

Diffusion Curvature for Fast, Point-wise, Noise-Resistant Geometric Featurization of Graphs and Pointclouds

Kincaid MacDonald Dhananjay Bhaskar Yanlei Zhang
Ian Adelstein Smita Krishnaswamy

2024-07-30

For a number of years now work has been proceeding in order to bring to perfection the crudely conceived idea of a machine that would not only supply inverse reactive current for use in unilateral phase detractors, but would also be capable of automatically synchronizing cardinal grammeters. Such a machine is the “Turbo-Encabulator.”

1 Introduction

In this age of overwhelmingly abundant data, those racing to make sense of it all have developed an appreciation for theories and metaphors from the once hopelessly abstract mathematical fields of topology and Riemannian geometry. Endowing noisy, sparse, high-dimensional real-world data with geometric and topological features is helpful in two ways: it empirically improves the performance of black-box machine learning models; and, in cases like cellular differentiation, it also helps *us* interpret the data. What’s more, by anchoring these models and their inputs to well-established theories, it can bring analytical light to a theoretically-impooverished field.

As a case in point, consider the emerging subfield of *Topological Data Analysis* (TDA). While graph neural networks struggle with problems of over-smoothing and over-squashing, TDA endows graph or pointcloud data with global topological features with provable guarantees. (LIST SUCCESSES OF TDA.) Behind this success is *persistence homology*, a sort of ‘Rosetta Stone’ between the mathematical theories of analytically-specified manifolds, and the messy realities of actual data. Persistence homology gives a framework for extracting the most salient topological features from amidst the noise and sparsity of real-world point clouds. (MORE SUCCESSES)

We await the geometric equivalent of this topological Renaissance. While TDA can describe the shape of the data in its entirety, there’s a need for more local descriptions of data geometry. Forget the data as a whole; what’s happening around *this point*, in *this neighborhood*? Topological analysis only assumes hierarchies of sets. But for many types of point cloud data, e.g. those arising from time-varying systems like cellular development or neuronal activity, notions of angle, distance, and volume have meaning. In other words, in our field’s famous ‘Manifold Hypothesis’ we’re often assuming a *Riemannian* manifold: samples from some continuous underlying object that can be analyzed at infinitesimal scales.

However, translating the language of Riemannian geometry to the discrete, sampled realm is troublesome. Nowhere is this more evident than curvature. It’s among the most basic of geometric properties, underlying volumes, distances, and geodesics. Where TDA captures global structure, curvature epitomizes the description of local structure. Classical definitions, however, are hard to reconcile with reality; many involve some infinitesimal shrinking, e.g. by fitting osculating circles of radius limiting to zero, or comparing volumes of balls converging to a point. Other definitions involve even more complex objects, like tangent bundles or local parameterizations of the manifold. In the discrete, sampled realm, taking such a local limit is impossible – one can’t ‘zoom in’ past the known samples. Moreover, in contrast to Riemannian manifolds, noisy samplings are *least* trustworthy at the local level – which is precisely where we wish to measure geometric information.

There have been several attempts to bridge curvature to the discrete realm. (BRIEF DESCRIPTIONS OF PREVIOUS WORK) Ollivier-Ricci Curvature and Forman Ricci curvature use optimal transport to define an edge-wise Ricci curvature at any given scale. Hickok & Blumberg define a point cloud curvature by estimating graph volumes over time, and fitting a quadratic to recover a scalar curvature. Sritharan et al. and, recently, Zhang et al. (CITATIONS PLEASE) approximate tangent planes with localized PCA, and track the movement of these planes to recover sectional curvatures. These curvatures have been found to exceed the WL-expressivity of graph neural networks, and when used to augment GNNs, improve their performance (CITE BASTION). They have also been used to study cellular differentiation, image data, etc...

With so many existing curvature methods, why introduce another one? Indeed, we came to this area interested only in using curvature to aid our study of complex high-dimensional systems like neural networks’ loss landscapes. However, our experimentation with existing methods suggested that, while existing methods have many elegant ideas, their techniques are in practice unreliable. The optimal-transport curvatures, like Ollivier Ricci, were designed for networks, and have poor resilience to the noise ubiquitous in point-cloud data. Many techniques rely on shortest-path distances, which are themselves easily corrupted by noise. And all techniques have parameters which must be carefully tuned, and – as our experiments found – give not just noisy, but *backwards* results if mistuned. And how *can* one tune these parameters, when the ground truth curvature of interest is unknown? The success of TDA is partially attributable to how easily a domain practitioner can apply it to their problem of interest; persistence homology

has few parameters and tunes itself. The field of graph curvature has no such Rosetta Stone for adapting geometric theory to the messy world of practice.

What might such a Rosetta Stone be? One candidate, with a long track-record of producing robust geometric features from high-dimensional, noisy data is *diffusion geometry*. The basic object in this framework is the random walk, or diffusion operator. When iterated, this operator endows each point with a probability distribution which is extremely robust to noise and spurious connections. In this way, a point cloud can be lifted into a *statistical manifold*: the original, unreliable sampling coordinates supplanted with diffusion probabilities. By manipulating these diffusion probabilities in various ways, one can define several varieties of ‘diffusion distances’. These distances can give rise to low-dimensional embedding and visualization techniques, like PHATE and HeatGeo. One can even bring in the Riemannian tool of pullback metrics to endow the point cloud with a Riemannian metric. Unsurprisingly, diffusion probabilities also contain information about curvature. In (CITE ORIGINAL PAPER), we presented a preliminary technique for extracting curvature from the ‘laziness’ of the diffusion operator. This fledgling method proved extremely robust to noise, though still somewhat sensitive to its parameters, and being an unsigned correlate, was unable to differentiate positive from negative curvature.

Here, we describe a significantly refined version of these ideas, which we christen *Diffusion Curvature*. It inherits the noise resilience of diffusion geometry, and, in a style reminiscent of persistence homology, further improves this resilience by incorporating many scales of diffusion into one measure. The result is far easier for the domain practitioner to use, having only one parameter, to which it is extremely robust. Moreover, we prove connections between diffusion curvature and the field’s bulwark, Ollivier-Ricci curvature, positioning diffusion curvature as a point-wise adaptation of Ollivier-Ricci curvature especially suitable for pointcloud data. We also describe a new technique for comparing diffusions across differently-sampled graphs and point clouds, with which diffusion curvature can distinguish between negative, flat, and positively curved spaces. Indeed, in our experiments, diffusion curvature appears both the most resilient to noise and the most capable of distinguishing negative from positive curvatures in high dimensions. This motivates the hypothesis that diffusion geometry is a uniquely capable ‘Rosetta Stone’ for translating Riemannian geometry into the sampled realm.

In sum, our contributions are: 1. *Diffusion Curvature*, a fast, noise-resistant and signed scalar curvature for point cloud data which is also natively differentiable. 2. Theoretical connections between the diffusion geometry underlying diffusion curvature and Ollivier-Ricci curvature, which enables adaptation of Ollivier’s results in metric measure theory to other realms of diffusion geometry. 3. *Diffusion Trajectory Normalization*, a technique for comparing diffusions across manifolds.

2 Background

2.1 Curvature in the Continuous Setting

There are many definitions of curvature on Riemannian manifolds. In this work, we focus on discrete analogs to the *Ricci* curvature $\text{Ric}(x, y)$, which we'll briefly motivate in the continuous setting.

Imagine two spheres of equal radius centered at x and y in a Riemannian manifold \mathcal{M} . Intuitively, the Ricci curvature between x and y measures the difference between the distance of these midpoints, and the average distance between corresponding points in each sphere. In an area of positive Ricci curvature, points in spheres are, on average, closer than their midpoints; in negative Ricci curvature, the points in the sphere are further, on average, than the midpoints. This is formally described by first defining the *sectional curvature* as the contraction of length incurred by parallel transport through a 2-plane of the manifold, then defining the Ricci curvature as an average of the sectional curvatures between two points. For the present work, we needn't reproduce the full definitions, but we will recall some properties of manifolds with Ricci curvature bounded from below by some k .

First, recall the relation between curvature and volume. In spaces of high positive curvature, the volume of a ball is *smaller* than a ball of the same radius in a flat space, and even smaller than a ball of the same radius in a hyperbolic space. This is formally expressed by the Bishop Gromov inequality (“Bishop–Gromov Inequality” 2021).

Theorem 2.1 (Bishop-Gromov). *Let \mathcal{M} be a complete d -dimensional Riemannian manifold with $\text{Ric}(x, y) > (d - 1)k$ for all $x, y \in \mathcal{M}$ and $k \in \mathbb{R}$. Let M_K^d be the complete d -dimensional simply connected space of constant sectional curvature k . Denote by $B(x, k)$ the ball of radius k centered at x . Then for any $x \in \mathcal{M}$ and $x_k \in M_K^d$, the function*

$$\phi(r) = \frac{\text{Vol } B(x, r)}{\text{Vol } B(x_k, r)}$$

is non-increasing on $(0, \infty)$.

This phenomenon of diminishing volume in positive curvature is related to the convergence of geodesic rays. In the plane (or saddle), two geodesic rays extending from the same point in different directions will never intersect. But on the sphere, they *will* meet again – at the opposite pole. This convergence constrains the maximum diameter a space of positive curvature may have, as expressed in the Bonnet-Myers Theorem (Ollivier 2009):

Theorem 2.2 (Bonnet-Myers). *Let X be an d -dimensional Riemannian manifold. Let $\inf \text{Ric}(X)$ be the infimum of the Ricci curvature $\text{Ric}(v, v)$ over all unit tangent vectors v . Let $S^d \subset \mathbb{R}^{d+1}$ be the unit sphere of the same dimension as X . Then, if $\inf \text{Ric}(X) \geq \inf \text{Ric}(S^d)$ then $\text{diam } X \leq \text{diam } S^d$.*

We’ll encounter versions of Bishop-Gromov and Bonnet-Myers in our discrete setting. But first, let’s describe the construction and properties of our specific discrete setting.

2.2 The Discrete Setting

Within the ambient setting of points $x_i \in \mathbb{R}^D$, the Euclidean distances between the points in our point cloud are not very useful. To perform geometric analysis, we want the manifold’s *geodesic* distances between $x_i, x_j \in \mathcal{M}$. However, manifolds are locally euclidean, so within a sufficiently small neighborhood of $x_i \in \mathcal{M}$, the euclidean distances are accurate. This is the basis of graph construction: retain only the trustworthy local distances, discard the rest, and then “integrate” over the local neighborhoods to recover features of the global geometry.

A graph $G = (V, E)$ is a collection of n vertices $v_i \in V$ connected by (possibly weighted) edges $e_{ij} \in E$. It is efficiently represented by a single *adjacency* (or *affinity*) matrix $A \in \mathbb{R}^{n \times n}$, where A_{ij} expresses the degree of connection between the vertices v_i and v_j . In a binary adjacency matrix, $A_{ij} = 1$ iff there is an edge between v_i and v_j . In a weighted affinity matrix, $0 < A_{ij} < 1$ with a higher affinity indicating a closer connection between the nodes.

One can construct an affinity matrix from a point cloud with the following algorithm: 1. Compute the matrix D of pairwise euclidean distances between points, so that $D_{ij} = \|x_i - x_j\|_2$. 2. Apply a kernel κ to the distances to construct the affinity matrix, where $A_{ij} = \kappa(D_{ij})$. This is typically the gaussian kernel:

$$\kappa(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right)$$

There are a variety of heuristics for selecting an appropriate kernel bandwidth σ . In this paper, we use an adaptive kernel bandwidth, in which, when computing $\kappa(D_{ij})$, σ is set to the mean distance from the points x_i and x_j to their k -th nearest neighbor.

After building our graph affinity matrix A , we created a new representation of the point cloud X – turning it from an $n \times D$ matrix of unwieldy ambient coordinates into an $n \times n$ matrix of pairwise connections between points. The challenge is now to reassemble this information of local connectivity to recover the features of \mathcal{M} . Graph diffusion does precisely this.

2.3 Graph Diffusion

By row-normalizing A , one obtains the graph diffusion matrix $P = D^{-1}A$, which is a commonly-used method of “integrating” the local connectivity of the graph A into global geometric descriptors of \mathcal{M} . Coifman and Lafon (Coifman and Lafon 2006) proved a correspondence between iterated graph diffusion P^t and the Neumann heat kernel on \mathcal{M} . Their technique, *Diffusion Maps*, uses the Euclidean distances between eigencoordinates of P to approximate the geodesic distances on \mathcal{M} . The visualization technique *PHATE* (Moon et al. 2019) constructs a low-dimensional embedding of a point cloud X such that a distance between the

transition probabilities P of X is preserved in the embedding. (More on properties of phase, trajectory preservation.) *Diffusion Earth Mover’s Distance* (A. Y. Tong et al. 2021) efficiently approximates the transportation distance between distributions on a graph using multi-scale wavelet transform obtained by applying different scales of diffusion. *LEGSNet*’s “learnable geometric scattering” computes tunable scales of diffusion with a graph neural network and achieves state of the art performance on biochemistry graph classification (A. Tong et al. 2021). These are but a few of the many manifold learning techniques based in diffusion.

Constructing the diffusion matrix from the affinity matrix A is straightforward: you simply row-normalize A , with an optional step to normalizing by density.

Here is the algorithm presented in Coifman and Lafon (Coifman and Lafon 2006):

1. (Optional) Compute an *anisotropic density normalization* on A , obtaining the anisotropic adjacency matrix A_\star .
2. Construct the degree matrix D , whose diagonal entries are the rowsums of A , i.e. $D_{ii} = \sum_j A_{ij}$. The other entries are zeros.
3. Define $P = D^{-1}A$, the graph diffusion matrix.

□ Clean this up: get anisotropic equation, and clarify the role of the self affinity. When is it removed? When is laziness added?

P has several nice properties. The rows $P[i]$ give the transition probabilities of a single step random walk starting at point x_i ; each row $P[i]$ can be viewed as a probability distribution centered at x_i . This is preserved under powers of the matrix. The rows of P^t still sum to 1, and $P^t[i]$ now gives the probability distribution of a t -step random walk starting at x_i .

Although P is not symmetric, it is conjugate to a symmetric matrix, via $D^{0.5}PD^{-0.5} = D^{-0.5}AD^{-0.5}$, granting it a full basis of real-valued eigenvectors and eigenvalues. These eigenvectors are shared with the normalized graph Laplacian $L = I - D^{-0.5}AD^{-0.5}$. The eigenvalues of P have magnitude less than or equal to 1. Powering the matrix P^t thus corresponds to powering the eigenvalues λ_i^t of P , via diagonalization

$$P^t = \Psi \Lambda^t \Psi^T$$

This is similar to applying a low-pass filter to the graph. As t increases, the smallest eigenvalues decay fastest under repeated powering, and their corresponding eigenvector vanishes from the eigenbasis – leaving only the largest λ_i , whose eigenvectors trace global geometric features.

This is a remarkable feature of the diffusion matrix: the ability to “denoise” itself by iterating the random walk over larger time scales. Intuitively, the paths through the data most robustly trafficked by random walkers are those supported by multiple high-probability connections from independent starting points.

2.4 Ollivier-Ricci Curvature

Developed by Yann Ollivier in 2007, *Coarse Ricci Curvature* (or sometimes, “Ollivier Ricci Curvature”) is a direct translation of Ricci curvature to discrete metric spaces like graphs (Ollivier 2009). Several classical properties of Ricci curvature can be extended to the graph setting using Coarse Ricci Curvature. Ollivier has, for instance, proven versions of concentration inequalities, Bonnet Myers (more). Coarse Ricci Curvature has, in this way, become something of a bridge between continuous and coarse geometry. The basis of this bridge is optimal transport, and specifically, the 1-Wasserstein distance.

In the Riemannian setting, Ricci curvature captures the phenomenon that, in positive curvature, “small spheres are closer (in transportation distance) than their centers are” (Ollivier 2009). On the sphere, for instance, imagine two circles centered on the north and south poles: every point in each circle is closer to the corresponding point in the opposite circle than are the circles’ centers. This effect diminishes as one moves the circles closer together, but never reaches equality. In negatively curved spaces, the discrepancy reverses, while in a flat space, the average distance between the points of the circles is the distance between the centers.

Coarse Ricci Curvature captures a similar phenomenon on graphs. Instead of spheres, it uses locally-centered probability distributions defined by random walks. And to measure the distance between these walks, it uses the 1-Wasserstein (or Earth Mover’s) distance. We’ll briefly define each.

The 1-Wasserstein distance is a measure of the distance between probability distributions. Given distributions μ_x and μ_y over some shared space X , the Wasserstein distance quantifies the smallest amount of “work” needed to transform one distribution into another, by transporting probability “mass” between pairs of points over the ground metric $d(x, y)$:

Definition 2.1 (1-Wasserstein Distance). The 1-Wasserstein distance between distributions μ_x and μ_y is

$$W_1(\mu_x, \mu_y) := \inf_{\xi \in \Pi(\mu_x, \mu_y)} \int \int d(x, y) d\xi(x, y)$$

where the “transportation plan” ξ is drawn from the space $\Pi(\mu_x, \mu_y)$ of joint probability distributions over $X \times X$ which project onto μ_x and μ_y . In the discrete setting, this translates naturally into an infimum over a summation.

$$W_1(\mu_x, \mu_y) := \inf_{\xi \in \Pi(\mu_x, \mu_y)} \sum_{x \in X} \sum_{y \in X} d(x, y) \xi(x, y)$$

What is the analog on a graph of a “small sphere” around a point? Ollivier replaces spheres with a family of measures $m_x(\cdot)$ defined for each point x , where

1. Each $\mu_x(\cdot)$ depends measurably on x , i.e. the map $x \rightarrow \mu_x$ is measurable.
2. Each $\mu_x(\cdot)$ has finite first moment, or *Jump*, i.e. for some $o \in X$ $\int d(o, y) \mu_x(y) dx < \infty$.

In graphs, Ollivier defines these μ_x as the probability distributions created by a single-step random walk from the point x .

With a transition probability α , and equal probability of moving to each of x 's neighbors on the graph, $\mu_x(x) = (1 - \alpha)$ and $m_x(y) = \alpha$ if $y \in N(x)$ or 0 otherwise.

This is analogous to defining $m_x = Pe_x$, if P is the diffusion matrix created from a binary adjacency matrix. Note, however, that there is nothing limiting us to binary adjacency matrices, or even single steps of diffusion; the two conditions above are equally satisfied by weighted adjacency matrices and t -step diffusions, and in sparse or noisy graphs, this may be desirable.

Definition 2.2 (Coarse Ricci Curvature). The *Coarse Ricci Curvature* between x and y is

$$\kappa(x, y) := 1 - \frac{W_1(m_x, m_y)}{d(x, y)}$$

There are a number of provisos attached to this definition, which tries to approximate a continuous phenomenon within discrete constraints. These constraints, and the relationship between Ricci and Ollivier's coarse Ricci curvature are illustrated Ollivier's Example 2.6 (Ollivier 2009):

Example 2.1. Let (X, d) be a smooth Riemannian manifold of dimension d and let vol be the Riemannian volume measure. Let $\epsilon > 0$ small enough and consider the ball of radius ϵ around each point x . Let $x, y \in X$ be two sufficiently close points. Let v be the unit tangent vector at x directed towards y . The coarse Ricci curvature along v is then

$$\kappa(x, y) = \frac{\epsilon^2 \text{Ric}(v, v)}{2(d + 2)} + o(\epsilon^3 + \epsilon^2 d(x, y))$$

Hence the coarse Ricci curvature applied to a manifold recovers the Ricci curvature, up to a scaling factor contingent on dimension, and plus an error term that grows with the radius of ball and distance between points.

Ollivier's choice not to scale $\kappa(x, y)$ by dimension is interesting, and likely motivated by his application of coarse Ricci curvature to graph-like spaces for which dimension isn't clearly defined, like social networks. Within our domain of point-cloud data, incorporating dimension may be desirable; without it, spaces of high dimension can be conflated with spaces of lower dimension but higher negative curvature.

□ Ollivier actually does define a local diffusion-based dimension

A result on coarse Ricci curvature which will prove useful concerns the *contraction (or expansion) of measure* that occurs under diffusion in spaces of positive (or negative) curvature. This is Ollivier (2009)'s Proposition 20:

Proposition 2.1 (W_1 Contraction of Measure). *Let (X, d, m) be a metric space with a random walk. Let $\kappa \in \mathbb{R}$. Then we have $\kappa(x, y) \geq \kappa$ for all $x, y \in X$ iff for any two probability distributions $\mu, \mu' \in \mathcal{P}(X)$ one has*

$$W_1(\mu \star m, \mu' \star m) \leq (1 - \kappa)W_1(\mu, \mu')$$

Where

$$\mu \star m := \int_{x \in X} d\mu(x) m_x dx$$

3 Methods

We consider a measure space (X, p) equipped with a random walk, e.g. a point cloud and a diffusion operator derived as above. In this section, we define a new notion of curvature deriving from this random walk.

The core intuition of Diffusion Curvature is that the *laziness* of random walks on a graph is a proxy for the curvature of the underlying manifold. Picture a “random walker” drunkenly traversing a sphere. If he manages, over the course of several steps, to wander to the opposite pole, he has many ways of getting back to where he started. By contrast, if he begins on the top of a (negatively-curved) saddle and wanders down one side, any path aside from exactly retracing his steps incurs a steep penalty in extra distance. On the sphere, the random walker is more likely to find his way home: his walks are “lazier”.

Our previous paper (Bhaskar et al. 2022) measured this laziness directly, as the return probability within a k -neighborhood of the starting point. This required tuning the parameter k , and neglected the information provided by probabilities outside of this neighborhood.

We refine our previous definition of diffusion laziness by replacing the neighborhood-sum with a distributional distance measure D between a dirac δ_x and its t -step diffusion $p_x^X(t)$. This provides a more sensitive and parameter-free measure of how ‘spread out’ the diffusion is.

Definition 3.1 (Diffusion Energy Snapshot). Given some distributional distance D , the D -energy snapshot of a diffusion $p_x^X(t)$ is

$$l_X(x, t) := D(\delta_x, p_x^X(t))$$

By $p_x^X(t)$, we indicate the t -step diffusion of the dirac δ_x centered at $x \in X$, using the diffusion operator for the measure space X . We use the term ‘Diffusion Energy’ in reference to our previous work (CITE), which measured curvature via ‘Diffusion Laziness’. That laziness measurement increased with positive curvature; diffusion energy does the opposite, increasing with more negative curvature.

The distributional distance D may be the Wasserstein-1 distance, as with Ollivier-Ricci curvature; it may be the Jensen-Shannon Distance, evoking the classical use of entropy to measure the diffusion of heat; or it may be some other domain-specific form chosen for a particular metric measure space.

This measure of diffusion energy provides a snapshot of curvature at single diffusion time. However, this curvature is unsigned, and sensitive to the choice of t . Moreover, as we'll discuss, differences in graph construction cause the times associated with the diffusion operators from different spaces to change at different speeds. This makes it hard to compare the geometric information gleaned from the diffusion operator across manifolds.

Instead of these static and fickle snapshots of a diffusion process, we consider the process in its entirety. As t ranges from 0 to ∞ , the diffusions $p_x^X(t)$ can be viewed as a trajectory on the n -dimensional probability simplex, where n is the number of points in X . Given a distributional distance and its generating Riemannian metric, we can define Diffusion Curvature as the path integral of diffusion energy along this trajectory.

Definition 3.2 (Diffusion Energy). Given a metric g_p on the statistical manifold of diffusion probabilities, and the distributional distance D it generates the Diffusion Energy of a point $x \in X$ is

$$l_X(x, d) = \int_{t=0}^{T(d)} D(\delta_x, p_x^X(t)) g_p \left(\frac{dp_x^X(t)}{dt}, \frac{dp_x^X(t)}{dt} \right) dt$$

where $T(d)$ is a function that converts between the calculated *Diffusion Trajectory Distances* d and diffusion times t .

In practice, we don't have an infinitesimal parameterization of p_x^X , so approximate this with the algorithm in BOX WAWA: we compute T scales of diffusion (e.g. 50), sum the distributional distances between each scale to estimate the path integral, and then use a trapezoidal approximation of the integral of $L_x(t)$ with respect to the distances along the trajectory.

By viewing the energy integrated along a diffusion trajectory, we eliminate many of the differences between diffusions on different spaces. This enables the computation of a *signed* curvature, by taking the difference of diffusion energies from a given space and a comparison space of known Euclidean curvature and equal dimension to X . In practice, we construct the comparison space as uniform samples from \mathbb{R}^d , where d is the intrinsic dimension of X .

Definition 3.3 (Diffusion Curvature). Given points $X \subseteq \mathcal{M}$, where \mathcal{M} is a Riemannian manifold of dimension d , the *Diffusion Curvature* of $x \in X$ is

$$k(x, d) = l_X(x, d) - l_{\mathbb{R}^d}(x, d)$$

where $E \subseteq \mathbb{R}^d$ is a collection of uniformly sampled points from \mathbb{R}^d .

The usefulness of this definition rests upon two claims:

1. That diffusion energy is indeed a measure of (unsigned) curvature
2. That our integration along diffusion trajectories enables this energy to be compared across point clouds with different diffusion processes.

The next two sections will analyze this first claim. We first analyze diffusion energy within Ollivier’s framework of metric measure theory, and bound the diffusion energy from above by the Ollivier-Ricci curvature. We then provide a motivating example recovering a Bishop-Gromov type volume comparison from the diffusion curvature with an entropic distance.

Finally, we support the second claim with illustrations of point clouds which diffusion trajectories can uniquely compare. We also describe how common techniques of constructing graphs from point clouds can obscure geometric information, and propose a ‘curvature-agnostic kernel’ to facilitate more geometrically-faithful graph construction.

3.1 Diffusion Energy is bounded by Ollivier-Ricci curvature

Ollivier Ricci curvature measures the extent to which ‘spheres are closer than their centers’, using graph diffusions to create spheres, and optimal transport to measure distances. Diffusion energy uses both ingredients. But whereas Ollivier Ricci curvature takes the Wasserstein distance between spheres of the same size, Diffusion energy takes the distance between successive sizes of the same sphere.

This simplification makes Diffusion energy more localized and much computationally lighter. Here we demonstrate that it also retains the theoretical guarantees of Ollivier Ricci curvature, and might be thought of as a natural node-wise adaptation of Ollivier Ricci curvature.

Proposition 3.1 (Coarse Ricci Curvature Bounds Diffusion Energy). *Let (X, d, m) be a metric space equipped with a random walk, with coarse Ricci curvature bounded from below by some k such that $\kappa(x, y) \geq k$ for all $x, y \in X$. The diffusion energy of a t step diffusion in X is bounded above by*

$$W_1(\delta_x, m_x^t) \leq W_1(\delta_x, m_x) \frac{(1 - (1 - k)^t)}{k}$$

In particular, if $k > 0$ then $W_1(\delta_x, m_x^t) \leq \frac{W_1(\delta_x, m_x)}{k}$, and if $k = 0$, then $W_1(\delta_x, m_x^t) \leq tW_1(\delta_x, m_x)$.

To prove this, first we bound $W_1(m_x^t, m_x^{t+1})$ using Proposition 2.1. The proposition states that a lower bound on curvature, such as we have, implies that

$$W_1(\mu \star m, \mu' \star m) \leq (1 - k)W_1(\mu, \mu')$$

where here μ, μ' are two probability distributions and m is a random walk. This provides an easy lemma:

Lemma 3.1 (Inductive Contraction of Measure). *Let (X, d, m) be a metric space with a random walk. Suppose there is some $k \in \mathbb{R}$ such that the coarse Ricci curvature $\kappa(x, y) \geq k$ for all $x, y \in X$. Then:*

$$W_1(m_x^t, m_x^{t+1}) \leq (1 - k)^t W_1(\delta_x, m_x)$$

Proof. We proceed by induction. For $t = 0$, the above is true, as $W_1(m_x^0, m_x^1) = W_1(\delta_x, m_x) = W_1(\delta_x, m_x)$. Suppose it holds for $t - 1$, e.g.

$$W_1(m_x^{t-1}, m_x^t) \leq (1 - k)^{t-1} W_1(\delta_x, m_x)$$

Consider $W_1(m_x^{t-1} \star m, m_x^t \star m)$, the application of another step of diffusion. By Ollivier's Proposition 20, this distance is bounded above by

$$W_1(\mu_1 \star m, \mu_2 \star m) \leq (1 - k) W_1(\mu_1, \mu_2)$$

So

$$W_1(m_x^{t-1} \star m, m_x^t \star m) \leq (1 - k) W_1(m_x^{t-1}, m_x^t)$$

which, since the statement holds for $t - 1$, yields

$$W_1(m_x^t, m_x^{t+1}) \leq (1 - k)^t W_1(\delta_x, m_x)$$

□

We can now use this lemma to decompose our t -step diffusion into a sum of single-step diffusions, allowing an easy proof of the proposition.

Proof. By the triangle inequality,

$$W_1(\delta_x, m_x^t) \leq W_1(\delta_x, m_x) + W_1(m_x, m_x^2) + \dots + W_1(m_x^{t-1}, m_x^t)$$

By Lemma 3.1,

$$\leq W_1(\delta_x, m_x) (1 + (1 - k) + (1 - k)^2 + \dots + (1 - k)^{t-1})$$

This truncated series is equal to $\frac{1 - (1 - k)^t}{1 - (1 - k)} = \frac{(1 - (1 - k)^t)}{k}$. If $k > 0$, then as $t \rightarrow \infty$, the infinite sum $\sum_{i=0}^t (1 - k)^i$ converges to the geometric series $\frac{1}{1 - (1 - k)} = \frac{1}{k}$. Because the sum is monotonically increasing with t , the partial sum is upper bounded by the infinite sum. It follows that

$$W_1(\delta_x, m_x^t) \leq \frac{W_1(\delta_x, m_x)}{k}$$

If $k = 0$, then obviously $\sum_{i=0}^t (1 - k)^i = t$.

□

The above shows that the diffusion energy l_{W_1} has an upper-bound given by the Ollivier Ricci curvature. We expect that the higher a manifold’s curvature, the lower l_{W_1} should be. This is formalized by Proposition 1: the higher the Ollivier-Ricci curvature of a manifold, the smaller its diffusion energy. This result motivates the use of diffusion curvature as a point-wise adaptation of Ollivier-Ricci curvature.

3.2 A Motivating Example: Entropic Diffusion Curvature Recovers a Bishop-Gromov Volume Comparison

Among the most obvious manifestations of curvature is differences in volume: the higher the curvature, the smaller the volume of geodesic balls of the same radius. In the continuous setting, this is expressed by Theorem 2.1.

One route to discrete curvature is thus estimating volumes; this is the route taken by Hickok and Blumberg (2023). However, standard methods of estimating volumes in discrete, sampled spaces like point clouds or graphs rely on more complex geometric quantities, like density estimation or approximated geodesic distances, which are not only susceptible to noise, but often themselves dependent on the curvature!¹ One wonders: can we use the tools of diffusion geometry to robustly estimate volume?

Indeed, this is one way to interpret diffusion energy: as an inverse volume measurement. Huguet et al. (2023) motivates this with a result from Saloff Costes et al. (Saloff-Coste 2010). For manifold satisfying the Parabolic Harnock Inequality (e.g. all manifolds of positive curvature), heat diffusion m_x^t on the manifold is bounded by

$$\frac{c_1}{V(x, \sqrt{t})} \exp\left(-\frac{d(x, y)^2}{c_2 t}\right) \leq m_x^t(y) \leq \frac{c_3}{V(x, \sqrt{t})} \exp\left(-\frac{d(x, y)^2}{c_4 t}\right)$$

In other words, heat diffusion on non-Euclidean manifolds behaves approximately like Euclidean heat diffusion scaled by the local volume.

Powering the diffusion matrix approximates heat diffusion [cite Coifman]. We can thus relate our measure of the ‘spread’ or ‘energy’ of this diffusion to the manifold’s volume.

The Shannon entropy of the diffusion m_x over the bounded neighborhood of X with non-zero diffusion mass, can be written

$$H(m_x^t) = - \int_{y \in X} m_x^t(y) \ln(m_x^t(y))$$

¹Consider kernel density estimation, where one places gaussians on each data point and sums up the probability assigned to each area. The higher the curvature, the higher the reported density, since at the end of an ellipsoid the gaussians have greater overlap with each other than in its center – even though the intrinsic density of the surface is uniform.

Here we use Shannon's differential entropy. Though this formulation encounters pathologies on unbounded domains, it approximates the entropy a signal on a sampled region as the number of samples goes to infinity.

We'll simplify the notation by combining the integral and distribution into an expected value over m_x^t

$$= -\mathbb{E}_{m_x} \ln(m_x(y)) dy$$

Recalling the above result, there are constants $c_1 \dots c_4$ such that

$$\frac{c_1}{V(x, \sqrt{t})} \exp\left(-\frac{d(x, y)^2}{c_2 t}\right) \leq m_x^t(y) \leq \frac{c_3}{V(x, \sqrt{t})} \exp\left(-\frac{d(x, y)^2}{c_4 t}\right)$$

hence we can approximate $H(m_x^t)$ by

$$\begin{aligned} &= -\mathbb{E}_{m_x^t} \ln\left(\frac{1}{V(y, \sqrt{t})} \exp\left(\frac{-d(x, y)^2}{4t}\right)\right) \\ &= -\mathbb{E}_{m_x^t} \ln\left(\frac{1}{V(y, \sqrt{t})}\right) - \frac{d(x, y)^2}{4t} \\ &= \mathbb{E}_{m_x^t} \ln(V(y, \sqrt{t})) + \mathbb{E}_{m_x^t} \frac{d(x, y)^2}{4t} \end{aligned}$$

By assumption, the volumes $V(y, \sqrt{t})$ equal some constant $V_m(\sqrt{t})$ over the support of m_x^t . Also note that the right hand term is precisely $\frac{1}{4t} W_2(\delta_x, m_x^t)$, so

$$H(m_x^t) \simeq \ln(V_m(\sqrt{t})) + \frac{1}{4t} W_2(d_x, m_x^t)$$

The Diffusion Curvature is then

$$\frac{1}{t}(H(m_E^t) - H(m_x^t)) \simeq \frac{1}{t} \ln(V_E(\sqrt{t})) - \frac{1}{t} \ln(V_m(\sqrt{t})) + \frac{1}{4t^2} (W_2(d_x, m_E^t) - W_2(d_x, m_x^t))$$

This presents us with two comparisons: one between the average volume within $N_t(x)$ both in Euclidean space and on the manifold, and one between the Wasserstein spread of diffusion in Euclidean space and on the manifold. Note that both exhibit the same inverse relationship to curvature. If the curvature of m is greater than in euclidean space (e.g. positive), the volume of a ball is smaller – just as the spread of diffusion is diminished. In negative curvature, we have the opposite.

3.3 Comparing Diffusion Across Point Clouds

3.3.1 Problem Setting & Motivation

For a single graph, the energy of diffusion is linked to graph volume and theoretically bounded by curvature. Within a given graph, the lower the curvature, the higher the diffusion energy. As shown above, we can draw analogy to Bishop-Gromov to convert this unsigned magnitude-of-curvature estimation to a signed curvature via a comparison between spaces.

A comparison like this is at the heart of many discrete curvatures. Ollivier-Ricci compares the transport distance between two distributions with the transport distance on a flat space. Hickok and Blumberg compare the approximate the volume of balls on a given point cloud with the Euclidean volume of balls with the same radius. In these definitions, the use of more complex geometric constructs (distances, volumes, densities) grants the Euclidean comparison an analytic form. Our definition makes a different trade-off: it doesn't depend on the error-prone approximations of those higher-order phenomena, but it doesn't benefit from their theoretical simplicity.

To put it bluntly, there's no analytical form of, e.g. the diffusion energy snapshot in \mathbb{R}^8 at $t = 25$ because there's no single translation between \mathbb{R}^8 and one of the "discrete measure space with a random walk" our definition operates on. There are many different ways of uniformly sampling N points from \mathbb{R}^d – and there are many ways of turning those points into a graph. But different samplings, and different modes of graph construction result in different *speeds of diffusion* on the graph, rendering direct comparison of diffusion energies difficult.

For example, imagine constructing two k -nearest neighbor graphs with a different value of k for each pointcloud. In the graph constructed with a larger k , diffusion will happen more quickly because everything is more tightly connected. The diffusion energy of a point on this graph, at any given t , will appear much higher than its sister graph – even if both were constructed from the same space! The vagaries of graph construction have obscured the underlying geometry of the manifold.

An immediate workaround is to try to bypass this obstruction by constructing graphs in *exactly* the same manner from both our target points and comparison points. One might use a k nn graph with identical k 's, or an adaptive kernel which sets the bandwidth locally based on the distance to each nearest neighbor. We experimented with this approach extensively, and came to the unexpected conclusion that the interplay between curvature and distance makes it extremely hard to create a kernel with the same relative bandwidth in spaces of differing curvatures. Specifically, the local Euclidean distance used by standard graph construction techniques like the k -neighbor adaptive kernel becomes less reliable the greater the magnitude of curvature. Thus, not only are comparisons between spaces compromised, but even the geometric information within a space is obscured by a gratuitously fluctuating kernel bandwidth. We explore this in more detail in the next section, and propose a simple change which lessens these symptoms.

However, the underlying ‘disease’ remains: although we assume our point clouds are sampled manifolds, there’s no one-to-one translation between these manifolds and the measure spaces with random walks diffusion geometry operates on. Different samplings and different means of graph construction all create different random walks. Trying to homogenize these random walks via careful graph construction relies on geometric information, like distances, that we don’t (yet) have – and tries to wage war against the notoriously fickle process of sampling. Far better than trying to homogenize these random walks on the level of *graph construction* would be a method that can reliably compare diffusions *across* graphs.

This problem, of reliably comparing diffusions across graphs, has wide applicability. ... research needed here ... Other techniques partially address this by considering *multiscale diffusions*. The *multiscale diffusion distance* takes a weighted average across powers of diffusion. Diffusion Earth Mover’s Distance extends this to an optimal transport measure based on the differences distributions diffused at multiple scales. The *Graph scattering transform* produces a rich set of features for comparing graphs derived from differences between powers of diffusion. And many graph neural networks are built upon flavors of message passing which resemble graph diffusion, while others have been modeled on graph scattering. All of these methods come with a caveat: if compared across graphs, discrepancies in graph construction will manifest as spurious differences in graph geometry.

3.3.2 Diffusion Trajectory Alignment

We consider a new approach to this problem, which replaces the diffusion time t – so heavily dependent on the vagaries of graph construction – with a distance along the *diffusion trajectory*. To obtain this, we view scales of diffusion as points along a trajectory on the N -dimensional probability simplex, where N is the number of nodes on the graph.² This trajectory has the advantage of erasing any differences caused by the speed of diffusion. Instead, we can use the distance along that trajectory (*Diffusion Trajectory Distance*) to align the scales of diffusion between two graphs. As illustrated in Figure 1, by comparing the diffusion energies of several graphs across their diffusion trajectory distances. Making the same comparison across times is uninformative – and misleadingly suggests that the sphere is the most negatively curved. *Diffusion Trajectory Alignment*, however, reveals the correct relation between the surfaces.

²This simplex corresponds to the ‘diffusion coordinates’ used by Coifman, as well as the setting of Fasina’s diffusion-based Fisher Information metric (CITE).

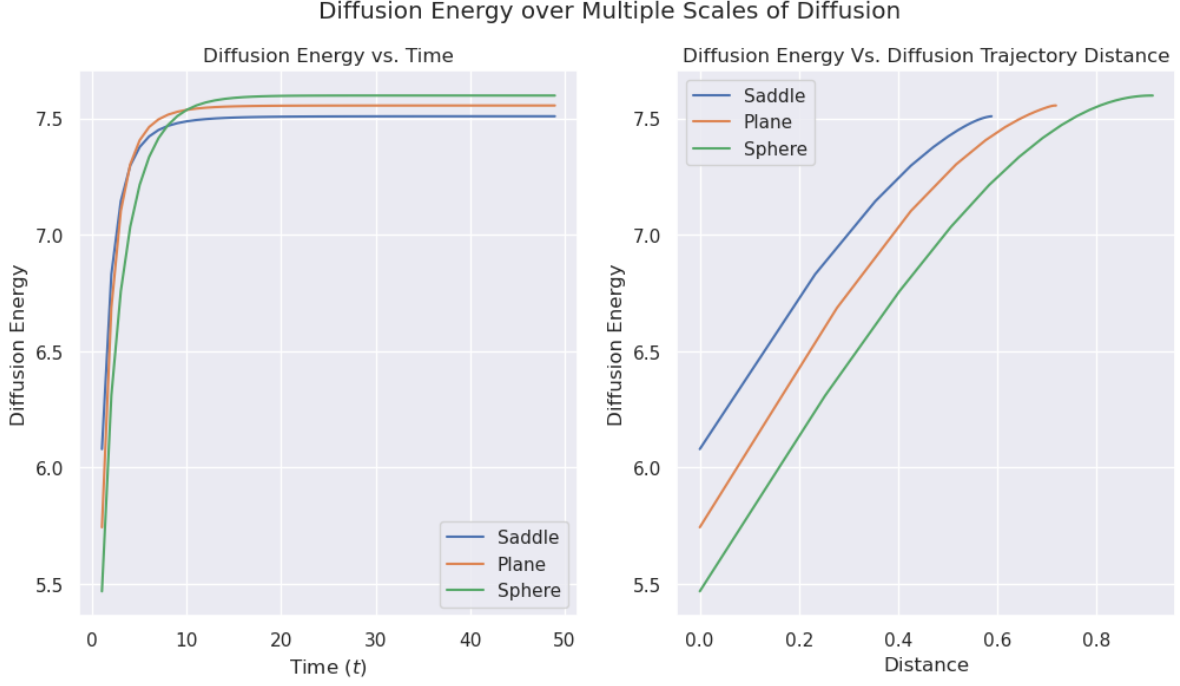


Figure 1: Diffusion energies across multiple scales on a 4-dimensional saddle, plane, and sphere (all 3-manifolds). When comparing directly between times, there is no clear separation of curvatures. Aligning diffusions by diffusion trajectory distance reveals curvature.

Source: [Standard libraries](#)

Applying diffusion trajectory alignment to our diffusion energies also has the happy effect of focusing our measurement on the most relevant scales of diffusion. When using diffusion times, one must be careful to select a time that is neither too small nor too large. Make t too small, and the diffusion might have barely differentiated itself from a dirac. Make t too large, and the diffusion might have progressed to the uninformative steady-state distribution. With the diffusion trajectory distance, all of the undesirably large t 's are compressed into a narrow range of distances – the diffusion trajectory limits to the steady state distribution, and the diffusion trajectory distances that correspond to it are all within some ϵ of the maximum diffusion trajectory distance. Hence, any diffusion trajectory distance between its minimum and maximum values corresponds to an interesting scale of diffusion.

With the alignment help of the diffusion trajectory distance, we can now motivate our path-integral definition of diffusion energy (Definition 3.2):

$$l_X(x, d) = \int_{t=0}^{T(d)} D(\delta_x, p_x^X(t)) g_p \left(\frac{dp_x^X(t)}{dt}, \frac{dp_x^X(t)}{dt} \right)$$

Recall that $D(\delta_x, p_x^X(t))$ is the snapshot diffusion energy (Definition 3.1) at time t . This is integrated with respect to the diffusion trajectory distance, here formulated via a Riemannian metric on the probability simplex. The parameter T controls how local the measurement of diffusion energy is. As T approaches ∞ , the diffusion approaches the steady state across the entire graph, and the diffusion energy measures a global curvature. Smaller T 's yield more localized measurements of curvature. In practice, T is set via some $T(d)$ that translates diffusion trajectory distances into the corresponding times.

The diffusion energy can be thought of as an integral of the geodesic distance from a distribution to a dirac along the distributions in a given diffusion trajectory. Imagine a long, bent noodle; while traveling along the noodle, we integrate the distances from each point on the noodle to its base point. If these geodesic distances corresponded to travel back along the diffusion trajectory, or back down the noodle – e.g. if the diffusion trajectory took the shortest path between the dirac and each subsequent distribution – then the diffusion energy would be equal across all manifolds. The extent to which this fails, and the diffusion trajectory takes a ‘lazier’ path than optimal, reveals the curvature.

This measurement of diffusion energy can now be converted into a signed *diffusion curvature* via Definition 3.3:

$$k(x, d) = l_X(x, d) - l_X(x, d)$$

This corresponds to taking the area under each of the curves in Fig. WAWA, up to some trajectory distance d , and subtracting them.

In practice, we don’t have access to a metric g_p on the probability simplex; this space is too high-dimensional, and we lack a continuous parameterization of graph diffusion. Instead, we approximate this integral by measuring the distances between subsequent diffusions (performed for every integer $1 \leq t \leq T(d)$). This approximates the diffusion trajectory distances for each scale of diffusion. We then use the trapezoidal rule to approximate the integral of diffusion energies with respect to these trajectory distances. This is summarized by Algorithm 1 REFERENCE WAWA.

3.3.3 The Curvature-Agnostic Kernel & Automatic Bandwidth Tuning

Diffusion trajectory alignment significantly reduces the interference caused by graph construction, but does not completely eliminate it. Due to our lack of access to a diffusion coordinate space metric, and the approximations used for the diffusion trajectory distance, we find the robustness of our algorithm is significantly improved by a few adjustments to standard graph construction techniques to make them more curvature agnostic.

To turn a point cloud into a graph, one places a kernel (e.g. a gaussian) at each point, and assigns edges to surrounding points with weights given by the kernel. The main parameter here is the bandwidth of each kernel: make it too high, and everything is connected; too

low, and points are isolated. Moreover, the distances between a point and its neighbors may vary. Where the points are sparse, one wants a higher bandwidth. Where dense, a lower bandwidth.

A standard technique for accommodating the variable densities of point cloud data is a k -neighbor *adaptive kernel*. Here, the bandwidth at each point is set to the squared distance to the k -th nearest neighbor, averaged in some way between the two inputs. Here's one standard adaptive kernel:

$$k(x, y) = \frac{e^{-\frac{d(x, y)^2}{d(x, N_k(x))^2}}}{d(y, N_k(y))} + \frac{e^{-\frac{d(x, y)^2}{d(y, N_k(y))^2}}}{d(x, N_k(x))}$$

where $N_k(x)$ gives the k th nearest neighbor to x .

Note that that parameter k plays roles in two kinds of adaptation. By setting the bandwidth to some Euclidean distance $d(x, N_k(x))$ between data points, it adapts the kernel to the *right scale* for the point cloud, concentrating its probability within a local neighborhood. It also controls the *size* of that local neighborhood; typically set between 5 and 15.

Unfortunately, this adaptive process hinges on the correspondence between the Euclidean distances $d(x, N_k(x))$, and the manifold distances – a correspondence that degrades *faster* the higher the magnitude of curvature! Consider a highly-curved paraboloid. The Euclidean distance from its center point to its nearest few neighbors is roughly the geodesic distance. But as the neighborhood widens, and the neighbors move higher up the paraboloid, the Euclidean distance now underestimates the true geodesic distance: it cuts across the manifold. This happens in spaces of high positive or high negative curvature.

As a result, the adaptive kernel constructed in highly curved spaces has a smaller bandwidth than in flat spaces, even using the same kernel parameters. If one directly compares the laziness of diffusion on two such graphs, it will appear lazier than on the flat manifold.

To remedy this, we make a simple adjustment to the standard adaptive kernel. The key is separating the parameter which controls neighborhood size from neighborhood scale. In the adaptive kernel, both are controlled by the parameter k ; as discussed above, in spaces of high curvature, the same k creates different scales. Instead, we always set $k = 1$ and use another parameter – ν , the neighbor-size – to control the variance of the kernel.

$$k(x, y) = \frac{e^{-\frac{d(x, y)^2}{d(x, N_1(x))^2 \nu^2}}}{d(y, N_1(y))} + \frac{e^{-\frac{d(x, y)^2}{d(y, N_1(y))^2 \nu^2}}}{d(x, N_1(x))}$$

ν behaves like the kernel bandwidth in the regular fixed Gaussian kernel. We call this the *Curvature-Agnostic Kernel*.

- Figure comparing the adaptive to curvature agnostic kernels with diffusion trajectory normalization. Use a stacked image.

Figure WAWA illustrates the increased robustness diffusion trajectory alignment gains from using this curvature-agnostic kernel. The relative ordering of graphs is still achieved by the traditional adaptive kernel, but is significantly denoised by these tweaks.

The main parameter of the curvature-agnostic kernel is the “neighbor scale” ν , which sets the kernel bandwidth to the specified multiple of the average distance to the k th-nearest neighbor. As a single parameter, this can be tuned easily. We use as a heuristic the desired median number of points with non-negligible single-step diffusion probability, and perform a binary search over values of ν until this number is within a desired tolerance.

We suspect the curvature-agnostic kernel may be generally useful in manifold learning. In any downstream analysis that leverages the graph geometry – be it distance estimation, [CITE HUGUET] or metric learning [CITE US] – this analysis is made easier by a kernel whose bandwidth does not gratuitously fluctuate in regions of high curvature.

- Can we provide experimental results here? Does PHATE or HeatGeo perform better with this kernel?

4 Results

Curvature is an easy quantity to find – indeed, at least two of the many recent curvature methods were defined *accidentally* (Steinerberger 2022; Bhaskar et al. 2022)! The challenge is to make one’s definition robust to the bogeymen of real-world data, especially 1) noise and 2) high dimensions. Our benchmarks of existing methods, including Hickok & Blumberg’s, Olivier Ricci Curvature, Adal-PCA [and Sritharan], [citations] reveal that all deteriorate quickly under noise, and most *fail entirely* in high-dimensional data. The dimension-independent, noise-resilient properties of the diffusion operator endow diffusion curvature with good performance on both of these axes.

But first, a sanity check. Does diffusion curvature recover the Gaussian curvature of basic 2-manifolds? Figure 2 shows the diffusion curvature of a Torus, Saddle, and Ellipsoid, along with a scatter plot correlating diffusion curvature and Gaussian curvature. Figure 3 places this in the context of existing methods.

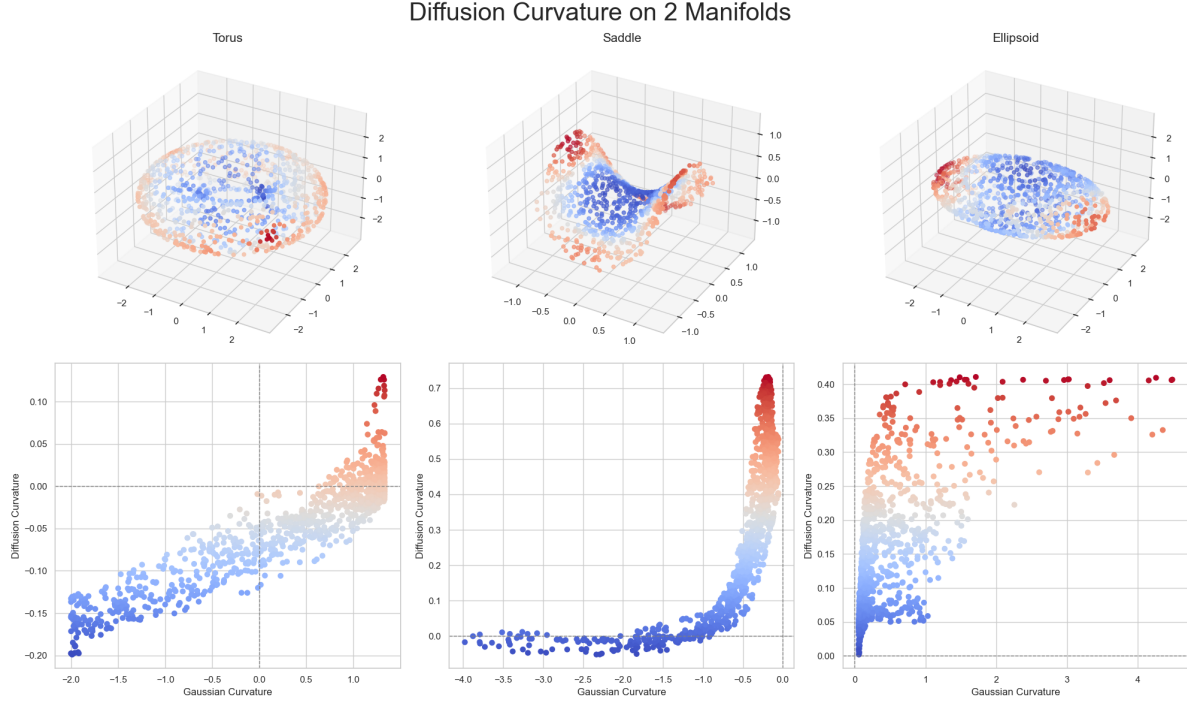


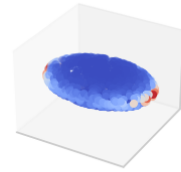
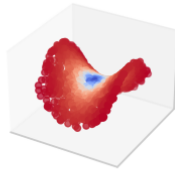
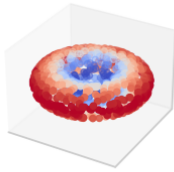
Figure 2: Diffusion Curvature vs Gaussian Curvature on 2-Manifolds.

Source: [Standard libraries](#)

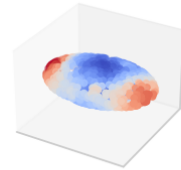
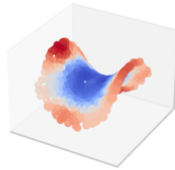
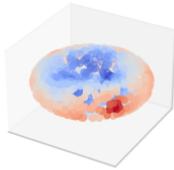
Though the correlation is strong – passing the ‘sniff test’ – this low-dimensional validation highlights two subtleties of our method. First, diffusion curvature, being an intrinsic, graph-based measurement, is susceptible to edge effects to a greater degree than extrinsic methods. When diffusion hits the edges of the saddle, it rebounds, creating the false appearance of positive curvature.

Estimated Gaussian Curvatures of 2-Manifolds

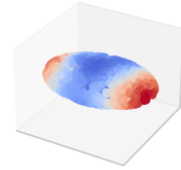
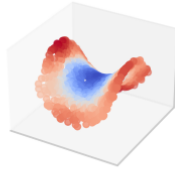
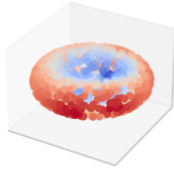
Ground Truth Curvature



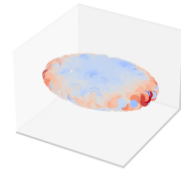
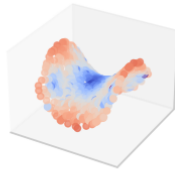
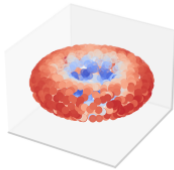
Diffusion Curvature



Unsigned Diffusion Curvature



AdaL-PCA



Hickok & Blumberg

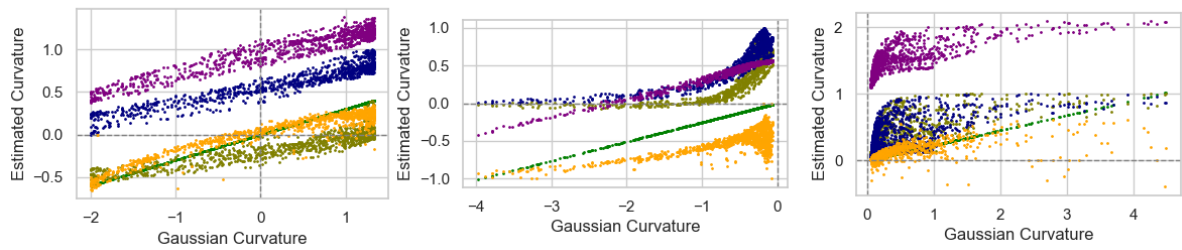
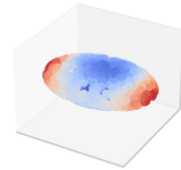
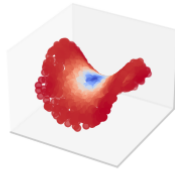
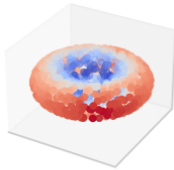


Figure 3: Diffusion Curvature vs Gaussian Curvature on 2-Manifolds.

Source: [Standard libraries](#)

The second subtly emerges in the context of related methods. Everything pictured does an excellent job of coloring the toy manifolds. But looking beyond this, *what matters?* Most methods don't care about the precise magnitude – or even exactly matching the correlations, as in the ellipsoid... %% mean vs gaussian curvature, and uniqueness of definitions%%

All of these methods in Figure 3 perform well on well-sampled noiseless toy manifolds. In Table ?@fig-2-manifolds-noise-table, we see the results of adding Gaussian noise to each manifold.

4.0.1 Differentiating Sign in High Dimensions

Most existing methods only quantify their performance in high dimensions on one or two test cases.

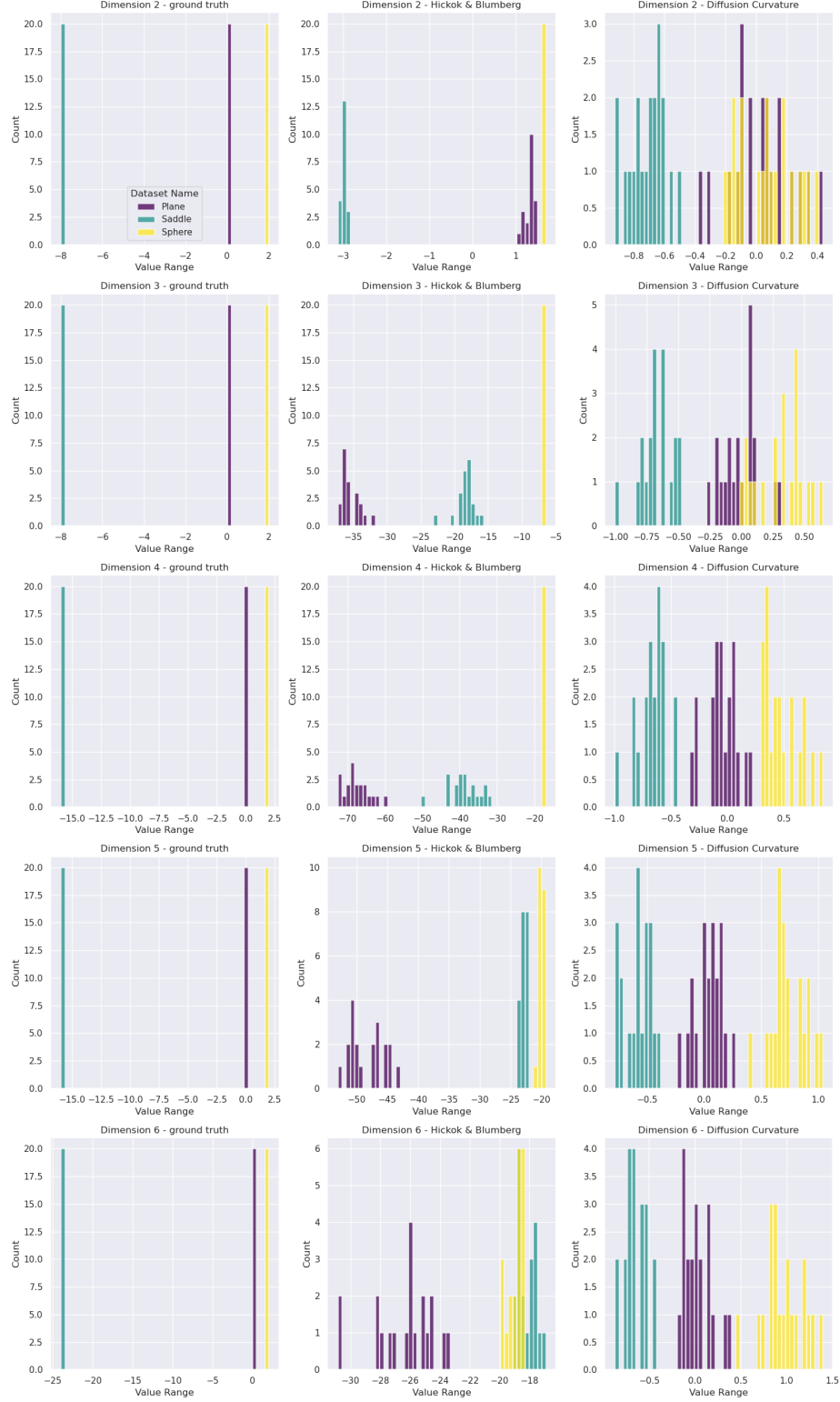


Figure 4: Predicted curvatures of Saddles and Spheres in dimensions 2-6. Diffusion Curvature robustly distinguishes between the signs of the data, even in high dimensions, and with relative sparsity.

4.1 Loss Landscapes

4.2 Curvature as a TDA Filtration

5 Related Work

5.1 Foreman Ricci Curvature

5.2 Hickock & Blumberg’s Volume Comparison Curvature

5.3 Sritharan

6 Conclusion

References

- Bhaskar, Dhananjay, Kincaid MacDonald, Oluwadamilola Fasina, Dawson Thomas, Bastian Rieck, Ian Adelstein, and Smita Krishnaswamy. 2022. “Diffusion Curvature for Estimating Local Curvature in High Dimensional Data.” *Advances in Neural Information Processing Systems* 35: 21738–49. https://proceedings.neurips.cc/paper_files/paper/2022/hash/88438dc62fc5c8777e2b5f1b4f6d37a2-Abstract-Conference.html.
- “Bishop–Gromov Inequality.” 2021. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Bishop%E2%80%93Gromov_inequality&oldid=1059331416.
- Coifman, Ronald R., and Stéphane Lafon. 2006. “Diffusion Maps.” *Applied and Computational Harmonic Analysis*, Special Issue: Diffusion Maps and Wavelets, 21 (1): 5–30. <https://doi.org/10.1016/j.acha.2006.04.006>.
- Hickok, Abigail, and Andrew J. Blumberg. 2023. “An Intrinsic Approach to Scalar-Curvature Estimation for Point Clouds.” arXiv. <https://doi.org/10.48550/arXiv.2308.02615>.
- Huguet, Guillaume, Alexander Tong, Edward De Brouwer, Yanlei Zhang, Guy Wolf, Ian Adelstein, and Smita Krishnaswamy. 2023. “A Heat Diffusion Perspective on Geodesic Preserving Dimensionality Reduction.” May 30, 2023. <https://doi.org/10.48550/arXiv.2305.19043>.
- Moon, Kevin R., David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S. Chen, Kristina Yim, et al. 2019. “Visualizing Structure and Transitions in High-Dimensional Biological Data.” *Nature Biotechnology* 37 (12, 12): 1482–92. <https://doi.org/10.1038/s41587-019-0336-3>.
- Ollivier, Yann. 2009. “Ricci Curvature of Markov Chains on Metric Spaces.” *Journal of Functional Analysis* 256 (3): 810–64. <https://doi.org/10.1016/j.jfa.2008.11.001>.

- Saloff-Coste, Laurent. 2010. “The Heat Kernel and Its Estimates.” In *Advanced Studies in Pure Mathematics*, 405–36. Kyoto University, Japan. <https://doi.org/10.2969/aspm/05710405>.
- Steinerberger, Stefan. 2022. “Curvature on Graphs via Equilibrium Measures.” September 5, 2022. <https://doi.org/10.48550/arXiv.2202.01658>.
- Tong, Alexander Y., Guillaume Huguet, Amine Natick, Kincaid Macdonald, Manik Kuchroo, Ronald Coifman, Guy Wolf, and Smita Krishnaswamy. 2021. “Diffusion Earth Mover’s Distance and Distribution Embeddings.” In *Proceedings of the 38th International Conference on Machine Learning*, 10336–46. PMLR. <https://proceedings.mlr.press/v139/tong21a.html>.
- Tong, Alexander, Frederick Wenkel, Kincaid Macdonald, Smita Krishnaswamy, and Guy Wolf. 2021. “Data-Driven Learning of Geometric Scattering Modules for Gnns.” In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6. IEEE. <https://ieeexplore.ieee.org/abstract/document/9596169/>.