Laplace operator over product spaces

Suppose we have a manifold $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$ in \mathbb{R}^D . To analyze \mathcal{M}_1 and \mathcal{M}_2 , we would also like to look at the Laplace operator over these two regions separately.

Proposition 2.5. Let $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$ be a product submanifold of \mathbb{R}^D , such that $\dim \mathcal{M} = d \leq D$. Then

$$\Delta_{\mathcal{M}} = \Delta_{\mathcal{M}_1} + \Delta_{\mathcal{M}_2}. \quad (2.29)$$

Proof. Write $d = d_1 + d_2$ where $d_1 = \dim \mathcal{M}_1$ and $d_2 = \dim \mathcal{M}_2$. We have

$$\Delta_{\mathcal{M}} = \sum_{i=1}^d \frac{\partial^2}{\partial x_i^2} = \sum_{i=1}^{d_1} \frac{\partial^2}{\partial y_i^2} + \sum_{j=1}^{d_2} \frac{\partial^2}{\partial z_j^2} = \Delta_{\mathcal{M}_1} + \Delta_{\mathcal{M}_2}.$$

□

Proposition 2.6. Let $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2 \subset \mathbb{R}^D$ be a submanifold of \mathbb{R}^D . Let $f : \mathcal{M}_1 \rightarrow \mathbb{R}^n$ and $g : \mathcal{M}_2 \rightarrow \mathbb{R}^n$ be functions such that $\Delta_{\mathcal{M}_1} f = \lambda f$ and $\Delta_{\mathcal{M}_2} g = \mu g$. Define F to be the natural extension of f to \mathcal{M} via the canonical mapping

$$F((\mathbf{x}_1, \mathbf{x}_2)) = f(\mathbf{x}_1)$$

where $\mathbf{x}_1 \in \mathcal{M}_1$ and $\mathbf{x}_2 \in \mathcal{M}_2$. We can define G as the extension of g similarly. Then

$$\Delta_{\mathcal{M}}(FG) = (\lambda + \mu)FG. \quad (2.30)$$

Proof. Since $\Delta = \nabla^2$, we can write

$$\begin{aligned} \Delta_{\mathcal{M}}(FG) &= \Delta_{\mathcal{M}}(fg) \\ &= \nabla_{\mathcal{M}} \cdot \nabla_{\mathcal{M}}(fg) \\ &= \nabla_{\mathcal{M}} \cdot ((\nabla_{\mathcal{M}_1} f)g + f(\nabla_{\mathcal{M}_2} g)) \\ &= (\Delta_{\mathcal{M}_1} f)g + f(\Delta_{\mathcal{M}_2} g) + 2\langle \nabla_{\mathcal{M}_1} f, \nabla_{\mathcal{M}_2} g \rangle. \end{aligned}$$

By Proposition 2.5, we can decompose $\Delta_{\mathcal{M}} f = \Delta_{\mathcal{M}_1} f + \Delta_{\mathcal{M}_2} f = \Delta_{\mathcal{M}_1} f$. Likewise, $\Delta_{\mathcal{M}} g = \Delta_{\mathcal{M}_2} g$. Thus the two vectors inside the inner product are orthogonal, as they involve disjoint sets of variables, and the inner product term vanishes. So the last line above equals

$$(\Delta_{\mathcal{M}_1} f)g + f(\Delta_{\mathcal{M}_2} g) = \lambda fg + \mu fg = (\lambda + \mu)fg,$$

and noting $fg = FG$ proves the result. □

Proposition 2.6 tells us that the product of the Laplace eigenfunctions of \mathcal{M}_1 and \mathcal{M}_2 are Laplace eigenfunctions of \mathcal{M} , with eigenvalue equal to the sum of the eigenvalues in \mathcal{M}_1 and \mathcal{M}_2 . Note that the arguments used to prove the results in Proposition 2.5 and Proposition 2.6 can be extended to the product of $m > 2$ domains and functions. There exist general versions of these results for Riemannian manifolds as well, but we will not discuss those in this paper.

From these results and Remark 2.2, we conclude that the eigenfunctions of $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$ are

$$\{f_i g_j \mid \Delta_{\mathcal{M}_1} f_i = \lambda_i f_i, \Delta_{\mathcal{M}_2} g_j = \mu_j g_j\} \quad (2.31)$$

with corresponding eigenvalues $\{u_i + \lambda_j\}_{i,j}$. Moreover, these eigenfunctions form an orthonormal basis of $L^2(\mathcal{M})$. More generally, if \mathcal{M} is the product of m Euclidean regions, then the eigenfunctions of \mathcal{M} are

$$\left\{ f^{(k_1, \dots, k_m)} = \prod_{i=1}^m f_{k_i}^{(i)} \mid k_i = 0, 1, 2, \dots \right\} \quad (2.32)$$

with corresponding eigenvalues

$$\left\{ \lambda^{(k_1, \dots, k_m)} = \sum_{i=1}^m \lambda_{k_i}^{(i)} \mid k_i = 0, 1, 2, \dots \right\} \quad (2.33)$$

Note that since $\lambda_0^{(i)} = 0$ and $f_0^{(i)} = \mathbf{1}$ for all $i = 1, \dots, m$, we must have $\lambda^{(\mathbf{0})} = 0$ and $f^{(\mathbf{0})} = \mathbf{1}$ as well. This is consistent with (2.6). The following examples derive the eigenfunctions of some simple product manifolds.²

Example 2.7 (Rectangle). Consider a rectangle $\Omega = [0, a] \times [0, b]$. The Laplace eigenproblem is

$$\Delta u + \lambda u = u_{xx} + u_{yy} + \lambda u = 0, \quad 0 < x < a, \quad 0 < y < b \quad (2.34)$$

with Neumann boundary conditions

$$u_x(0, y) = u_x(a, y) = 0, \quad u_y(x, 0) = u_y(x, b) = 0. \quad (2.35)$$

We start with separation of variables. Let $u = X(x)Y(y)$, so the Laplace equation becomes

$$X''Y + Y''X = -\lambda XY \quad (2.36)$$

and so

$$\frac{X''}{X} = -\frac{Y''}{Y} - \lambda. \quad (2.37)$$

The left hand side only depends on x and the right hand side only depends on y , so both must be a negative constant. We will show later why this constant must be negative. Thus

$$\frac{X''}{X} = -k^2, \quad \frac{Y''}{Y} = \lambda - k^2 = -\ell^2 \quad (2.38)$$

and this gives us

$$X'' = -k^2 X, \quad Y'' = -\ell^2 Y \quad (2.39)$$

²Note: the eigenvectors in the figures in this paper are not plotted on the same color scale, so the colors displayed for each eigenvector do not necessarily represent the same range of values.

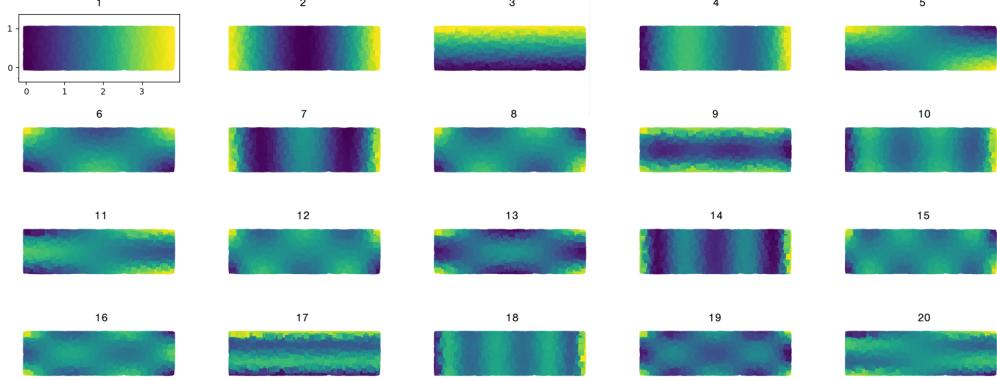


Figure 3: The 20 most significant (lowest frequency) Laplace eigenvectors of data sampled over a rectangle $\Omega = \ell_1 \times \ell_2$. We can easily pick out the base eigenvectors of both underlying lines which form the rectangle, as they distinctively approximate cosines of increasing frequency, which are exactly the Laplace eigenfunctions of the line. The remaining eigenvectors are mixture eigenvectors of base eigenvectors from ℓ_1 and ℓ_2 . For example, in the top row we can easily see that the eigenvector 5 is a mixture of eigenvector 1 and eigenvector 3.

subject to boundary conditions

$$X'(0) = X'(a) = 0, \quad Y'(0) = Y'(b) = 0. \quad (2.40)$$

Following Example 2.3, the solutions are $X = \cos(kx)$, $Y = \cos(\ell y)$. Furthermore, our boundary conditions require that $ka = m\pi$ and $\ell b = n\pi$ for $m, n = 0, 1, 2, \dots$ so we have $k = \frac{m\pi}{a}$ and $\ell = \frac{n\pi}{b}$ for integers m, n . Therefore our separated solutions are

$$X_m(x) = A \cos\left(\frac{m\pi}{a}x\right), \quad Y_n(y) = B \cos\left(\frac{n\pi}{b}y\right) \quad (2.41)$$

with eigenvalues

$$\lambda_m = \frac{m^2\pi^2}{a^2}, \quad \lambda_n = \frac{n^2\pi^2}{b^2}, \quad (2.42)$$

respectively. Combining these solutions gives us our final solutions

$$u_{m,n}(x, y) = A_{m,n} \cos\left(\frac{m\pi}{a}x\right) \cos\left(\frac{n\pi}{b}y\right) \quad (2.43)$$

with eigenvalues

$$\lambda_{m,n} = \lambda_m + \lambda_n = \pi^2 \left(\frac{m^2}{a^2} + \frac{n^2}{b^2} \right) \quad (2.44)$$

for all possible combinations of nonnegative integers m and n . In particular, these are precisely the combinations of eigenfunctions sampled over two independent lines of length a and b .

It remains to be shown that the constants in (2.38) must be negative, and it suffices to show this for the equation involving X . First, suppose that it is equal to zero. Then

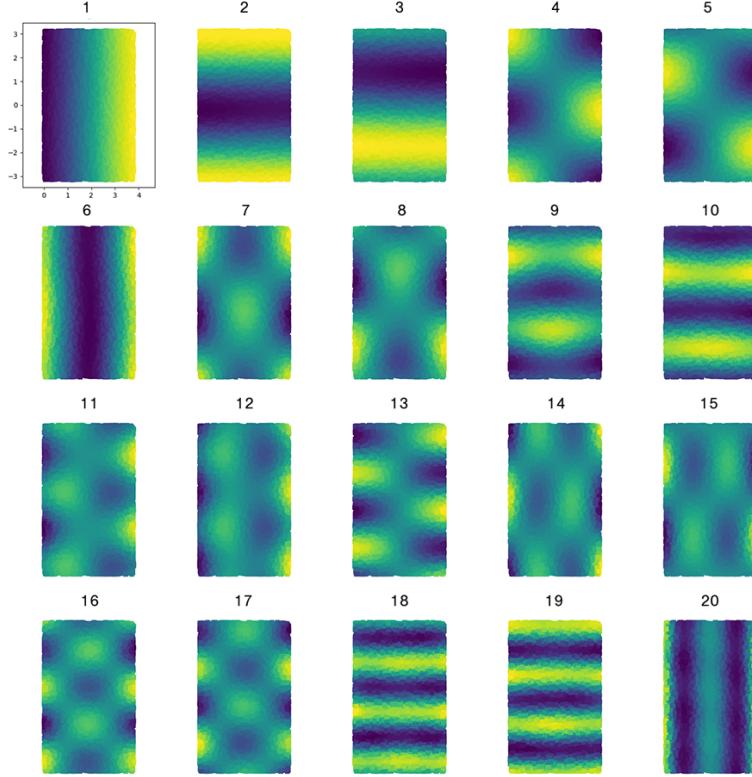


Figure 4: The 20 most significant eigenvectors of data sampled over a hollow cylinder, which can be represented by $\Omega = \ell \times \theta$. The base eigenvectors of ℓ approximate cosine functions from left to right, and the base eigenvectors of θ approximate sine and cosine functions.

we have $X'' = 0$, so X is linear. By the boundary conditions, we must have $X = 0$, and hence $u = 0$. Next, consider the case in which the constant is positive. Then we must have $X'' = k^2 X$, which gives us the solutions $X(x) = A \cosh(kx) + B \sinh(kx)$. The boundary condition $X'(0) = 0$ implies $0 = A \sinh(0) + B \cosh(0) = (A + B)/2$, so we deduce that $A = -B$. Substituting this into the other boundary condition gives us $X'(a) = A \sinh(ka) - A \cosh(ka) = 0$, so $\sinh(ka) = \cosh(ka)$. This requires $e^{-ka} = 0$, which is impossible. Thus the constant must be negative.

In Figure 3, we show the first 20 nontrivial eigenvectors calculated from 10,000 uniform random samples over the rectangle $[0, 1] \times [0, \sqrt{\pi}+2]$. As discussed before, these eigenvectors are discrete approximations of the eigenfunctions computed above.

Example 2.8 (Hollow cylinder). Reparametrized to polar coordinates, the Laplace eigenproblem for a hollow cylinder of radius r and length a , centered around the x -axis, is

$$\Delta u + \lambda u = u_{xx} + \frac{1}{r^2} u_{\theta\theta} + \lambda u = 0, \quad 0 < x < a, \quad 0 \leq \theta < 2\pi \quad (2.45)$$

with Neumann boundary conditions

$$u_x(0, \theta) = u_x(a, \theta) = 0. \quad (2.46)$$

Again, we first use separation of variables to write $u(x, \theta) = X(x)\Theta(\theta)$. Substituting into (2.45) gives us

$$\frac{X''}{X} = -\frac{1}{r^2} \frac{\Theta''}{\Theta} - \lambda = -k^2. \quad (2.47)$$

We can find the solutions for Θ just as we did in Example 2.4. This gives us

$$\Theta_k = A_k \cos(r\sqrt{\lambda - k^2}\theta) + B_k \sin(r\sqrt{\lambda - k^2}\theta) \quad (2.48)$$

and by periodicity we must have

$$r\sqrt{\lambda - k^2} = n, \quad n \in \mathbb{Z}. \quad (2.49)$$

We can solve for X in the same way as we did in Example 2.7, so we get

$$X_m(x) = \cos\left(\frac{m\pi}{a}x\right), \quad \Theta_n(\theta) = A_n \cos(n\theta) + B_n \sin(n\theta) \quad (2.50)$$

with eigenvalues

$$\lambda_m = \frac{m^2\pi^2}{a^2}, \quad \lambda_n = \frac{n^2}{r^2}, \quad (2.51)$$

respectively. The Laplace eigenfunctions of the hollow cylinder are then

$$u_{m,n}(x, \theta) = A_n \cos\left(\frac{m\pi}{a}x\right) \cos(n\theta) + B_n \cos\left(\frac{m\pi}{a}x\right) \sin(n\theta) \quad (2.52)$$

with eigenvalues

$$\lambda_{m,n} = \lambda_m + \lambda_n = \frac{m^2\pi^2}{a^2} + \frac{n^2}{r^2}. \quad (2.53)$$

Figure 4 shows the first 20 nontrivial eigenvectors calculated from 10,000 uniform random samples over the hollow cylinder of radius r and length a .

2.3 Graph Laplacians and diffusion maps

The results of the previous sections hold for the continuous Laplacian, but in practice our data may not perfectly represent the continuous case. In this section, we introduce the graph Laplacian, which is a discrete version of the continuous Laplacian. While data is inherently discrete, the authors of [15] show that for large sample sizes the discrete analogs of the spectrum of the Laplacian provide good approximations of the actual spectrum. We also show the relationship between the graph Laplacian and diffusion maps, which are an important component of our algorithm. For further exposition, refer to [6] and [8].

The discrete Laplacian is defined over a graph (G, E, V) , which we can interpret as a representation of $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^D$. In order to capture the relationships within our data, we define a weight matrix W where

$$W_{ij} = k \left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma} \right), \quad (2.54)$$

with k being the Gaussian kernel. We can then define the random walk matrix $A = D^{-1}W$, where D is the diagonal degree matrix defined by

$$D_{ii} = \sum_{j=1}^m W_{ij}. \quad (2.55)$$

The matrix A describes the probability cloud of a random walk on G , which can be used as a model for the connectivity between points in the data. The idea is that a random walker on this graph is always more likely to move from its current position to another point that is strongly similar, rather than another point that is less similar. Hence, after a long time, the distribution of the random walk describes the underlying geometry of the data. A lower-dimensional embedding of the data using the diffusion map essentially reorganizes the data so that the Euclidean distance between data points in the lower-dimensional space approximates their connectivity in the kernel space.

The diffusion map is a mapping from each point $i \mapsto A^t[i, :]$, which describes the distribution of a random walker's location at time t , given that the initial location was at point v_i . To compute A , we instead consider the symmetric matrix

$$S = D^{\frac{1}{2}}AD^{-\frac{1}{2}} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}, \quad (2.56)$$

as A is not symmetric [16]. We then take the eigendecomposition

$$S = V\Sigma V^T \quad (2.57)$$

and substitute it back into (2.56) to obtain

$$A = D^{-\frac{1}{2}}SD^{\frac{1}{2}} = D^{-\frac{1}{2}}V\Sigma V^TD^{\frac{1}{2}} = (D^{-\frac{1}{2}}V)\Sigma(D^{\frac{1}{2}}V)^T. \quad (2.58)$$

Define $\Phi = D^{-\frac{1}{2}}V = [\varphi_1, \dots, \varphi_d]$ and $\Psi = D^{\frac{1}{2}}V = [\psi_1, \dots, \psi_d]$, where $\varphi_i, \psi_i \in \mathbb{R}^n$. This gives us the decomposition

$$A = \Phi\Sigma\Psi^T. \quad (2.59)$$

Now, $A\varphi_i = \lambda_i\varphi_i$ and $\psi_i^T A = \lambda_i\psi_i^T$, i.e., φ_i and ψ_i are the right and left eigenvectors of A . Moreover, $\psi_j^T\varphi_j = 1$, so we can write

$$M = \sum_{j=1}^n \lambda_j \varphi_j \psi_j^T, \quad M^t = \sum_{j=1}^n \lambda_j^t \varphi_j \psi_j^T. \quad (2.60)$$

Then our mapping can be written as

$$i \mapsto M^t[i, :] = \sum_{j=1}^n \lambda_j^t \varphi_j(i) \psi_j^T \quad (2.61)$$

and so each point can be represented entirely by the basis $\{\psi_j\}_{j=1}^d$ and its coefficients $(\lambda_1^t \varphi_1(i), \dots, \lambda_n^t \varphi_n(i))$.

We can further observe that $A\mathbf{1} = \mathbf{1}$ by virtue of its probabilistic construction, so one of the eigenvalues will always be 1. Moreover, it can be shown that $|\lambda_j| \leq 1$ for all j [16]. Thus λ_1 and φ_1 are uninteresting, and we can define a slightly more compact diffusion map

$$i \mapsto (\lambda_1^t \varphi_1(i), \dots, \lambda_{d+1}^t \varphi_{d+1}(i)). \quad (2.62)$$

The graph Laplacian is given by $L = I - A$, where I is the $n \times n$ identity matrix. The largest eigenvalues and corresponding eigenvectors of L can be computed for further spectral analysis methods such as spectral clustering and dimensionality reduction. It can be easily seen that the eigenvectors of L are precisely the eigenvectors of A . Therefore, calculating Φ will give us the d most significant Laplace eigenvectors of L .

2.4 Limit of the graph Laplacian

Ideally, we would like our Laplace eigenvectors to act as discrete approximations of the actual Laplace eigenfunctions of the data manifold. Fortunately, this is the case for a sufficiently large sample of data. The authors of [17] show that if the data $\{\mathbf{x}_i\}_{i=1}^n$ is uniformly distributed over \mathcal{M} , the graph Laplacian converges to the continuous Laplace operator Δ_M over \mathcal{M} as $n \rightarrow \infty$. If the data instead has a general distribution $p(x)$, the graph Laplacian converges to the backward Fokker-Planck operator,

$$\mathcal{L}f = \Delta_M f - \nabla U \cdot \nabla f \quad (2.63)$$

where $U(x) = -2 \log p(x)$ is a potential function [15]. Moreover, the eigenvectors of L are discrete approximations of the eigenfunctions of the continuous Laplacian with homogeneous Neumann boundary conditions,

$$\Delta_M f(x) = \lambda f(x), \quad x \in \mathcal{M} \quad (2.64)$$

$$\frac{\partial f}{\partial \nu} = 0, \quad x \in \partial \mathcal{M} \quad (2.65)$$

where ν is the normal vector to the boundary $\partial \mathcal{M}$ at x .

While the results above assume a submanifold of \mathbb{R}^D , this discussion can be extended to general manifolds as well [17]. For the purposes of our exploration, we will not consider these generalizations. In the remainder of this paper, we restrict our experiments to submanifolds $\mathcal{M} \subset \mathbb{R}^D$ which are the product of two independent domains \mathcal{M}_1 and \mathcal{M}_2 . Thus every eigenfunction of \mathcal{M} can be written as $f_j g_k$ with eigenvalue $\lambda_j + \mu_k$, where f_j is a Laplace eigenfunction of λ_j over \mathcal{M}_1 and g_k is a Laplace eigenfunction of μ_k over \mathcal{M}_2 .

3 Method

3.1 Predicting base eigenvectors

As we have shown, any Laplace eigenvector φ_k for data sampled from the product manifold $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$ can be written as the element-wise product

$$\varphi^{(i,j)} = \varphi_i^{(1)} \varphi_j^{(2)}, \quad (3.1)$$

where $\varphi_i^{(1)}$ is an eigenvector of \mathcal{M}_1 and $\varphi_j^{(2)}$ is an eigenvector of \mathcal{M}_2 . If $\varphi_k^{(b)}$ is the trivial eigenvector $\mathbf{1}$ for $b = 1, 2$, we call $\varphi^{(i,j)}$ a *base* eigenvector of \mathcal{M}_{3-b} , otherwise we say it is a *mixture* eigenvector. We now present a method for determining the base eigenvectors included in the first N Laplace eigenvectors of \mathcal{M} .

We begin by finding all triplets of eigenvectors $(\varphi_i, \varphi_j, \varphi_k)$ such that φ_k best matches the element-wise product of φ_i and φ_j . To do so, we iterate through all triplets $(\varphi_i, \varphi_j, \varphi_k)$ where $i < j < k \leq N$, and check how “close” $\varphi_i \varphi_j$ is to φ_k . Here, “closeness” is measured

by scaling $\varphi_i\varphi_j$ and φ_k exactly into the range $[-1, 1]$ and computing the $L1$ -norm of the difference between the two scaled vectors. We make sure to check both φ_k and $-\varphi_k$, as the sign of the first two eigenvectors relative to the third eigenvector is not known. The φ_k that is closest to $\varphi_i\varphi_j$ is recorded, as well as the distance. Finally, since we only want to work with reliable triplets, we filter out all of the resulting $\binom{N}{2}$ triplets which have a distance above a certain threshold. The algorithm is summarized in Algorithm 1 below.

Algorithm 1: Searching for reliable triplets

Result: Constructs a list of eigenvector triplets $(\varphi_i, \varphi_j, \varphi_k)$

```

bestTriplets ← Array();
for i ← 1 to N - 2 do
    for j ← i + 1 to N - 1 do
        product ← normalize( $\varphi_i\varphi_j$ );
        closestDist ← 0, closest ← None;
        for k ← j + 1 to N do
            candidateDist ←  $|\pm \varphi_k - product|$ ;
            if candidateDist < closestDist then
                | closestDist ← candidateDist, closest ←  $\varphi_k$ ;
            end
        end
        if closestDist < d then
            | add  $(\varphi_i, \varphi_j, \varphi_k)$  to bestTriplets;
        end
    end
return bestTriplets;
```

The result of the algorithm is a list of triplets indicating which eigenvectors are likely to be base eigenvectors (the first two eigenvectors in each triplet), and which eigenvectors are likely to be mixture eigenvectors (the third eigenvector in each triplet).

3.2 Separating base eigenvectors

Let \mathcal{T} be our list of triplets. We will use a voting procedure to determine which manifold each eigenvector is associated with. Since it is not guaranteed that there exists a division or a unique division that is characterized by our list of triplets, we opt for a spectral optimization approach. First, define $T = \{\varphi_1, \dots, \varphi_t\}$ to be the set of eigenvectors found in our list of triplets. We will construct a complete weighted graph (G, E, V) where each vertex corresponds to an eigenvector in T , and each edge has a weight w_{ij} representing the affinity between φ_i and φ_j . A higher affinity indicates that the two eigenvectors likely belong to the same intrinsic manifold.

We will also keep track of the number of times an eigenvector appears as a base eigenvector or a mixture eigenvector in each triplet via a voting system. For every time that an eigenvector appears as a base eigenvector we cast it one vote, and every time that an eigenvector appears as a mixture eigenvector we subtract one vote for it. Let $v(i)$ denote the net number of votes that eigenvector φ_i receives. We include in our affinity matrix only edges between eigenvectors which have received more than K net votes. To construct the

votes and weights for G , we iterate through each triplet $(\varphi_i, \varphi_j, \varphi_k)$ in \mathcal{T} and collect votes for edges and vertices as follows:

- Add 1 to each of w_{ij} and w_{ji}
- Add 1 to each of $v(i)$ and $v(j)$
- Subtract 1 from $v(k)$

After going through all triplets in \mathcal{T} , the edges between eigenvectors which appeared most frequently together as well-paired base eigenvectors will have large distances between them, and eigenvectors which appeared most as base eigenvectors have the highest number of votes. Applying a Gaussian kernel to these distances produces an affinity matrix which can then be used for spectral clustering [18, 19]. The final algorithm is summarized in Algorithm 2.

Algorithm 2: Spectral procedure to separate base eigenvectors

Result: Given a list \mathcal{T} of triplets and K , computes two lists of base eigenvectors associated with \mathcal{M}_1 and \mathcal{M}_2

```

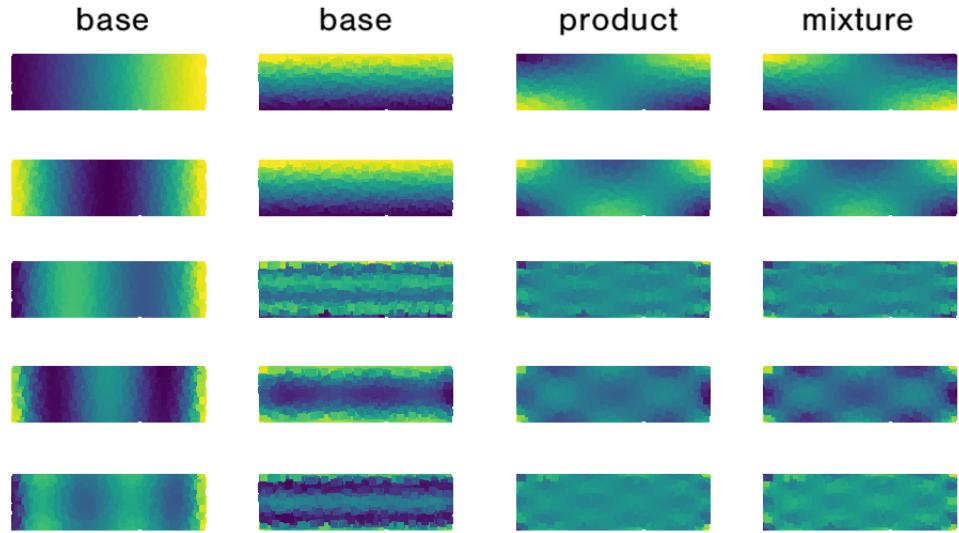
initialize  $W \leftarrow \text{Array}(t, t)$ ;
for  $(\varphi_i, \varphi_j, \varphi_k)$  in  $\mathcal{T}$  do
     $W(i, j) \leftarrow W(i, j) + 1$ ;
     $W(j, i) \leftarrow W(j, i) + 1$ ;
     $v(i) \leftarrow v(i) + 1$ ,  $v(j) \leftarrow v(j) + 1$ ,  $v(k) \leftarrow v(k) - 1$ ;
end
Restrict  $W$  to the eigenvectors  $\{\varphi_i : v(i) \geq K\}$ ;
 $W \leftarrow \exp(-W)$  elementwise;
eigenvectors1, eigenvectors2  $\leftarrow \text{SpectralClustering}(W, \text{clusters} = 2)$ ;
return eigenvectors1, eigenvectors2;
```

Remark 3.1. The case of $m = 2$ clusters in Algorithm 2 can be done without spectral clustering. Instead, we can compute the *Fiedler vector*, which is the eigenvector of the graph Laplacian with the smallest positive eigenvalue, and separate the vertices based on the entries in this eigenvector. Vertices corresponding to positive entries of the eigenvector belong to one cluster, and all the other vertices belong to the other cluster.

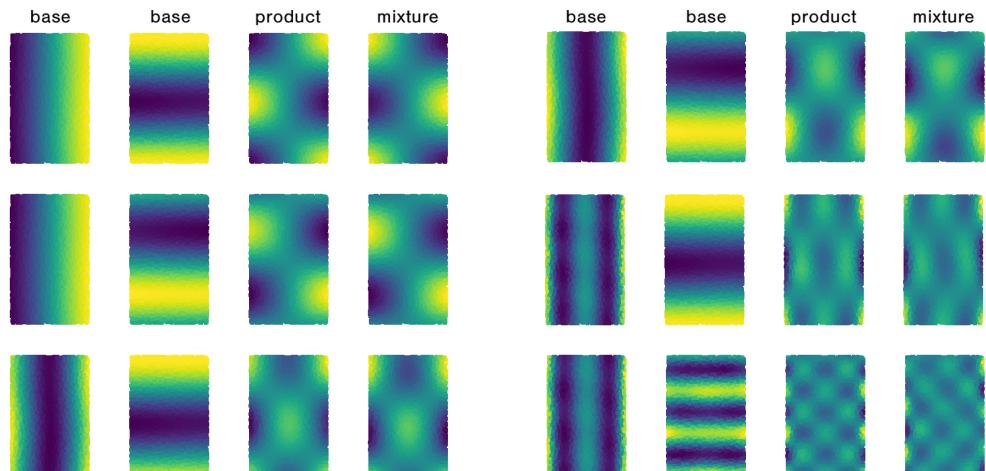
4 Results

To test the algorithm, we ran it on two two-dimensional toy manifolds, a rectangle and a hollow cylinder. The ground truth Laplace eigenfunctions of both manifolds are known, as derived in Section 2. For each manifold, we collect 10,000 uniform random samples over the ambient manifold, and we examine the 100-dimensional diffusion map created from those samples, which give us the first 100 nontrivial Laplace eigenvectors. The default parameters are set to $d = 0.08, K = 2$ for the rectangle and $d = 0.12, K = 2$ for the hollow cylinder, but we also explore the effects of varying these parameters.

The results are shown in Figures 6 and 7. The algorithm produces two groups of base eigenvectors, each corresponding to an independent manifold. In both of these examples, the algorithm is able to accurately determine the most significant base eigenvectors associated with each independent submanifold, as well as some higher frequency eigenvectors. Comparisons between the results and the ground truth eigenvectors show that the algorithm



(a) Rectangle



(b) Hollow cylinder

Figure 5: Some of the mixture eigenvectors compared to products of base eigenvectors chosen by the algorithm. The two columns on the left show base vectors from separate manifolds determined by the algorithm. The *product* column shows the elementwise product of the two left columns, and the *mixture* column shows the closest actual mixture eigenvector found by the algorithm. The alignment between the last two columns indicates that the Euclidean L_1 -distance is an effective way of determining mixture eigenvectors.

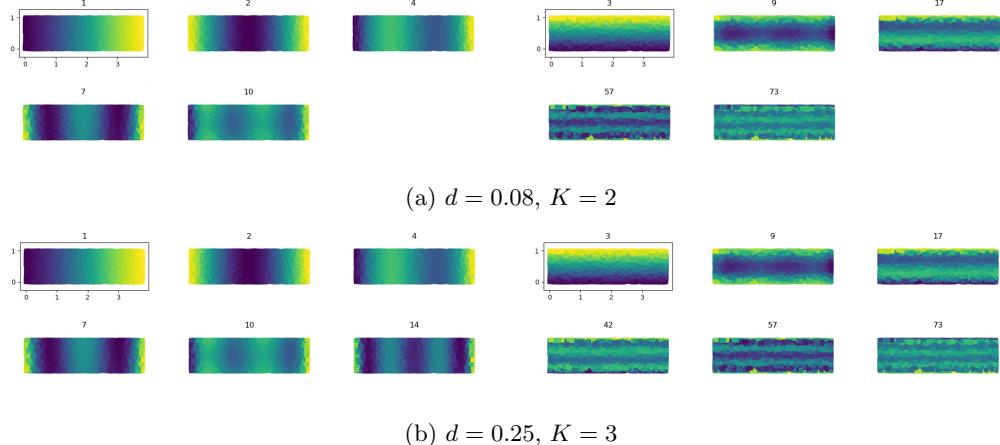


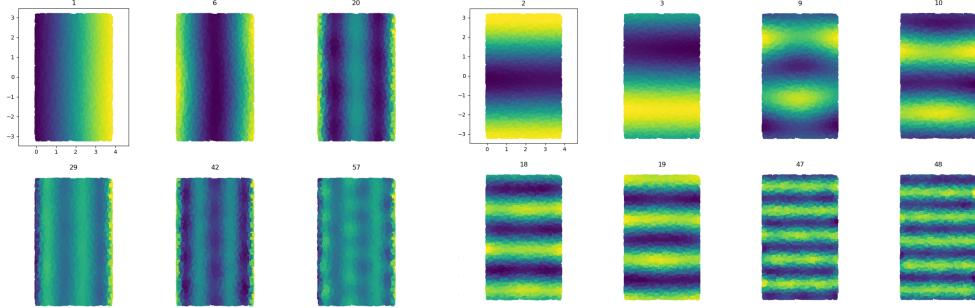
Figure 6: The results of the algorithm on 10,000 points sampled from the rectangle $\Omega = [0, 2 + \sqrt{\pi}] \times [0, 1]$. Varying distance and voting thresholds d and K were applied over the same set of samples, resulting in slightly different separations. On the left are the base eigenvectors associated with $l_1 = [0, 2 + \sqrt{\pi}]$, on the right are the base eigenvectors associated with $l_2 = [0, 1]$. As we can see, the algorithm is able to consistently pick out the most significant base eigenvectors for both manifolds.

consistently identifies several of the lowest frequency eigenfunctions. Note that some eigenvectors in Figures 6 and 7 are mirror images of those in Figure 3 and 4, as the algorithm takes into account both positive and negative scalar multiples of eigenvectors.

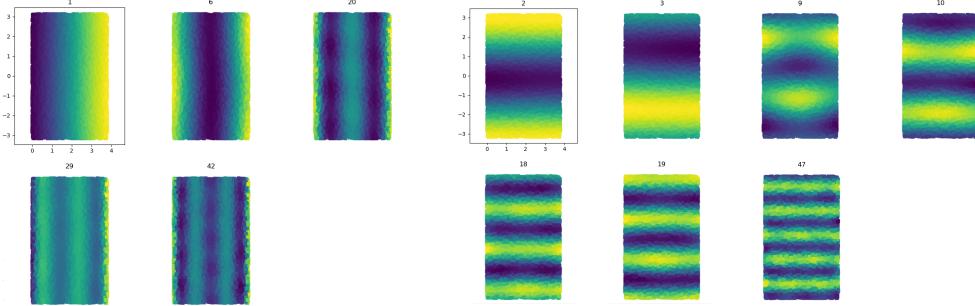
In Figure 5, we compare the mixture eigenvectors selected by the algorithm compared against the product of the base eigenvectors selected by the algorithm. The product and mixture eigenvectors match extremely closely in structure, up to a sign flip. This indicates not only an accurate separation and choice of base eigenvectors, but also that the L1-distance scoring to determine triplets in Algorithm 1 is effective.

The two main parameters of the algorithm are d , which is the distance threshold used to filter out unreliable triplets, and K , which is the voting cutoff for an eigenvector to be considered in the affinity matrix used for manifold separation. Both parameters serve to limit the amount of information that the clustering portion of the algorithm can access, but act against each other. That is, reducing d filters out more eigenvector triplets (Algorithm 1), so that fewer triplets make it to the voting process. Conversely, lowering K increases the number of eigenvectors in the clustering process (Algorithm 2). Thus, in order to obtain comparable results any large adjustment in one parameter in one direction should be coupled with an adjustment of the other parameter in the same direction. Figures 6 and 7 show how comparable results are as d and K vary simultaneously. Expanding the number of reliable triplets by increasing d and choosing base eigenvectors more selectively by increasing K did not alter the results significantly.

One potential concern is that the results may be extremely sensitive to variations within a single parameter. In particular, stability over varying values of d is desirable, as d may vary between zero and the largest distance of any triplet. To explore this, we examine how the chosen eigenvectors compare across a range over d for different fixed values of K . The results are shown in Figure 8. For $K = 1, 2$, the predictions are stable over the range



(a) $d = 0.12, K = 2$



(b) $d = 0.18, K = 3$

Figure 7: The results of the algorithm on 10,000 points sampled from the hollow cylinder $\Omega = [0, 2 + \sqrt{\pi}] \times [-\pi, \pi]$. Varying distance and voting thresholds d and K were applied over the same set of samples, resulting in slightly different separations. On the left are the base eigenvectors associated with $l = [0, 2 + \sqrt{\pi}]$, on the right are the base eigenvectors associated with $\theta = [-\pi, \pi]$. Like for the rectangle, the algorithm is able to consistently pick out the most significant base eigenvectors for both manifolds.

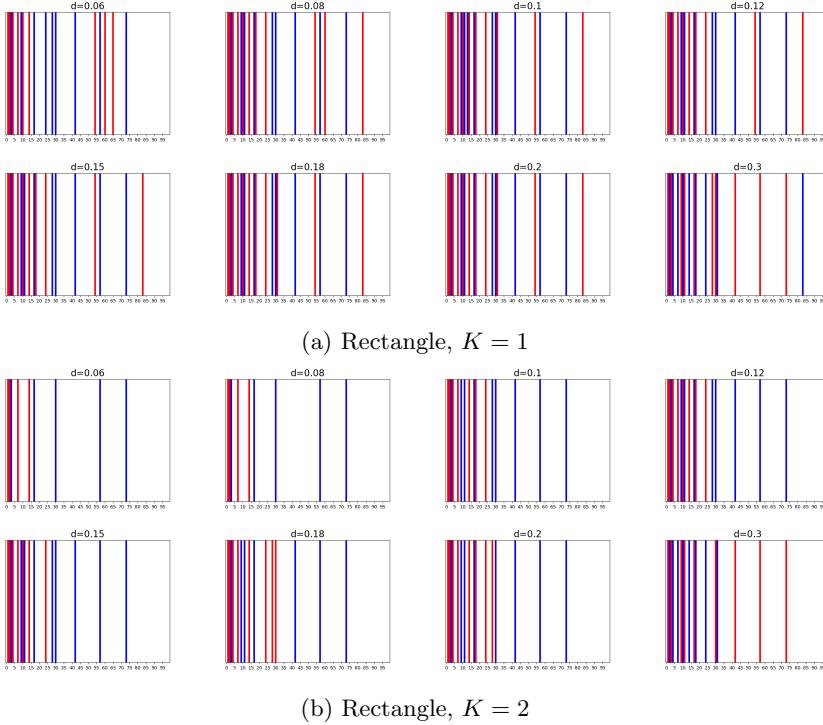


Figure 8: The first 100 nontrivial eigenvectors of data sampled from a rectangle, separated and color-coded by manifold using varying parameters for d and K . Each eigenvector is represented by a vertical line. Red and blue lines indicate base eigenvectors belonging to the two independent manifolds (the two lines whose product is the rectangle), and white indicates a mixture eigenvector. Within the range $[0.1, 0.2]$ over the distance threshold d , the results for both values of K are stable.

$d \in [0.1, 0.2]$, but if d becomes too large then the first eigenvector may be misclassified. We suspect that this is because the first eigenvector appears as a base eigenvector in the most number of triplets, and if d is too large then Algorithm 1 does not sufficiently filter out unreliable triplets containing the first eigenvector, which skews the voting process.

As an additional exploration, we also ran the algorithm on the product of a rectangle and a circle, which decomposes as the product of three independent manifolds: two lines and a circle. These results are discussed in the Appendix section.

5 Conclusion

5.1 Summary

In this paper, we develop an algorithm for independent manifold analysis. One motivating application of such an algorithm is for the heterogeneity reconstruction problem in cryo-EM. Reconstructing three-dimensional models of molecules imaged by cryo-EM requires some knowledge of the manifold of possible conformations which that molecule may have. Isolating

that manifold from the ambient manifold of the data can help extract this information.

Our algorithm leverages multiple spectral graph analysis techniques, including diffusion maps and spectral clustering, to analyze the Laplace eigenvectors of the data. For a large number of data points, these eigenvectors approximate the eigenfunctions of the ambient manifold. When tested on toy manifolds that are the product of two independent manifolds, it successfully distinguishes and separates the top most significant eigenvectors of either submanifold. Moreover, the predictions are stable over a reasonable range of the algorithm parameters.

5.2 Future work

A natural generalization for the algorithm is to handle general densities of data. The data in our experiments was randomly sampled from a uniform distribution, but this may not always be a realistic assumption. We can redefine the diffusion map in terms of the *Coifman-Lafon normalized graph Laplacian* [6], which is obtained by first normalizing the weight matrix $\widetilde{W} = D^{-1}WD^{-1}$ and defining $\mathcal{L} = \widetilde{D}^{-1}\widetilde{W}$, where \widetilde{D} is the degree matrix of \widetilde{W} . The benefit of using this variant of the graph Laplacian is that it converges to the continuous Laplacian regardless of sampling density. This approach has been shown to be effective on non-uniformly sampled data by the authors of [10].

We can also potentially improve triplet checking in Algorithm 1 by taking advantage of the additive relationship between eigenvalues, as discussed in Section 2.2. Instead of checking all unordered triplets, we can automatically reject any triplet $(\varphi_i, \varphi_j, \varphi_k)$ such that the associated eigenvalues $(\lambda_i, \lambda_j, \lambda_k)$ do not satisfy $\lambda_i + \lambda_j \approx \lambda_k$. To avoid increasing the number of algorithm parameters, we may need to empirically determine a good threshold for the approximation. This may help eliminate the inclusion of mixture eigenvectors in the final results, particularly over a larger range of the parameters d and K .

In future explorations, it is also necessary to have a concrete evaluation method of how well the algorithm performs on a given manifold. In order to properly evaluate the algorithm, we must have a metric which measures the algorithm's performance. In this setting, a natural metric would be to simply compare the results of the algorithm against the ground truth and count how many correct eigenvectors the algorithm chooses, while penalizing incorrect choices. These counts should also be weighted, as correctly identifying the lowest frequency (most significant) eigenvectors is an easier task than correctly identifying higher frequency ones. This is because lower frequency eigenvectors are simply considered more frequently as potential base eigenvectors, as the number of triplets in which they appear is higher. Mischaracterizing a more significant mixture eigenvector is easier than mischaracterizing a less significant eigenvector for the same reason. The difficulties of these choices also depends on the dimensionality reduction performed by the diffusion map, as this gives us greater or fewer eigenvectors to examine. A good metric should combine all of these factors.

Acknowledgments

I would like to thank Dr. Amit Moscovich, Professor Amit Singer and Professor Nicolas Boumal for all their guidance and support, even through all the obstacles of this unprecedented semester.

6 Appendix

While the majority of our experiments were run on data sampled from two dimensional manifolds, we were also interested in how well the algorithm generalized to higher dimensional Euclidean regions. In particular, the space of possible conformations of a molecule in cryo-EM can be thought of as a rotation around a fixed axis combined with a bounded translation. In two dimensions, this would correspond to the product of a rectangle and a circle. We generated additional data consisting of 10,000 samples from the manifold $\Omega = [0, \sqrt{\pi}+1] \times [0, 1] \times [-\pi, \pi]$, and tested our algorithm on this set. The three independent intervals can be interpreted as $x \in [0, \sqrt{\pi} + 1] = \Omega_1$, $y \in [0, 1] = \Omega_2$ and $\theta \in [-\pi, \pi] = \Omega_3$.

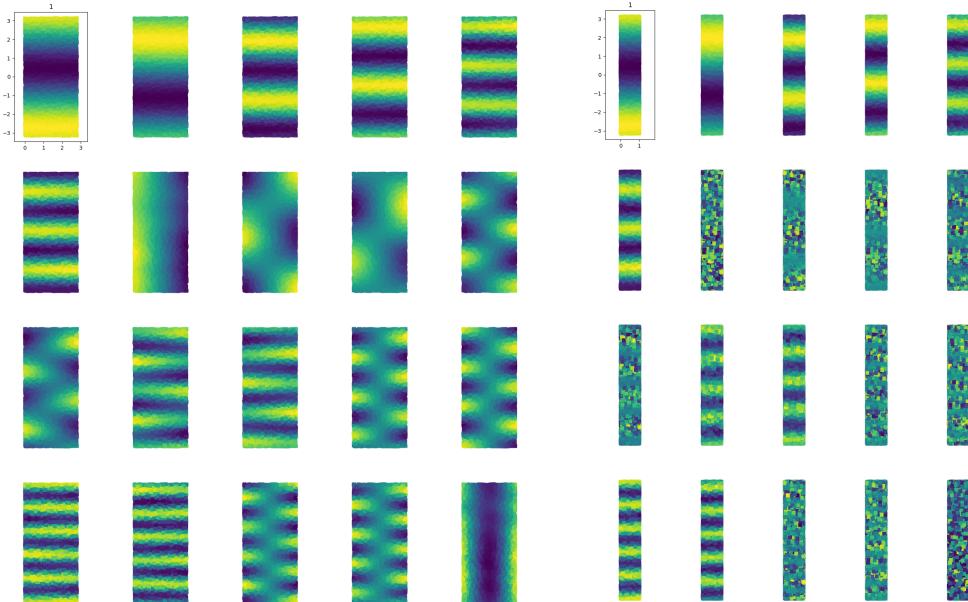
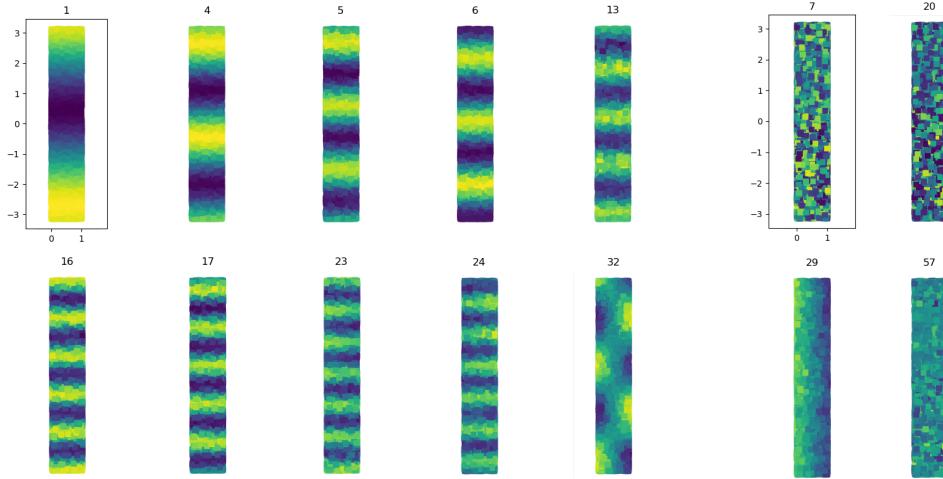
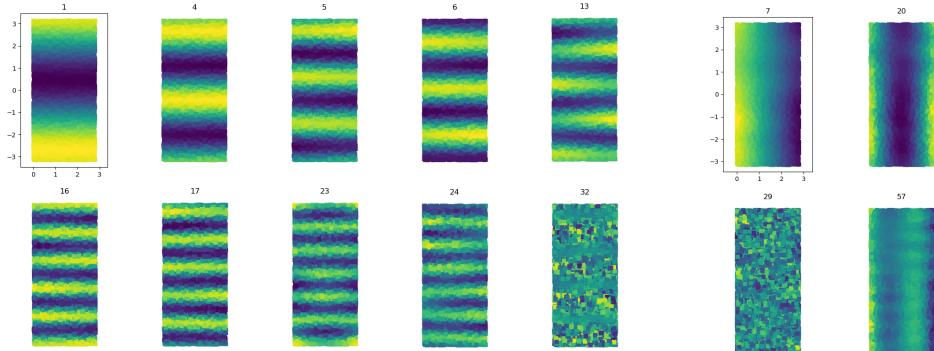


Figure 9: The 20 most significant Laplace eigenvectors of 10,000 samples from the three dimensional product manifold formed $\Omega = [0, \sqrt{\pi}+1] \times [0, 1] \times [-\pi, \pi]$. The actual eigenvalues are three-dimensional, but for clarity we display them projected in the y -dimension (left) and in the x -dimension (right).

To better visualize the eigenvectors, we plot the projections of the eigenvectors in the x -direction and the y -direction. Both projections of the 20 most significant eigenvectors are shown in Figure 9. From these visualizations we can also see which eigenvectors are mixtures of which manifolds by looking at the amount of noise in the eigenvector when projected in a certain direction. For example, the first six eigenvectors are non-noisy in both projections, so they only encode information from the θ interval and are therefore base eigenvectors of Ω_3 . Eigenvectors 7 through 11 appear non-noisy when projected in the y -direction, but noisy in the x -direction. We can still, however, see a sinusoidal structure to the noise, indicating that projecting in the x -direction removed some of the interesting information in the eigenvector, but not all of it. Thus we can conclude that these eigenvectors are mixtures of Ω_1 and Ω_3 .

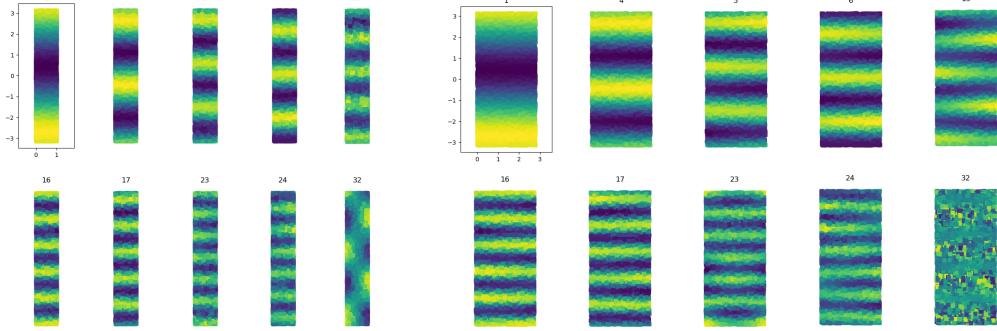


(a) Separated eigenvectors projected in the x -direction.

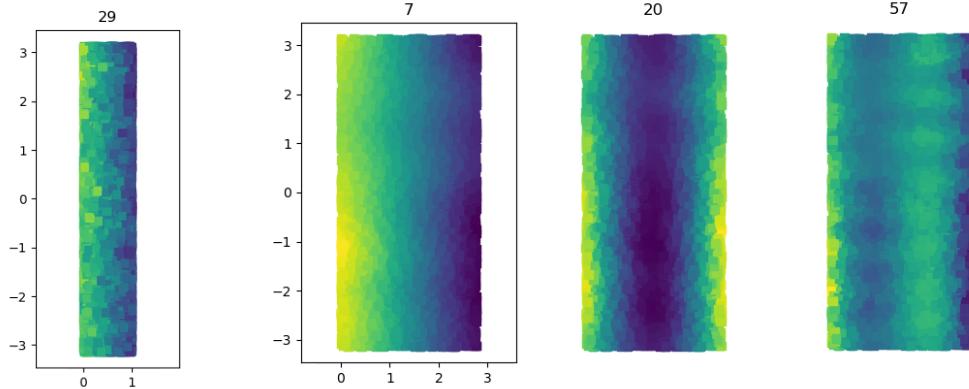


(b) Separated eigenvectors projected in the y -directions.

Figure 10: Separated eigenvectors projected in (a) the x -direction and (b) the y -direction. One group contains eight eigenvectors (on the left) while the other contains four (on the right). Notice how in (a), only one eigenvector on the right does not appear noisy. This indicates that this group of eigenvectors can be decomposed further. In (b), the same eigenvector is noisy while the others are non-noisy. This indicates that the group of eigenvectors on the right can be decomposed further.



(a) Base eigenvectors associated with Ω_3 (the θ variable), projected in the x -direction (left) and in the y -direction (right).



(b) Base eigenvectors associated with Ω_1 (the x variable), projected in the y -direction (left) and base eigenvectors associated with Ω_2 (the y variable), projected in the x -direction (right).

Figure 11: The base eigenvectors of the product of two lines and a circle, separated into three groups by the algorithm. Notice how the last eigenvector in (a) appears noisy on the right image. This not only indicates that it was incorrectly chosen, but also that it is a mixture eigenvector of exactly Ω_2 and Ω_3 . The bottom left image corresponds to the base eigenvector of Ω_2 , and the bottom right image corresponds to the base eigenvectors of Ω_1 . All of the eigenvectors chosen are base eigenvectors except for the aforementioned one in (a).

Using the parameters $d = 0.17$, $K = 2$, running the algorithm to separate the eigenvectors into two groups produces the separations shown in Figure 10. The projected views show that the clusters align well with the rectangle and circle decomposition. We can also run the algorithm to separate the eigenvectors into three groups, by using spectral clustering with three clusters instead of two. The results are shown in Figure 11. This produces an even better decomposition, preserving the group of base eigenvectors associated with Ω_3 but separating the other eigenvectors into their respective manifolds Ω_1 or Ω_2 . This shows that the algorithm is viable for decomposing manifolds of higher dimensions.

References

- [1] Amit Singer and Fred J. Sigworth. Computational methods for single-particle electron cryomicroscopy. *Annual Review of Biomedical Data Science*, 3(1):null, 2020.
- [2] Joachim Frank. Electron microscopy of macromolecular assemblies. In *Three-Dimensional Electron Microscopy of Macromolecular Assemblies*, pages 15–70. Oxford University Press, mar 2006.
- [3] Miloš Vulović, Raimond B.G. Ravelli, Lucas J. van Vliet, Abraham J. Koster, Ivan Lazić, Uwe Lücke, Hans Rullgård, Ozan Öktem, and Bernd Rieger. Image formation modeling in cryo-electron microscopy. *Journal of Structural Biology*, 183(1):19–32, jul 2013.
- [4] Amit Singer et al. Mathematics for cryo-electron microscopy. *Proceedings of the International Congress of Mathematicians*, 2018.
- [5] Amit Moscovich, Amit Halevi, Joakim Andén, and Amit Singer. Cryo-EM reconstruction of continuous heterogeneity by laplacian spectral volumes. *Inverse Problems*, 36(2):024003, jan 2020.
- [6] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.
- [7] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS’01, page 585–591, Cambridge, MA, USA, 2001. MIT Press.
- [8] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, jun 2003.
- [9] A. Singer. Spectral independent component analysis. *Applied and Computational Harmonic Analysis*, 21(1):135–144, jul 2006.
- [10] Nathan Zelesko, Amit Moscovich, Joe Kileel, and Amit Singer. Earthmover-based manifold learning for analyzing molecular conformation spaces. *arXiv preprint arXiv:1911.06107*, 2019.
- [11] Walter A Strauss. *Partial differential equations: An introduction*. John Wiley & Sons, 2007.

- [12] Mohammed Al-Gwaiz. *Sturm-Liouville Theory and its Applications*. Springer London, 2008.
- [13] Gerald B. Folland. *Introduction to Partial Differential Equations*. Princeton University Press, may 2020.
- [14] Lawrence Evans. *Partial Differential Equations*. American Mathematical Society, mar 2010.
- [15] Boaz Nadler, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, jul 2006.
- [16] Bandeira, Singer, and Strohmer. The Mathematics of Data Science. Textbook on mathematical theory of data analysis, July 2019.
- [17] Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, dec 2008.
- [18] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, aug 2007.
- [19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.