

Tangent Blow-Ups for Processing Non-Manifold Geometry

Alice Petrov¹ Mohammad Sina Nabizadeh¹ Ana Dodik¹ Justin Solomon¹
¹MIT CSAIL, Cambridge, USA

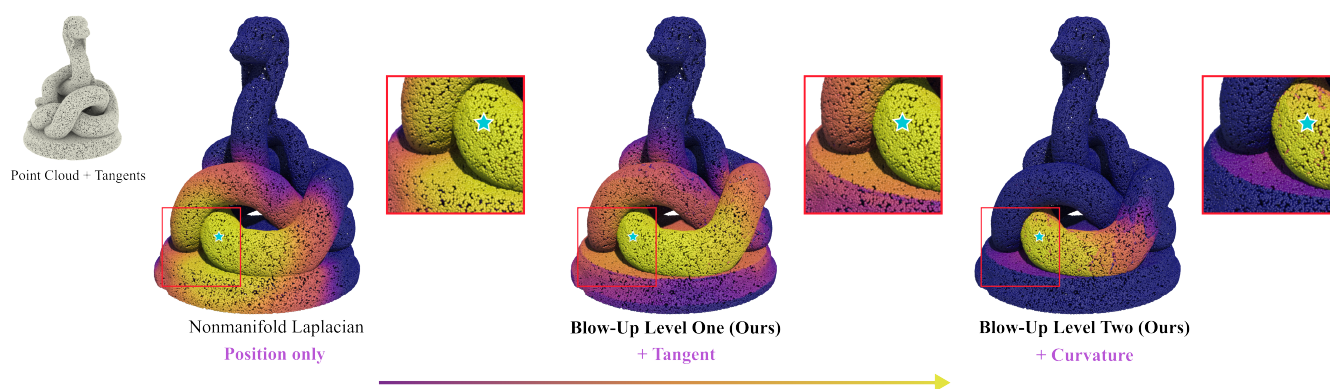


Figure 1: In the figure, different metrics give different patterns of heat diffusion. At singularities and intersections, standard methods (left) [SC20] treat all geometric components equally. Our tangent blow-up representation enriches each point with geometric information such as tangents (middle) and curvature (right) by iteratively lifting data into a higher-dimensional space where singularities are resolved and distances can better reflect the underlying geometry. This construction enables manifold-based point cloud processing pipelines to extend to non-manifold and singular shapes.

Abstract

Many geometry processing pipelines implicitly assume their input data is a manifold, or is sampled from one, with a unique tangent plane at every point. Geometric data, however, routinely contains sharp features like edges, corners, self-intersections, branching junctions, and other singularities, rendering standard methods ill-defined at these points. To bring geometry processing to these and other singular spaces, we introduce the “tangent blow-up,” a representation inspired by algebraic geometry that restores structure at singularities by lifting to the product of the ambient space and the Grassmannian of tangent planes. After iterating this construction, points that coincide in position but differ in tangent direction, curvature, or higher-order contact become well-separated. We equip the tangent blow-up with a product metric and define discretized differential operators, such as the gradient, divergence, and Laplacian, directly in the lifted domain. We demonstrate our framework across geodesic computation, segmentation, surface parameterization, and curvature estimation.

1. Introduction

The discretization of differential operators on point clouds typically relies on building local neighborhoods where the surface is well-approximated by a single tangent plane [HDD*92; VCD*16; RS22]. For example, the point cloud Laplacian constructs adjacency graphs from spatially-nearby points [CL06; BSW09], and curvature estimators like polynomial jet fitting locally fit truncated Taylor expansions to the surface [CP05]. The unifying assumption behind these methods is that data lies on or near a manifold. Non-manifold structures, however, abound in real-world data. For example, in geological modeling, the geometry and mechanical interac-

tions at intersecting fault junctions dictate stress accumulation and slip distributions during seismic events [MCO08].

Standard geometry processing algorithms can be applied to non-manifold data, but their output at points with multiple intersecting tangent planes—so-called *singular points*—typically does not respect the underlying geometry. For example, in Figure 1, the non-manifold Laplacian [SC20] diffuses heat across intersecting surfaces. More generally, any algorithm that builds neighborhoods from Euclidean distance alone will inevitably mix points from geometrically unrelated components. Hence, the question we address in this work is: *how can differential operators be defined on point*

clouds whose underlying geometry contains singularities, without requiring singularity detection or manifold repair as a preprocessing step?

Few computational approaches exist for directly handling non-manifold data. Existing techniques either detect and resolve singularities during preprocessing, as in non-manifold mesh repair [CBK21], or estimate topological invariants while bypassing discretization of differential operators, as in persistent intersection homology [BH11]. One can attempt to stratify a given dataset into clusters, each of which is well-approximated by a suitable submanifold of the ambient space [STHN20; AB24], a procedure known as *stratification learning* [BMW10; Nan20]. However, such decompositions can alter the topology of the space. For example, the lemniscate (figure-eight) is a classic example of an immersed curve that is not an embedding, and decomposing it into two manifold pieces destroys the global connectivity of the self-intersecting loop (see Figure 2).

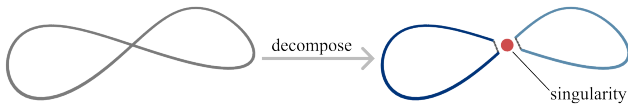


Figure 2: Decomposing a figure-eight curve into manifold pieces destroys the global connectivity of the self-intersecting loop.

The core difficulty at a singularity is that the tangent plane is not unique, and it is precisely the tangent plane that many algorithms use to define local neighborhoods, differential operators, and curvature. In the continuous realm, the *Nash blow-up* [Nob75; Hir83] in algebraic geometry resolves singularities in algebraic varieties by lifting points into the product space of positions and tangent planes (see Figure 4). In the resulting space, transversely intersecting points that coincide spatially are separated by their tangent-plane coordinates. Unlike more heuristic approaches, the Nash blow-up resolves singularities by embedding points into regular manifolds in a higher-dimensional space.

We bring this continuous construction into the discrete, computational setting of point cloud geometry processing. We equip the product space of positions and tangent planes with a metric that admits an isometric embedding into Euclidean space. This embedding reduces the entire blow-up pipeline to a coordinate transformation, after which Euclidean constructions such as k -nearest-neighbor graphs and Gaussian kernels apply directly, enabling the spectral and curvature analyses shown in Figure 3. We further show that, since the output has the same structure as the input, the blow-up can be iterated. Each iteration resolves tangential intersections of progressively higher contact order (see Figure 9). To justify the discrete pipeline, we analyze the continuous lifted space from which we assume points are sampled. We prove a *separation theorem*: the lifted metric enforces a lower bound between points with distinct tangents and a *smoothness theorem*: the lifted embedding of a smooth manifold is itself smooth. These results guarantee before sampling that transversely intersecting components are fully resolved into smooth, well-separated manifolds in the lifted space. Since we sample point clouds from this lifted space, the separation bound transfers directly to the embedded points (Corollary 1), and

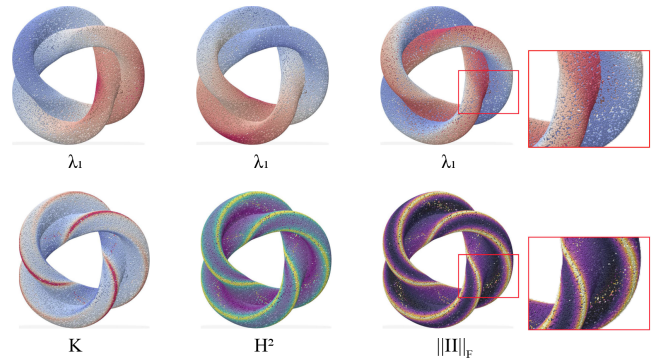


Figure 3: Spectral and curvature analysis of Banchoff's Klein bottle, a self-intersecting nonorientable surface [Ban76; Bus12]. Top row: the three smallest non-trivial eigenvectors of our lifted Laplacian (§6.1). Bottom row: Gaussian curvature, squared mean curvature, and the Frobenius norm of the second fundamental form, all computed from the unoriented projector field (§7.3)

standard convergence results for point cloud Laplacians [BN06; BSW09] apply to each manifold component individually. We address nontransverse intersections in §4.2.

Our contributions are:

- A discrete tangent blow-up representation for lifting point clouds that enables discretized differential operators to be well-defined directly on non-manifold geometry without singularity detection or manifold repair.
- An iterated blow-up that resolves tangential intersections of progressively higher contact order. We demonstrate that the second level separates components that agree to first order but differ in second-order geometry.
- Separation and smoothness theorems for the continuous lifted space that justify the discrete construction. For transverse intersections, we prove a distance lower bound proportional to the angular gap between tangent planes and show that the lifted embedding of a smooth manifold is also smooth.
- Discretized Laplacian, gradient, and divergence operators defined directly on the lifted space.
- Experimental evaluation of segmentation, geodesic approximation, and curvature estimation on non-manifold point clouds and validation that the blow-up yields correct curvature estimates on parametric immersions where existing methods fail near singularities.

2. Related Work

Nash blow-ups in algebraic geometry. The Nash blow-up pairs each point of a surface with its tangent plane and takes the closure, filling in limiting tangent planes over the singularities. Introduced by Semple [Sem54] and Nash [Nas96] and studied by Nobile [Nob75], Gonzalez-Sprinberg [Gon77], and Hironaka [Hir83], the construction was originally motivated by the question of whether iterated Nash blow-ups can resolve all singularities in algebraic varieties. For curves in characteristic zero, the answer is

affirmative [Nob75; Reb77]; in dimension four and higher, counterexamples exist [CDLL26]. This distinction is important for our setting since we do not claim that iterated tangent blow-ups resolve arbitrary non-smooth geometry. Instead, our work imports the tangent-separation idea of the Nash blow-up into geometry processing, where the goal is to define operators on point clouds.

Varifolds. A d -varifold is a (Radon) measure on $\mathbb{R}^n \times \text{Gr}(d, n)$, encoding a joint distribution of mass and unoriented tangent planes. The framework naturally accommodates singularities and tangent multiplicity, and has been used for surface approximation [BLM17], curvature estimation [BLM22], and mean curvature flow of point clouds [BR22]. Our lifted representation uses the same product space, but varifold methods define geometric quantities through integration against test forms, while our construction produces a Euclidean point cloud.

Feature-sensitive embeddings. Several geometry processing methods incorporate normals into distance computations to preserve geometric features. The isophotic metric of Pottmann et al. [PSH*04] weights curve lengths on a triangle mesh by the variation of surface normals. Lai et al. [LZH*06] embed mesh vertices and normals into \mathbb{R}^6 for feature classification and editing. Related constructions lift meshes into position-normal product spaces so that isotropic operations in the lifted space yield anisotropic or feature-aligned results in the original domain, with applications to remeshing [CG06; LB13]. These methods share with our work the idea of augmenting spatial coordinates with tangent or normal data, but they differ in scope and assumptions. They operate on manifold triangle meshes and assign a single oriented normal per vertex. On a non-manifold mesh, one vertex can have multiple normals, and the augmented embedding does not separate them.

Modified operators on manifolds. Bilateral filters suppress averaging across discontinuities by weighting distances using both spatial proximity and feature similarity [TM98; JDD03; FDC03]. Anisotropic diffusion operators [ARAC14; BMR*16] control the direction of information flow via position-dependent diffusion tensors on the tangent plane. These methods modify operators to incorporate geometric information, but they are task-specific and do not define a global geometric representation.

Singularity detection and resolution. In mesh processing, non-manifold singularities are typically resolved as a preprocessing step where edges and vertices are classified combinatorially, and the mesh is split or repaired before downstream algorithms [ZGZJ16; VL25; CBK21; ACK13]. Sharp and Crane [SC20] define an intrinsic Delaunay triangulation that handles non-manifold edges directly and yields a well-defined Laplacian without mesh repair, but their construction operates on mesh connectivity and does not use tangent plane information. For point clouds, recent work detects singularities via local topological or statistical tests [VR23; LON25], but these methods detect singularities without defining differential operators.

Curvature estimation. Point cloud curvature estimation typically proceeds by fitting a local model to the k -nearest neighbors. For example, jet fitting [CP05] uses truncated Taylor expansions, while

corrected curvature measures (CNC) [LRT22; LCL*23] integrate Lipschitz-Killing forms over randomly sampled triangles. Classical normal-difference approaches [Rus04; CLS*21] estimate curvature from how normals vary across mesh edges or over local neighborhoods of point clouds. These methods build neighborhoods in the ambient space \mathbb{R}^3 and assume that points are sampled from a smooth manifold.

Stratification learning. A separate line of work recovers the topological structure of stratified spaces from data, using tools such as persistent intersection homology [BH11], local cohomology and homology transfer [BWM12; Nan20], geometric anomaly detection [STHN20], subspace clustering [VMS05], and mixed-density estimation [HRS06]. These methods identify non-manifold structure but do not resolve singularities or discretize differential operators, which is the focus of our work. Most closely related to our work, Tinarrage [Tin23] estimates the tangent bundle of a sampled immersed manifold and applies persistent homology to the resulting lifted measure. This shares our principle of separating self-intersections, while aiming at topological inference from a single lift rather than the differential operators and iterated higher-order lifts developed here.

Point cloud reconstruction and processing. Neural implicit methods reconstruct surfaces from noisy point samples [EGO*20; MHLZ21]. Other recent reconstruction methods include Symmetrized Poisson Reconstruction [KLAK25], which encodes each sample by the outer product of its normal, and EdgeMovingNet [YJL*25], which jointly predicts edge features and normals to preserve sharp structures. Recent work has also produced increasingly accurate estimators for local differential quantities; we refer the reader to the survey of Arnal–Anger et al. [ALC*26]. Similarly, learning-based estimators such as PCPNet [GKOM18] and DeepFit [BG20] extract normals and higher-order quantities such as curvature from point clouds. However, nearly all the methods above rely on local neighborhoods in the ambient space or assume a manifold structure. Our construction is meant to enable existing algorithms, like those above, to apply to non-manifold data without additional priors or per-task training.

Normal estimation. Our discrete blow-up assumes that each sample carries at least one tangent plane, equivalently an unoriented normal direction for surfaces in \mathbb{R}^3 . Robustly estimating normals from raw point clouds is itself a substantial problem. Classical approaches include local PCA [KL97], while recent learned estimators improve robustness to noise, outliers, density variation, and orientation ambiguity [LLC*22; LHW*25]. These methods extract tangent information and are therefore complementary to our work, with better estimation improving the lifted representation directly. We rely on this in our evaluation: Figure 13 uses HSurfNet [LLC*22] normals on real indoor scans, while Figures 23 and 24 use local PCA tangents under controlled noise.

3. Mathematical Preliminaries

At a singular point, multiple tangent planes coexist at the same position. The tangent blow-up resolves this ambiguity by recording

which tangent plane is associated to each point. Rather than representing a sample by its position $x \in \mathbb{R}^n$ alone, we represent it by pairing each position with its tangent plane. Singular points that coincide in space but have different tangents become multiple, distinct points in the lifted representation. For example, at the transverse self-intersection of a figure-eight, the singular point x_0 is replaced by two lifted points, $(x_0, T_1 \nearrow)$ and $(x_0, T_2 \searrow)$ (see Figure 4).

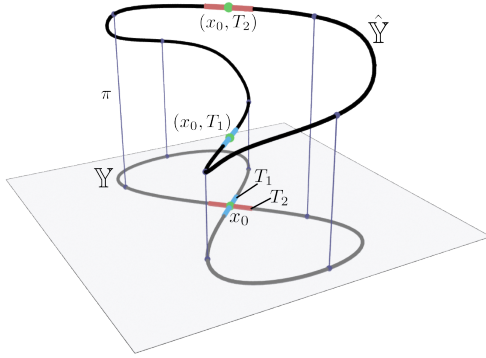


Figure 4: A schematic cartoon illustrating the tangent blow-up of a figure-eight curve. In the base space, the curve crosses itself at x_0 . After lifting to the product space $\mathbb{R}^n \times \text{Gr}(d, n)$, the singular point is replaced by two lifted points $(x_0, T_1 \nearrow)$ and $(x_0, T_2 \searrow)$. The projection π (defined in §3.2) is an isomorphism everywhere away from the set of singular points.

3.1. Continuous Construction

We begin by making our continuous construction precise. We first define the class of singular spaces we assume our point clouds are sampled from, then introduce the Grassmannian of tangent planes together with a suitable computational representation for its elements via *projector* matrices, and finally define the Nash blow-up. Because every object defined here admits a concrete matrix representation, the blow-up procedure ultimately reduces to concatenating position coordinates with the vectorized projector matrices. This allows for simplified linear algebra and avoids the need for orienting global normal fields.

Stratified spaces. Recall that a smooth d -dimensional manifold is a space that locally resembles \mathbb{R}^d at every point [Lee03]. Stratified spaces generalize this definition to address singular spaces. Informally, a stratified space \mathbb{Y} is a topological space that can be decomposed into manifold pieces meeting along lower-dimensional singular sets (see Figure 5); we refer the reader to Goresky and MacPherson [GM88] or Friedman [Fri20] for formal treatments. This is a natural setting for geometry processing since many “real-world singularities,” such as intersecting components of CAD models, produce this structure. We refer to the union of d -dimensional manifold pieces as the *regular part* \mathbb{Y}_{reg} and assume each manifold piece of \mathbb{Y}_{reg} is smooth.

The Grassmannian and its projector representation. To represent and compute over (point, tangent) pairs, we need a space that

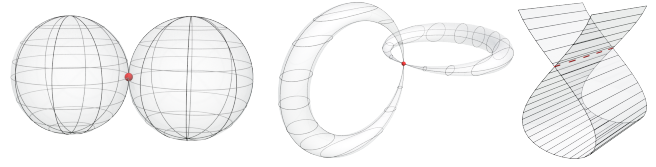


Figure 5: Examples of stratified spaces.

parametrizes all possible tangent planes. This is the *Grassmannian* $\text{Gr}(d, n)$, the compact smooth manifold of d -dimensional linear subspaces of \mathbb{R}^n [Lee03]. The Grassmannian admits multiple concrete computational representations [BZA24]. We opt to represent each element of $\text{Gr}(d, n)$ by its orthogonal projector $P = UU^T \in \mathbb{R}^{n \times n}$, where $U \in \mathbb{R}^{n \times d}$ is any orthonormal basis of the subspace. The projector is independent of the choice of basis, and allows one to map $\text{Gr}(d, n)$ injectively into the space of symmetric matrices. We also define the vectorization operator $\text{vec} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n^2}$, which stacks the columns of a matrix into a single vector. With this, distances between subspaces become Frobenius distances between projector matrices, which after vectorization become ordinary Euclidean distances [GV13].

The Nash blow-up. With the Grassmannian and its projector representation, we can now define the Nash blow-up. The idea is to pair each non-singular point with its tangent plane and take the closure to fill in the singular points.

Let \mathbb{Y} be a stratified space. On the regular part, each point $x \in \mathbb{Y}_{\text{reg}}$ has a unique tangent d -plane $T_x \mathbb{Y} \in \text{Gr}(d, n)$. The *generalized Gauss map*

$$v : \mathbb{Y}_{\text{reg}} \rightarrow \mathbb{R}^n \times \text{Gr}(d, n), \quad v(x) = (x, T_x \mathbb{Y}),$$

sends each regular point to the pair of its position and its tangent plane, embedding \mathbb{Y}_{reg} into the product of the ambient space with the Grassmannian (see Figure 6). The *Nash blow-up* of \mathbb{Y} is the closure of this image, which introduces limit points of convergent sequences on \mathbb{Y} :

$$\widehat{\mathbb{Y}} = \overline{\{(x, T_x \mathbb{Y}) \mid x \in \mathbb{Y}_{\text{reg}}\}} \subset \mathbb{R}^n \times \text{Gr}(d, n).$$

Over the regular part, $\widehat{\mathbb{Y}}$ is the graph of the tangent-plane assignment. Over the set of singular points, however, something more interesting happens when we take the closure: by definition, a point $(x_0, P) \in \widehat{\mathbb{Y}}$ with $x_0 \in \Sigma$ exists if and only if P is the limit of tangent planes along some sequence of regular points converging to x_0 . In other words, the closure fills in *limiting* tangent planes at each singularity. Returning to the figure-eight example (Figure 4), the crossing point x_0 acquires two lifted copies (x_0, T_1) and (x_0, T_2) .

The projection $\pi : \widehat{\mathbb{Y}} \rightarrow \mathbb{Y}$, $\pi(x, P) = x$, maps the blown-up space back to the original by “forgetting” the tangent plane. This map is continuous, surjective, and restricts to a homeomorphism over \mathbb{Y}_{reg} , so the blow-up modifies \mathbb{Y} only over the singular points. The preimage $\pi^{-1}(x_0)$ of a singular point is its *exceptional fiber*. Each point in the exceptional fiber corresponds to a distinct limiting tangent plane; we call these branches *sheets*. For example, the figure-eight has two sheets at its crossing point, each carrying its own tangent.

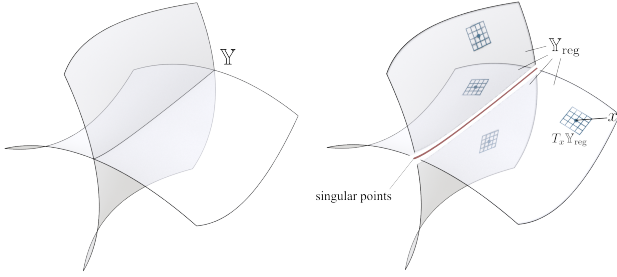


Figure 6: Left: A stratified space consisting of four smooth manifold sheets intersecting along singular points. Right: The space \mathbb{Y} is decomposed into its regular part \mathbb{Y}_{reg} and singular set of points (shown in red). Each smooth sheet is then lifted via the generalized Gauss map, associating to each regular point its tangent space.

3.2. Iterated Blow-Ups

The blow-up, as presented thus far, only separates transversely intersecting points with distinct tangent planes. This construction can be extended beyond transverse intersections to resolve higher-order contact phenomena, including osculating structures, under suitable conditions. The extension is achieved by iterating the blow-up, lifting successively from the previously lifted space. While transverse intersections are common in, e.g., messy geometry, iterated blow-ups are practically useful for separating out point clouds of multiple touching or proximate objects (see Figure 1).

Projector derivative. Iterating upon the first blow-up necessitates formally defining tangents at each lifted point (x, P) . Because our blow-up is a product space, we know that the tangent space at (x, P) decomposes into a direct sum of the tangent space at $x \in \mathbb{R}^n$, the so-called *horizontal* component, and the tangent space at $P \in \text{Gr}(d, n)$, the *vertical* component:

$$T_{(x,P)}(\mathbb{R}^n \times \text{Gr}(d,n)) = \underbrace{T_x \mathbb{R}^n}_{\text{horizontal}} \oplus \underbrace{T_P \text{Gr}(d,n)}_{\text{vertical}}.$$

The projector field $x \mapsto P$ assigns to each $x \in \mathbb{Y}_{\text{reg}} \subset \mathbb{R}^n$ the orthogonal projector onto the tangent plane $T_x \mathbb{Y}$. For a curve $c(t)$ on \mathbb{Y}_{reg} with $c(0) = x$ and $c'(0) = v \in T_x \mathbb{Y}$, the lifted curve $t \mapsto (c(t), P(c(t)))$ has tangent

$$\left. \frac{d}{dt}(c(t), P(c(t))) \right|_{t=0} = (v, \nabla_v P),$$

where $\mathcal{P}_v := \nabla_v P \in T_P \text{Gr}(d,n)$ denotes the vertical component of the lifted tangent space. Differentiating the idempotency condition $P^2 = P$ along v gives $\mathcal{P}_v P + P \mathcal{P}_v = \mathcal{P}_v$, from which it follows that

$$P \mathcal{P}_v P = 0 \quad \text{and} \quad (I - P) \mathcal{P}_v (I - P) = 0.$$

This means \mathcal{P}_v has no tangent-to-tangent or normal-to-normal component, so it maps tangent vectors into the normal space and normal vectors into the tangent space. Indeed, any $n \times n$ matrix satisfying these two conditions is a tangent vector at $P \in \text{Gr}(d,n)$ [Bou23; BZA24].

Curvature. As a point moves along the surface, its tangent plane rotates, and the rate of this rotation encodes curvature. The projector derivative is therefore naturally related to the curvature of \mathbb{Y}_{reg} . Specifically, it represents the normal component of the differential change in tangent vector w in direction v , i.e., it is the same as the vector-valued second fundamental form, $\Pi(v, w) = \mathcal{P}_v(w)$, which is a generalization of the second fundamental form to arbitrary codimension [RS22] (see Figure 7). The *scalar* valued second fundamental form, more commonly used in geometry processing, picks a unit normal ℓ and returns a number $\Pi(v, w) = \langle \mathcal{P}_v(w), \ell \rangle \in \mathbb{R}$ rather than a vector in the normal space.

The projector derivative \mathcal{P}_v can be represented as an $n \times n$ matrix, but its subspace structure (tangent-to-normal and normal-to-tangent only) means it is fully determined by the $(n-d) \times d$ matrix

$$B_x(v) = N^\top \mathcal{P}_v U \in \mathbb{R}^{(n-d) \times d},$$

where $U \in \mathbb{R}^{n \times d}$ and $N \in \mathbb{R}^{n \times (n-d)}$ are orthonormal bases for the tangent and normal spaces at x [BZA24]. Each entry $B_x(v)_{\ell k}$ records how much the k -th tangent basis vector rotates into the ℓ -th normal direction when the basepoint moves along v . The (vector valued) second fundamental form is recovered by mapping these coordinates back to ambient space: $\Pi(v, w) = N B_x(v) (U^\top w)$.

For hypersurfaces ($n-d=1$), the normal space is one-dimensional and $B_x(v)$ reduces to a row vector for any tangent direction v . Stacking the rows $B_x(e_1), \dots, B_x(e_d)$ retrieves the shape operator, whose eigenvalues are the principal curvatures $\kappa_1, \dots, \kappa_d$.

Computation. Computing $\mathcal{P}_v \in \mathbb{R}^{n \times n}$ directly requires differentiating the projector field. In Section 4.2 we show how to estimate the compact representation B_x from a point cloud by linear regression, after which the full projector derivative can be reconstructed as

$$\mathcal{P}_v = N B_x(v) U^\top + U B_x(v)^\top N^\top.$$

While $B_x(v)$ depends on the choice of tangent and normal bases U and N , the reconstructed \mathcal{P}_v does not [BZA24].

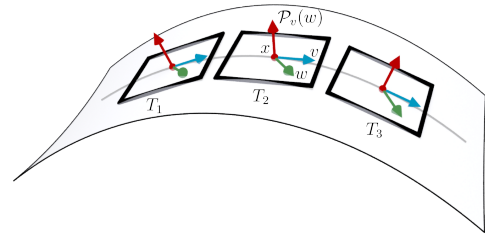


Figure 7: The projector derivative on a smooth surface. As the basepoint moves along the curve with tangent v (blue arrow), the tangent plane rotates from T_1 to T_2 to T_3 . At x_1 , the projector derivative $\mathcal{P}_v(w)$ (red arrow) measures the component of the tangent vector w (green arrow) that leaves the tangent plane, which is the vector valued second fundamental form $\Pi(v, w)$.

4. The Discrete Tangent Blow-Up

We now present our main contribution: a discrete representation that makes the blow-up algorithmic and applicable to point cloud

data. We begin by defining a metric on the product space $\mathbb{R}^n \times \text{Gr}(d, n)$ and its Euclidean embedding, then describe the discrete lifting algorithm that takes a set of points and tangents as input and produces a point cloud with resolved singularities in the lifted space as output.

Lifted metric. All discretized operators on the lifted point cloud, such as the Laplacian (§6.1), gradient (§6.2), and the second fundamental form regression (§4.2), require a notion of neighborhood and proximity. To perform nearest-neighbor search and kernel evaluation in the product space $\mathcal{L} = \mathbb{R}^n \times \text{Gr}(d, n)$, we define a metric that combines spatial and tangential components. The *chordal distance* on the Grassmannian,

$$d_{\text{chord}}(U, V) = \frac{1}{\sqrt{2}} \|P_U - P_V\|_F = \left(\sum_{i=1}^d \sin^2 \theta_i \right)^{1/2},$$

where $\theta_1, \dots, \theta_d$ are the principal angles between $U, V \in \text{Gr}(d, n)$, is computationally efficient, and its square is smooth everywhere [CHS96]. Furthermore, it approximates the geodesic distance on $\text{Gr}(d, n)$ with cubic error as distances approach zero, $d_{\text{chord}} = d_{\text{geo}} + O(d_{\text{geo}}^3)$, as follows directly from the Taylor expansion of the principal-angle characterizations of both metrics [EAS98; BZA24].

We define the *lifted metric* on $\mathbb{R}^n \times \text{Gr}(d, n)$ as the weighted product

$$d_{\mathcal{L}}^2((p, U), (q, V)) = \|p - q\|^2 + \frac{\alpha}{2} \|P_U - P_V\|_F^2, \quad (1)$$

where $\alpha > 0$ is a parameter balancing the spatial and angular parts. Since the lifted metric measures both spatial displacement and tangent-plane variation, $d_{\mathcal{L}}$ changes distances by stretching regions where curvature causes tangent planes to vary rapidly. Geodesics and differential quantities computed after lifting should therefore be interpreted as quantities of the lifted, curvature-sensitive geometry, even for a smooth embedded manifold.

Isometric Euclidean embedding. A useful computational property of the chordal product metric is that it admits an isometric embedding into Euclidean space. Define the embedding map

$$\Phi: \mathbb{R}^n \times \text{Gr}(d, n) \rightarrow \mathbb{R}^{n+n^2}, \quad \Phi(p, U) = (p, \sqrt{\alpha/2} \text{vec}(P_U)),$$

where $\text{vec}(\cdot)$ denotes vectorization. Then $\|\Phi(p, U) - \Phi(q, V)\|^2 = d_{\mathcal{L}}^2((p, U), (q, V))$. Thus, Euclidean distance in the embedding space exactly reproduces the lifted metric. Consequently, computing neighborhoods and kernels in the lifted metric reduces to applying ordinary Euclidean algorithms to the coordinates $\Phi(p, U)$.

4.1. Algorithm

Input. The input to our discrete blow-up algorithm is a point cloud equipped with tangent planes: positions $\{x_i\}_{i=1}^N \subset \mathbb{R}^n$, orthonormal tangent frames $\{U_i\}_{i=1}^N$ with $U_i \in \mathbb{R}^{n \times d}$ spanning the tangent d -plane at point x_i with index i , and a weight parameter $\alpha > 0$ used in our metric in (1). For surfaces in \mathbb{R}^3 , a tangent plane is equivalent to a normal vector, though no globally consistent orientation is required. For curves, a unit tangent vector suffices. We assume that tangent planes are provided; the practical problem of tangent estimation near singularities is discussed in §8.

The lift. For each sample, we compute the tangent-plane projector $P_i = U_i U_i^\top \in \mathbb{R}^{n \times n}$, the orthogonal projection onto $\text{col}(U_i)$. The projector depends only on the subspace spanned by U_i , not on the choice of basis. Consequently any quantity derived from P , such as distances between projectors and derivatives of the projector field, is also coordinate-free. We then apply the isometric embedding:

$$\Phi_i = (x_i, \sqrt{\alpha/2} \text{vec}(P_i)) \in \mathbb{R}^{n+n^2}.$$

The output is a Euclidean point cloud $\{\Phi_1, \dots, \Phi_N\} \subset \mathbb{R}^{n+n^2}$ on which Euclidean distances reproduce the lifted metric by construction:

$$\|\Phi_i - \Phi_j\|^2 = \|x_i - x_j\|^2 + \frac{\alpha}{2} \|P_i - P_j\|_F^2 = d_{\mathcal{L}}^2((x_i, P_i), (x_j, P_j)).$$

The coordinate transformation is applied uniformly to every sample point; no knowledge of singular points or any global topology is required.

To understand how the lifting separates intersecting sheets, consider two points x_i and x_j near a transverse self-intersection. In the ambient space \mathbb{R}^n , these points may be arbitrarily close. In the lifted space, however, the projector term $\frac{\alpha}{2} \|P_i - P_j\|_F^2$ contributes a penalty proportional to the angular gap between their tangent planes. Larger α widens the gap at the cost of stretching distances along curved regions where the tangent plane varies rapidly. Figure 8 illustrates this effect on a self-intersecting figure-eight curve, where the projector entries vary smoothly along each branch but differ sharply at the crossing; we offer a formal treatment in §5.

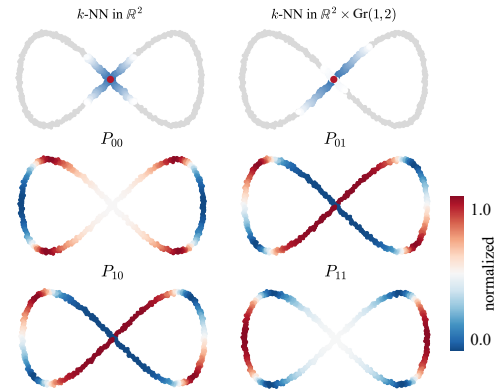


Figure 8: The discrete tangent blow-up of a self-intersecting figure-eight curve. Top row: k -nearest neighbors of a query point (red), computed in the ambient space \mathbb{R}^2 (left) and in the lifted embedding $\mathbb{R}^2 \times \text{Gr}(1, 2)$ (right). Bottom rows: entries of the tangent projector $P \in \text{Gr}(1, 2)$, shown component-wise over the curve. Points at the crossing share spatial position but have distinct tangents, yielding different values of P .

Remark 1 (The exceptional fiber in the discrete setting.) At a singular point x_0 , the continuous blow-up $\hat{\mathbb{Y}}$ has a well-defined exceptional fiber (defined in §3.2). In the discrete setting, we do not explicitly compute a closure, but the fiber structure is inherited by the samples. A point x_i sampled near a singularity x_0 on a sheet M carries a tangent estimate $P_i \approx T_{x_0}M$, so its lifted image Φ_i lies near the fiber point $(x_0, T_{x_0}M)$. As the sampling density increases, the

lifted point cloud approximates this structure without any explicit closure.

4.2. The Discrete Iterated Blow-Up

The first lift separates sheets with distinct tangent planes. However, multiple sheets can share the same tangent at a point and still be geometrically distinct. For example, a line and a parabola meeting tangentially at the origin have the same position and tangent, but different curvature (see Figure 9). To resolve such tangential intersections, we need to capture how the tangent plane varies from point to point.

The discrete blow-up can be applied iteratively: the output of the first lift is a Euclidean point cloud $\{\Phi_i\}$, and the tangent projector at each lifted point is itself a point on a new Grassmannian. Hence, we can define a new product space to which the entire blow-up machinery applies again. To perform this second lift, we need the tangent planes of the lifted product space $\mathbb{R}^n \times \text{Gr}(d, n)$, which requires estimating how P_i varies across the point cloud (recall §3.2). This variation is encoded by the vector valued second fundamental form.

Discrete estimation of the second fundamental form. Although the lifted point cloud lives in \mathbb{R}^{n+n^2} , the second fundamental form can be estimated from the original data (x_i, U_i) alone, without performing PCA or tangent estimation in the high-dimensional lifted space. This is because the vertical component of the product tangent vector is determined entirely by the projector derivative, which depends only on n -dimensional positions and d -dimensional tangent frames.

Broadly speaking, we approximate the vector valued second fundamental form \mathcal{P}_v at each sample x_i by finite-differencing the projector P_i against neighboring projectors, and recovering a discrete estimate of $B_i \approx B_{x_i}$ by linear regression. Thus, the procedure to estimate B_i is as follows. For each point (x_i, U_i) , fix an orthonormal basis $N_i \in \mathbb{R}^{n \times (n-d)}$ for $\text{col}(U_i)^\perp$. Throughout, $a, b = 1, \dots, d$ index tangent directions with e_a the standard basis of \mathbb{R}^d , and $\ell = 1, \dots, n-d$ indexes normal directions. Then:

1. Find k -nearest neighbors of Φ_i in the lifted metric.
2. Compute intrinsic displacements $t_{ij} = U_i^\top (x_j - x_i) \in \mathbb{R}^d$: the spatial offset projected into the tangent plane at x_i .
3. Compute Grassmannian tangent coordinates $C_{ij} = N_i^\top (P_j - P_i) U_i \in \mathbb{R}^{(n-d) \times d}$: the projector difference expressed in the tangent and normal bases at x_i . For nearby points on the same smooth sheet, $C_{ij} \approx B_{x_i}(t_{ij}) + O(\|x_j - x_i\|^2)$.
4. Solve the constrained least-squares problem:

$$B_i = \underset{B}{\text{argmin}} \sum_{j \in \mathcal{N}(i)} \|C_{ij} - B t_{ij}\|_F^2$$

subject to $(B(e_a))_{\ell b} = (B(e_b))_{\ell a}$ for each normal direction ℓ .

The symmetry constraint reflects the symmetry of the vector valued second fundamental form on smooth manifolds [Lee03]. The resulting $B_i \approx B_{x_i}$ is the discrete analogue of the matrix defined in §3.2, from which the full projector derivative is reconstructed as $\mathcal{P}_v = N_i B_i(v) U_i^\top + U_i B_i(v)^\top N_i^\top$ and all curvature invariants

(§7.3) are extracted. In the hypersurface case ($n-d=1$), N_i reduces to a single unit normal, $B_i(e_a) \in \mathbb{R}^{1 \times d}$ is a row vector for each tangent direction, and the symmetry constraint makes the $d \times d$ matrix with rows $B_i(e_1), \dots, B_i(e_d)$ symmetric. Hence, we recover the shape operator.

Constructing the second blow-up. Having estimated each B_i , we can construct the tangent space of the lifted manifold at each point. For each tangent basis direction $e_a \in \mathbb{R}^d$, the tangent vector to the first lift at (x_i, P_i) has a horizontal component (the spatial velocity $U_i e_a$) and a vertical component (the corresponding Grassmannian velocity, encoding how the projector changes when the basepoint moves along e_a).

The vertical component is reconstructed from its coordinate representation $B_i(e_a)$ via the tangent space characterization of the Grassmannian [BZA24]: any tangent vector to $\text{Gr}(d, n)$ at P_i takes the form $N_i B U_i^\top + U_i B^\top N_i^\top$ for some $B \in \mathbb{R}^{(n-d) \times d}$.

$$g_a = \left(U_i e_a, \sqrt{\alpha/2} \text{vec}(N_i B_i(e_a) U_i^\top + U_i B_i(e_a)^\top N_i^\top) \right) \in \mathbb{R}^{n+n^2}.$$

The vectors g_1, \dots, g_d span the tangent space of the lifted manifold at Φ_i , but may not in general be orthonormal.

To obtain a tangent projector, we stack them into $G_i = [g_1 \dots g_d] \in \mathbb{R}^{(n+n^2) \times d}$ and form

$$Q_i = G_i (G_i^\top G_i)^{-1} G_i^\top \in \text{Gr}(d, n+n^2).$$

Although the intermediate quantities B_i , g_a , and G_i depend on the choice of bases U_i and N_i , the projector Q_i does not. This is the same coordinate-independence that holds for P_i itself, as noted in §3.2.

The second lift applies the blow-up to its own output. After the first lift, each sample carries a position $\Phi_i^{(1)} \in \mathbb{R}^{D_1}$ (with $D_1 = n+n^2$) and a tangent projector $P_i^{(1)} = Q_i \in \text{Gr}(d, D_1)$, computed from the procedure above. The goal of the second lift is to produce a new tangent projector $P_i^{(2)} \in \text{Gr}(d, D_2)$ that encodes how $P_i^{(1)}$ itself varies across the point cloud. The second-lift embedding and its associated tangent projector are constructed by applying the same algorithm to the pair $(\Phi_i^{(1)}, P_i^{(1)})$ to obtain

$$\Phi_i^{(2)} = \left(\Phi_i^{(1)}, \sqrt{\alpha^{(2)}/2} \text{vec}(P_i^{(1)}) \right) \in \mathbb{R}^{D_2}, \quad D_2 = D_1 + D_1^2$$

where $\alpha^{(2)} > 0$ is the weight parameter for the second level. The second-lift distance is

$$d_{\mathcal{L}_2}^2 = \|\Phi_i^{(1)} - \Phi_j^{(1)}\|^2 + \frac{\alpha^{(2)}}{2} \|P_i^{(1)} - P_j^{(1)}\|_F^2$$

which penalizes differences in position, tangent plane, and curvature simultaneously. The procedure is summarized in Algorithm 1.

Remark 2 (Higher-order iteration) In principle, the discrete blow-up construction iterates indefinitely. Each level ℓ produces a new Euclidean point cloud with tangent projectors, and each successive level separates points that agree up to one higher order of contact: level zero distinguishes positions, level one distinguishes tangent planes, level two distinguishes curvatures, and so on. In practice, however, each iteration estimates a higher-order information from finite noisy samples, and the accuracy of these estimates

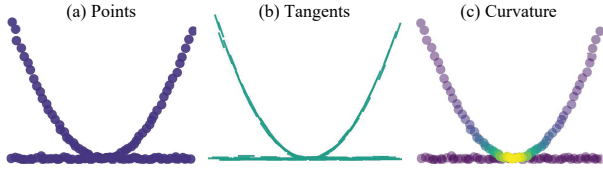


Figure 9: A line and parabola sharing position (left) and tangent (center) at the origin. The Frobenius norm of the vector valued second fundamental form $\|II\|_F$ distinguishes the intersection (right).

likely degrades with order. We conjecture that singularities of contact order r (sheets sharing the same “ r -jet” but differing at order $r + 1$) are resolved at exactly level r . In this paper, we restrict attention to levels 0 and 1, which handle transverse and tangential intersections respectively. Our method is designed for point clouds that are sampled from a finite union of smooth sheets and thus is not a general-purpose resolution method for arbitrary non-smooth sets. A systematic study of the relationship between iterated blow-ups, jet spaces, and higher-order differential invariants such as torsion is left to future work.

5. Theoretical Analysis

Assuming all intersections are transverse and singularities are thus resolved after one lift, we now consider whether the tangent blow-up achieves two properties needed for geometry processing. First, *separation*: points on distinct sheets that are spatially close must become metrically well-separated after lifting. Second, *smoothness*: the lifted image of a smooth manifold must itself be smooth.

We establish both properties in the continuous setting. Since we assume that the discrete construction is sampled from this continuous space, these results help justify the computational pipeline. They ensure that, after lifting, each sheet is a smooth submanifold of Euclidean space with positive separation from other sheets, which is the setting where standard convergence results for point cloud Laplacians [BN06; BSW09] are formulated.

Our first guarantee is that the lifted metric admits a lower bound between distinct transversely intersecting sheets, even as their spatial distance tends to zero.

Theorem 1 (Lifted Separation) Let $M_1, M_2 \subset \mathbb{R}^n$ be smooth d -dimensional submanifolds intersecting at a point x_0 with distinct tangent planes. Write P_{x_0, M_a} for the tangent projector of sheet M_a at x_0 , and let

$$\delta = d_{\text{chord}}(T_{x_0}M_1, T_{x_0}M_2) > 0$$

be the chordal distance. Then there exists $\epsilon_0 > 0$ (depending on δ and the curvatures of M_1, M_2 near x_0) such that for all $p \in M_1, q \in M_2$ with $\|p - x_0\|, \|q - x_0\| < \epsilon_0$,

$$d_{\mathcal{L}}(\tilde{p}, \tilde{q}) \geq \frac{1}{\sqrt{2}} \sqrt{\alpha} \delta > 0,$$

where $\tilde{p} = (p, T_p M_1)$ and $\tilde{q} = (q, T_q M_2)$ are the lifted points and $d_{\mathcal{L}}$ is the lifted metric of §4.

Algorithm 1 Discrete iterated blow-up (levels one and two)

Input: Positions $\{x_i\}_{i=1}^N \subset \mathbb{R}^n$, tangent frames $\{U_i\}_{i=1}^N$, weights $\alpha^{(1)}, \alpha^{(2)} > 0$, neighbors k

Output: Level-two embedding $\{\Phi_i^{(2)}\}_{i=1}^N \subset \mathbb{R}^{D_2}$, tangent projectors $\{P_i^{(1)}\}_{i=1}^N, \{P_i^{(2)}\}_{i=1}^N$

- 1: — **Level-one lift: embed into** $\mathbb{R}^{D_1}, D_1 = n + n^2$ —
- 2: **for** $i = 1, \dots, N$ **do**
- 3: $P_i^{(0)} \leftarrow U_i U_i^\top$ {tangent projector}
- 4: $\Phi_i^{(1)} \leftarrow (x_i, \sqrt{\alpha^{(1)}/2} \text{vec}(P_i^{(0)}))$ {isometric embedding}
- 5: **end for**
- 6: — **Curvature estimation and tangent projectors** —
- 7: **for** $i = 1, \dots, N$ **do**
- 8: $N_i \leftarrow \text{orth}(\ker(P_i^{(0)}))$ {normal basis}
- 9: $\mathcal{N}(i) \leftarrow k\text{-NN of } \Phi_i^{(1)} \text{ in } \mathbb{R}^{D_1}$ {lifted neighbors}
- 10: **for** $j \in \mathcal{N}(i)$ **do**
- 11: $t_{ij} \leftarrow U_i^\top (x_j - x_i)$ {tangent-plane displacement}
- 12: $C_{ij} \leftarrow N_i^\top (P_j^{(0)} - P_i^{(0)}) U_i$ {projector variation}
- 13: **end for**
- 14: $B_i^{(1)} \leftarrow \text{argmin}_{B: B_i^\top = B_i} \sum_j \|C_{ij} - B t_{ij}\|_F^2$ {fit second fundamental form}
- 15: **for** $a = 1, \dots, d$ **do**
- 16: $(\mathcal{P}e_a)_i \leftarrow N_i B_i^{(1)}(e_a) U_i^\top + U_i B_i^{(1)}(e_a)^\top N_i^\top$ {Grassmannian velocity}
- 17: $g_a \leftarrow (U_i e_a, \sqrt{\alpha^{(1)}/2} \text{vec}(\Delta_a))$ {product tangent vector}
- 18: **end for**
- 19: $G_i \leftarrow [g_1 \dots g_d]$
- 20: $P_i^{(1)} \leftarrow G_i (G_i^\top G_i)^{-1} G_i^\top$ {tangent projector of the lifted manifold}
- 21: **end for**
- 22: — **Level-two lift: embed into** $\mathbb{R}^{D_2}, D_2 = D_1 + D_1^2$ —
- 23: **for** $i = 1, \dots, N$ **do**
- 24: $\Phi_i^{(2)} \leftarrow (\Phi_i^{(1)}, \sqrt{\alpha^{(2)}/2} \text{vec}(P_i^{(1)}))$ {same structure as level one}
- 25: **end for**

The proof is given in Appendix A.

We validate the separation bound empirically in Figure 10. For the discrete pipeline, the theorem has the following consequence.

Corollary 1 (Discrete sheet separation) Let $\{(x_i, U_i)\}_{i=1}^N$ be a point cloud sampled from $M_1 \cup M_2$ with correct tangent planes, and let $\Phi_i = (x_i, \sqrt{\alpha/2} \text{vec}(P_i))$ be the lifted embedding. By Theorem 1, every pair of lifted points Φ_i, Φ_j with $x_i \in M_1, x_j \in M_2$, and $\|x_i - x_0\|, \|x_j - x_0\| < \epsilon_0$ satisfies

$$\|\Phi_i - \Phi_j\| \geq \frac{1}{\sqrt{2}} \sqrt{\alpha} \delta.$$

In particular, if k -nearest-neighbor queries in the lifted space use a search radius smaller than $\frac{1}{\sqrt{2}} \sqrt{\alpha} \delta$, no query point on one sheet will return a neighbor from the other sheet within distance ϵ_0 of the singularity.

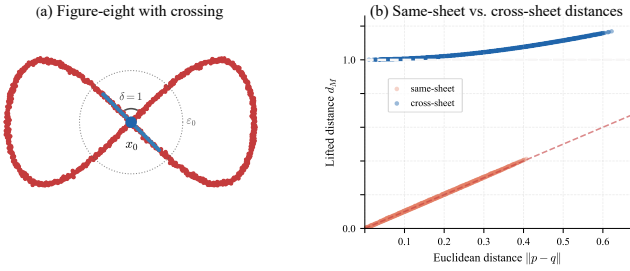


Figure 10: Same-sheet vs. cross-sheet lifted distances on a figure-eight curve. (a) The curve with crossing point marked. (b) Lifted distance $d_{\mathcal{L}}$ as a function of Euclidean distance $\|p - q\|$. Same-sheet pairs (red) vanish with spatial distance; cross-sheet pairs (blue) are bounded below by the separation floor $\frac{1}{\sqrt{2}}\sqrt{\alpha}\delta$ (dashed).

Proof By construction, $\|\Phi_i - \Phi_j\|^2 = d_{\mathcal{L}}^2(\tilde{x}_i, \tilde{x}_j)$, so the bound follows directly from Theorem 1. \square

Corollary 1 makes the separation theorem practical. Given a non-manifold configuration whose minimal tangent disparity δ is known (or conservatively estimated), one can choose $\alpha > 2\|\Phi_i - \Phi_j\|^2/\delta^2$ and guarantee that nearest-neighbor queries never cross sheet boundaries near the singularity.

The second guarantee is that the lifted image of each smooth sheet is regular enough to support the differential operators defined in §6.2 and preserves intrinsic dimension. The blow-up differentiates the Gauss map, so each iteration costs exactly one order of differentiability.

Theorem 2 (Regularity of the Lifted Manifold) Let $M \subset \mathbb{R}^n$ be a C^r submanifold of dimension d with $r \geq 2$. Then the lifted embedding $\Phi : M \rightarrow \mathbb{R}^{n+n^2}$, $\Phi(x) = (x, \sqrt{\alpha/2} \text{vec}(P_x))$, is a C^{r-1} embedding, and the lifted manifold $\widehat{M} = \Phi(M)$ is a C^{r-1} submanifold of dimension d .

The proof is given in Appendix A.

These results establish that, for transversely intersecting C^r manifolds, the blow-up produces a collection of well-separated, C^{r-1} -regular submanifolds in Euclidean space.

6. Lifted Differential Operators

To perform downstream geometry processing tasks directly on non-manifold point clouds, we now define examples of discretized differential operators on the tangent blow-up representation following standard meshless constructions. We construct a point cloud Laplacian for diffusion and spectral analysis (§6.1), gradient and divergence operators for vector field computations (§6.2), and define pointwise curvature invariants extracted from the second fundamental form (§6.3).

6.1. Laplacian

We model the lifted point cloud as a weighted graph by using a multi-scale product kernel derived from the lifted product metric

that decouples spatial and tangential components:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma_0^2}\right) \cdot \prod_{m=1}^M \exp\left(-\frac{\|P_i^{(m)} - P_j^{(m)}\|_F^2}{\sigma_m^2}\right),$$

where σ_0 is the spatial bandwidth and σ_m is the bandwidth for the m -th tangent projector given M blow-up iterations. Each bandwidth can either be fixed, or set to the median k -nearest-neighbor distance in the corresponding factor of the product metric [ZP04], so that the kernel adapts to the intrinsic scale of each component. Incorporating Grassmannian components results in geometrically distinct points with differing unoriented tangents having reduced affinity near singularities.

From the affinity W and degree matrix $\text{diag}(W\mathbf{1})$ we form either the standard unnormalized Laplacian $L = \text{diag}(W\mathbf{1}) - W$ or its symmetric normalized variant $L_{\text{sym}} = I - \text{diag}(W\mathbf{1})^{-1/2}W \text{diag}(W\mathbf{1})^{-1/2}$ [CL06; Von07]. Recall from §5 that transversely intersecting sheets are pulled apart in the lifted embedding by their tangent spaces, while each smooth sheet remains smooth under the lift. Thus, if the point cloud sample is drawn from s smooth components whose pairwise tangent angles exceed the kernel bandwidth, for sufficiently fine sampling the lifted affinity is close to block diagonal, with one block per component. Standard perturbation theory for block-diagonal symmetric matrices [Von07] yields exactly s near-zero eigenvalues whose corresponding eigenvectors approximate indicators of the individual components, which are then recovered via the spectral clustering pipeline of §7.2.

6.2. Gradient and Divergence

We construct standalone gradient and divergence operators using a standard weighted moving least squares (MLS) construction on the lifted point cloud [Nea04; LZ13; WNEH22]. Both operators reuse the self-tuning product-kernel weights W_{ij} from the affinity matrix W defined in §6.1. At each point Φ_i the tangent projector $Q_i = R_i R_i^T$ provides an orthonormal tangent frame $R_i \in \mathbb{R}^{D \times d}$, where $D = n + n^2$ is the ambient dimension of the first lift and d is the rank of the tangent projector Q_i . For a second-level lift, Φ_i , R_i , and Q_i are replaced by their level-two counterparts from Algorithm 1, with ambient dimension $D_2 = D_1 + D_1^2$.

Discrete gradient. Let $f_i = f(\Phi_i)$ be a scalar function on the lifted point cloud. We estimate $\text{grad } f_i$ by fitting a degree-one polynomial in the tangent plane at point Φ_i . First, project each neighbor displacement into the local tangent frame, $\tilde{\delta}_{ij} = R_i^T(\Phi_j - \Phi_i) \in \mathbb{R}^d$, and then form the $d \times d$ weighted moment matrix

$$S_i = \sum_{j \in \mathcal{N}(i)} W_{ij} \tilde{\delta}_{ij} \tilde{\delta}_{ij}^T.$$

Substituting $f(\Phi_j) \approx f_i + \tilde{g}^T \tilde{\delta}_{ij}$ and minimizing $\sum_j W_{ij} \|\tilde{g}^T \tilde{\delta}_{ij} - (f_j - f_i)\|^2$ over $\tilde{g} \in \mathbb{R}^d$ yields

$$\tilde{g}_i = S_i^{-1} \sum_{j \in \mathcal{N}(i)} W_{ij} \tilde{\delta}_{ij} (f_j - f_i).$$

Lifting back to the ambient space gives the estimated gradient: $(\text{grad}_{\widehat{\mathcal{V}}} f)_i = R_i \tilde{g}_i \in \mathbb{R}^D$.

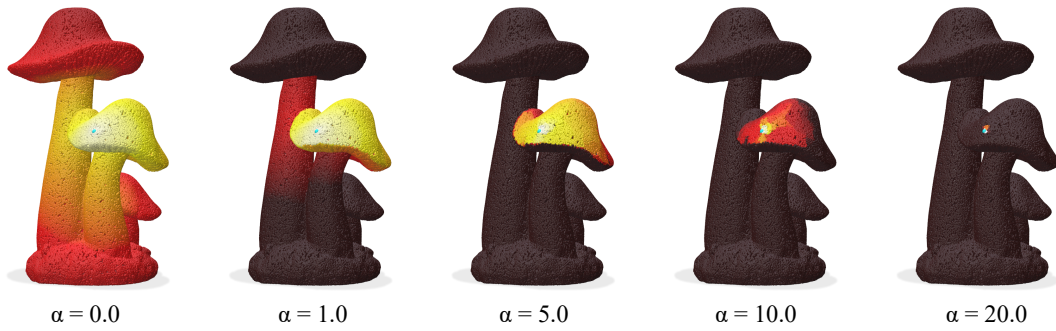


Figure 11: Ablation over the projector weight α on the first lift of a mushroom point cloud [ZJ16]. Heat is placed at a single source point on the middle cap. At $\alpha = 0$ the kernel ignores tangents entirely and heat bleeds across all sheets; as α grows, diffusion remains on each smooth sheet. At higher values of α , diffusion becomes highly sensitive to underlying tangents.

Discrete divergence. Given a tangent vector field $X_i \in \text{col}(R_i)$ on the lifted manifold, we estimate its divergence at vertex i by regressing changes in X against displacements, with all quantities expressed in the local tangent frame R_i . Concretely:

1. Project the vector field at i and at each neighbor j into point i 's frame: $\bar{X}_k^{(i)} = R_i^\top X_k$ for $k \in i \cup \mathcal{N}(i)$.
2. Estimate the tangent-plane Jacobian by regressing the frame-consistent differences against projected displacements:

$$J_i^a = \sum_{j \in \mathcal{N}(i)} W_{ij} \bar{\delta}_{ij} (\bar{X}_j^{(i)} - \bar{X}_i^{(i)})^a \in \mathbb{R}^d.$$

3. Take the trace: $(\text{div}_{\mathbb{V}} X)_i = \text{tr}(S_i^{-1} J_i)$, where $J_i = [J_i^1; \dots; J_i^d]$.

6.3. Curvature

Having estimated $B_i \approx B_{x_i}$ at each sample (§4.2), we now extract classical curvature invariants. We restrict to hypersurfaces ($n - d = 1$) in this section; the estimation of B_i applies in any codimension, but the invariants extracted differ.

For a hypersurface, N_i is a single unit normal and $B_i(e_a) \in \mathbb{R}^{1 \times d}$ is a row vector. Stacking the rows gives the $d \times d$ symmetric matrix $S_i = [B_i(e_1); \dots; B_i(e_d)]$, which is the shape operator in the tangent basis U_i . Its eigenvalues are the principal curvatures $\kappa_1, \dots, \kappa_d$, from which we compute:

$$\begin{aligned} K &= \prod_a \kappa_a = \det(S_i), \\ H &= \frac{1}{d} \sum_a \kappa_a = \frac{1}{d} \text{tr}(S_i), \\ \|\Pi\|_F &= \sqrt{\sum_a \kappa_a^2} = \|S_i\|_F. \end{aligned}$$

Although S_i depends on the choice of U_i and N_i , these invariants do not. K and $\|\Pi\|_F$ are unchanged under any change of basis, and H changes only in sign under $N_i \rightarrow -N_i$. We therefore report the sign-free quantities K , H^2 , and $\|\Pi\|_F$, which require no consistent orientation (we discuss the implications of orientation invariance in §7.5).

7. Experiments

We demonstrate the tangent blow-up on four tasks: geodesic computation (§7.1), segmentation (§7.2), surface parameterization (§7.2), and curvature estimation (§6.3). We set $\alpha = 1.0$ for all examples, except in Figure 1, where we use $\alpha = 2.0$. For transverse intersections, α can be chosen using the separation bound in Theorem 1. In principle, the same criterion applies recursively to higher-order lifts. In practice, increasing α makes the metric more sensitive to curvature-induced tangent variation, as illustrated in Figure 11. For iterated blow-ups, this applies at each level, with higher-level weights controlling sensitivity to higher-order features.

For kernel constructions, we use $k = 20$ nearest neighbors. Spatial and angular bandwidths are set via self-tuning median k -NN distances [ZP04], where σ_0 is the median k -NN distance in spatial coordinates across all points, and each angular bandwidth σ_m is the median k -NN distance in the corresponding projector component of the lifted product metric defined in §4. For geodesic computation, these product-kernel weights also serve as the regression weights W_{ij} for the lifted gradient and divergence operators.

We test our algorithm on four classes of data. *Analytic shapes* like the Klein bottle provide controlled ground truth points and normals for quantitative evaluation [Do 16]. The *Thing10k* [ZJ16] and *ThreeDScans* [Lar12] datasets provide non-manifold geometry with self-intersections and T-junctions; we uniformly sample points from these meshes and take face normals. Lastly, the *SceneNN* [HPN*16] dataset provides real scans of rooms for testing our method on noisy data with estimated normals.

7.1. Geodesics

We compute approximate lifted geodesic distances via the heat method [CWW13], which recovers geodesic distance in three steps: (1) diffuse a delta source for short time by solving $(I + tL)u = \delta_s$, (2) normalize the gradient of the diffused field to obtain a unit vector field $X = -\nabla u / |\nabla u|$ pointing away from the source, and (3) solve a Poisson equation $L\phi = -\text{div}(X)$ to recover the distance function. In the lifted setting, we replace the standard Laplacian, gradient, and divergence with their lifted counterparts.

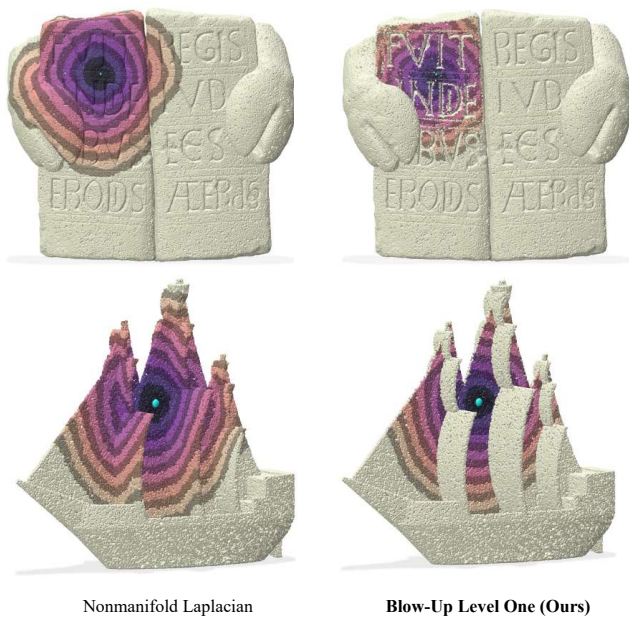


Figure 12: Geodesic distance via the heat method, with the source point marked in cyan. The nonmanifold Laplacian [SC20] (left) diffuses across self-intersections, while the lifted Laplacian (right) is restricted to geometrically distinct components.

We compare against the nonmanifold Laplacian of Sharp and Crane [SC20], which constructs an intrinsic Delaunay triangulation that handles non-manifold edges and boundaries. It operates on positions only and does not use tangent-plane information for sheet separation. The distances recovered by the lifted heat method approximate geodesics in the product metric, which penalizes tangent-plane rotation proportionally to α . Hence, as seen in Figure 12, lifted geodesics tend to favor straighter paths with smoothly varying tangents. This notion of distance is applicable whenever one wants shortest paths that respect the sheet structure of a non-manifold surface – for example, tracing paths on individual components of an object like the ship in Figure 12, without segmenting the hull, deck, and sails beforehand.

To demonstrate our approach on real scanned data with estimated normals, we compute geodesics for several rooms from the *SceneNN* dataset [HPN*16]. Any normal estimation algorithm can be used; we opt for *HSurf-Net* [LLC*22]. The results are illustrated in Figure 13, where we compute lifted geodesics along scanned floors. Despite missing regions, clutter, and noise, the resulting distance contours are constrained to the floor surface and respect obstacles such as walls and furniture. Such behavior may be useful for downstream navigation tasks such as path planning, where lifted geodesics could allow a robot to reason about shortest traversable paths along the floor while avoiding non-traversable regions.

7.2. Spectral Segmentation

The block structure of L_{sym} established in §6.1 reduces segmentation of a non-manifold point cloud to standard spectral clus-

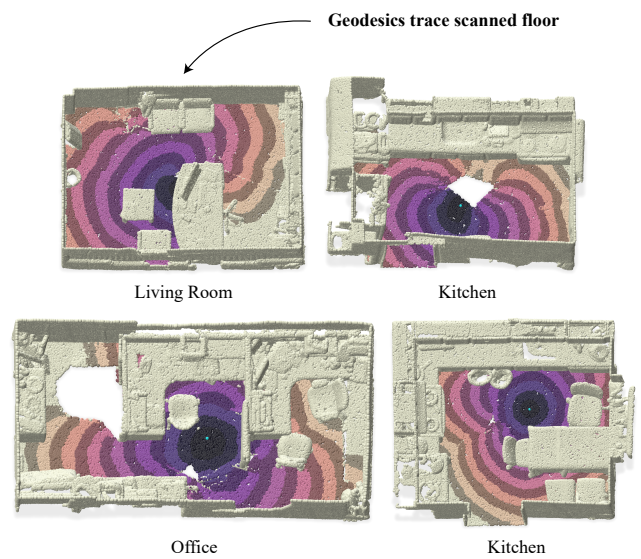


Figure 13: Lifted geodesics on scanned indoor scenes from the *SceneNN* dataset [HPN*16] using normals estimated by *HSurf-Net* [LLC*22]. Across living room, kitchen, and office scans, the computed distance contours trace the floor surface while respecting walls, furniture, and missing regions. This suggests that our method can operate on noisy real-world reconstructions and is relevant for applications such as path planning on scanned indoor environments.

tering [NJW01; Von07]. We take the $m = 10$ smallest non-trivial eigenvectors of L_{sym} , stack them as rows, and then unit-normalise each row [NJW01]. We then use DBSCAN [EKSX*96] with radius $\epsilon = 0.2$ to cluster points, which avoids fixing the number of components and, because it ignores directions of low variance, is insensitive to moderate overestimates of m .

As seen in Figure 14, lifted segmentation recovers geometrically distinct components such as individual icicles or the distinct sides of the fandisk. Since neighboring points on different components, as occur near sharp corners or sheet intersections, carry distinct projectors, the lifted metric suppresses their mutual affinity in W without any explicit singularity detection or feature classification. Recovering individual smooth components from non-manifold geometry in such a way can be useful for downstream tasks such as per-component surface parameterization (§7.2) and curvature estimation (§7.3).

Spectral Parameterization. Once the segmentation of §7.2 has isolated smooth components, we parameterize them individually using standard Laplacian eigenmaps [BN03] on the same lifted Laplacian used previously (the self-tuning product-kernel, symmetric normalised L_{sym} from §6.1).

We take the two smallest non-trivial eigenvectors of L_{sym} and use them directly as UV coordinates. Because L_{sym} is defined using the lifted metric, the corresponding eigenmap respects the local tangent frame even when the underlying components are self-intersecting and nonorientable; Figure 15 illustrates this on two

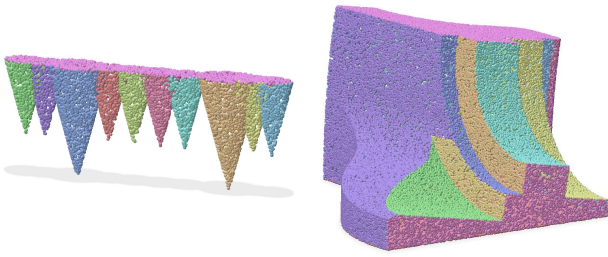


Figure 14: Lifted spectral segmentation on two non-manifold point clouds. Left: the self-intersecting icicles model (Thing10k [ZJ16]). Right: the piece-wise smooth fandisk [HDD*94]. Lifted spectral segmentation recovers individual smooth components.

nonorientable surfaces in \mathbb{R}^3 whose immersions necessarily self-intersect. The lifted Laplacian separates intersecting components through the projector term in the product metric, and because the projector $P = UU^T$ is sign-free by construction, the eigenmaps extend smoothly across regions where a consistent normal choice is impossible.

Typically, to perform surface parameterization, an artist would need to cut a shape into individual pieces that are separately parameterized and combined into a texture atlas. Our method resolves intersections and produces geometric cuts automatically, since the spectral gap induced by differing tangent planes (or, at the second blow-up level, differing curvatures) partitions the surface into smooth components. This means that an artist or automated pipeline can go from a non-manifold point cloud to per-component processing without ever manually cutting the geometry apart.

7.3. Curvature

For a parametric immersion $f: U \rightarrow \mathbb{R}^3$ whose Jacobian has full rank everywhere, Gaussian curvature K and mean curvature H are well-defined at every parameter value, even where the image surface self-intersects in \mathbb{R}^3 [Do 16]. A point cloud sampled from the image of f thus inherits per-point parameter coordinates (u_i, v_i) and well-defined curvature per sheet.

The lifted regression of B_i (§6.3) approximates the curvature of the underlying immersion at transverse intersections. We use this per-sheet ground truth for quantitative evaluation on two parametric surfaces and complement it with qualitative comparisons on Thing10k models [ZJ16] and ThreeDScans [Lar12], all sampled at up to $N = 200,000$ points.

Baselines. We compare against two point cloud curvature estimators. Jet fitting [CP05] estimates a truncated Taylor expansion (an osculating jet) of the surface in a local coordinate system aligned with the estimated normal. We use a degree-2 jet with $k = 20$ neighbors to match our blow-up baseline. Randomized Corrected Curvature Measures (CNC) [LCL*23] estimates the curvature tensor via corrected curvature measures [LRT22]: for each query point, L random triangles are sampled from the k nearest neighbors. We use the authors' code with $k = 20$ and $L = 100$. Jet fitting requires only

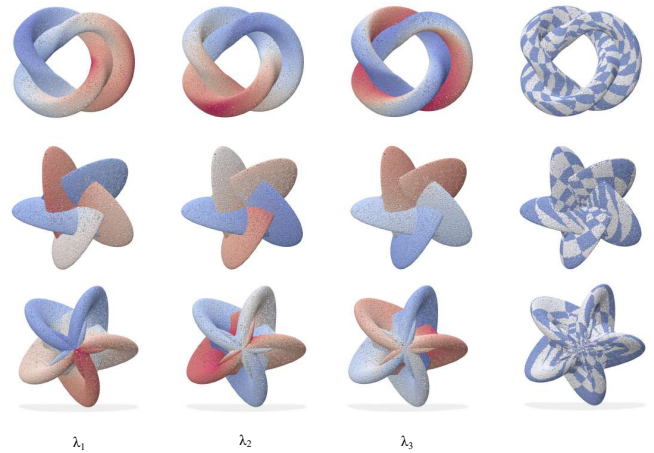


Figure 15: Spectral parameterization of two self-intersecting nonorientable surfaces: Banchoff's Klein bottle [Ban76; Bus12] (top) and a five-branch Boy-like surface [PS81; Bus12] (bottom two rows). Left to right: of the three smallest non-trivial eigenvectors of the lifted Laplacian L_{sym} , followed by the induced UV checker overlay. Because the lifted Laplacian is built from projectors, the eigenmaps extend smoothly through self-intersections and across regions where a consistent normal choice is impossible.

positions and estimates its own local frame from the k -NN neighborhood; CNC additionally requires oriented normals as input.

7.4. Quantitative evaluation.

We consider two surfaces: the torus, a smooth manifold that serves to validate our implementation, and the Klein bottle, a nonorientable surface whose immersion in \mathbb{R}^3 contains self-intersections.

The analytic curvature of the torus depend only on the poloidal angle v [RS22], so we can assess estimation accuracy as a function of a single variable. Figure 16 shows the estimated curvature profiles alongside the analytic ground truth (top row) and the corresponding absolute errors (middle row) as functions of the poloidal angle v . All three estimators track the ground truth closely, with median errors in the range 10^{-4} to 10^{-3} . The dip near $v = \pi$ in the H^2 error reflects the fact that H passes through zero at the inner equator for these parameter values ($R = 2r$), so the absolute error is small.

The Klein bottle is nonorientable and self-intersects when immersed in \mathbb{R}^3 [ASG17]. To quantify how each method degrades near singularities, we estimate each sample point's proximity to the Klein bottle's self-intersection. Two points that are close in \mathbb{R}^3 but far apart in the parameter domain (u, v) must lie on different sheets of the immersion. For each sample, we find its nearest neighbor in \mathbb{R}^3 among all samples whose parameter-space distance on the periodic $[0, 2\pi)^2$ domain exceeds the heuristic threshold $\pi/2$, and record the ambient distance to that neighbor. We call this the *singularity distance* of that sample.

Figure 17 plots the median absolute error as a function of singularity distance. Near the self-intersection (singularity distance

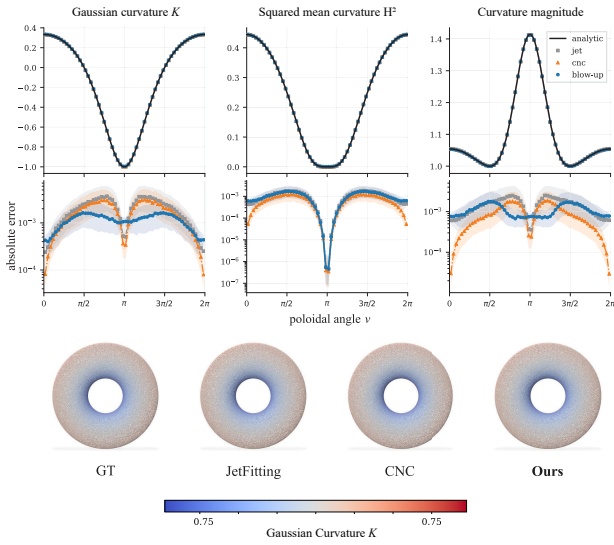


Figure 16: Curvature estimation on the torus ($R = 2$, $r = 1$, $N = 100,000$), a smooth manifold with no singularities. Top row: estimated K and H^2 as a function of the poloidal angle v ; the analytic ground truth (black) is nearly occluded by all three estimators. Middle row: median absolute error. All three methods achieve comparable accuracy. Bottom row: Gaussian curvature rendered on the surface.

< 0.1), both jet fitting and CNC exhibit error spikes of two to three orders of magnitude: median $|K - K_{gt}|$ rises from $\sim 10^{-3}$ to $\sim 10^1$ for jet fitting, and CNC rises similarly. The Klein bottle’s nonorientability compounds the problem for CNC, which requires oriented normals as input. The blow-up error, by contrast, remains flat at $\sim 5 \times 10^{-4}$ across all singularity distances. Figure 18 renders the three curvature invariants on the surface, with ground truth for comparison.

7.5. Qualitative evaluation.

We run all three methods on point clouds sampled from several meshes that contain self-intersections, T-junctions, or thin features. Each model is uniformly sampled at up to 200,000 oriented points. No analytic ground truth is available, so our criterion is whether the estimated curvature field is “spatially coherent” on each geometric component, meaning smooth where the surface is smooth, and free of high curvature artifacts at self-intersections. All three methods use $k = 20$ nearest neighbors; jet fitting uses degree 2, CNC uses $L = 100$ random triangles with a uniform kernel.

The torus knot (Figure 19) is a mesh from Thingi10k [ZJ16] whose geometry passes over and under itself. We render its Gaussian curvature K . The inset (red box) zooms into a self-intersecting region. CNC (left) produces a curvature field such that the local sign of K becomes negative at intersections. The blow-up (right) preserves positivity through every crossing, meaning the curvature on each tube segment is less affected by the presence of the other.

The propeller model [ZJ16] (Figure 20) consists of blades which

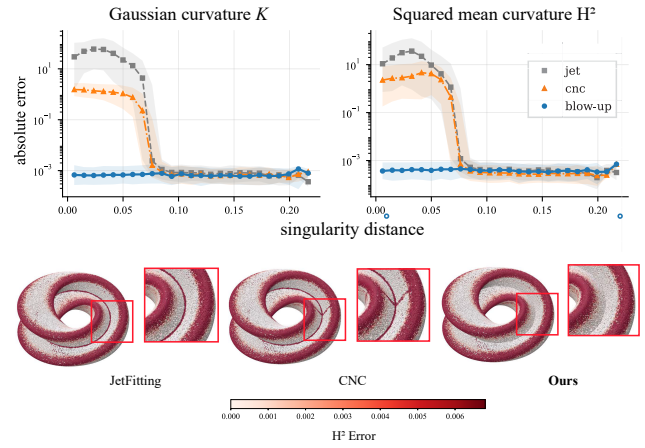


Figure 17: Curvature estimation error on the Klein bottle with $R = 2$ and $(u, v) \in [0, 2\pi)^2$, sampled at $N = 100,000$ points, as a function of singularity distance (defined in §7.3). Each sample is binned by its ambient distance to the nearest cross-sheet neighbor (25 bins, median point plotted with interquartile range shaded). Bottom row: per-point error in curvature magnitude $||I||_F - ||I_{gt}||_F$ rendered on the surface for each method; the blow-up shows no visible degradation at the intersection.

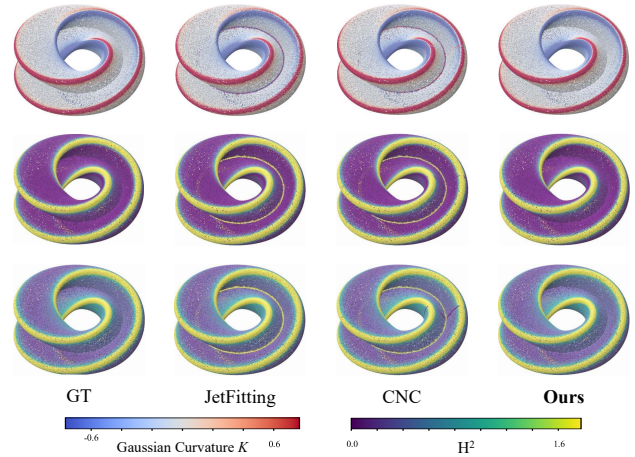


Figure 18: Curvature estimates on the Klein bottle (left to right: ground truth, jet fitting, CNC, blow-up). Rows show K (top), H^2 (middle), and curvature magnitude $||I||_F$ (bottom). Jet fitting and CNC produce visible artifacts along the self-intersection circle, while the blow-up closely matches the ground truth across all three invariants.

intersect a cylindrical hub. The blow-up recovers geometrically meaningful curvature on each separate component: the cylindrical base carries uniform magnitude and the flat blade faces read near zero. Jet fitting and CNC both report elevated H^2 along every intersection seam and sharp edge, making it difficult to distinguish the curvature of individual sheets from non-manifold junctions.

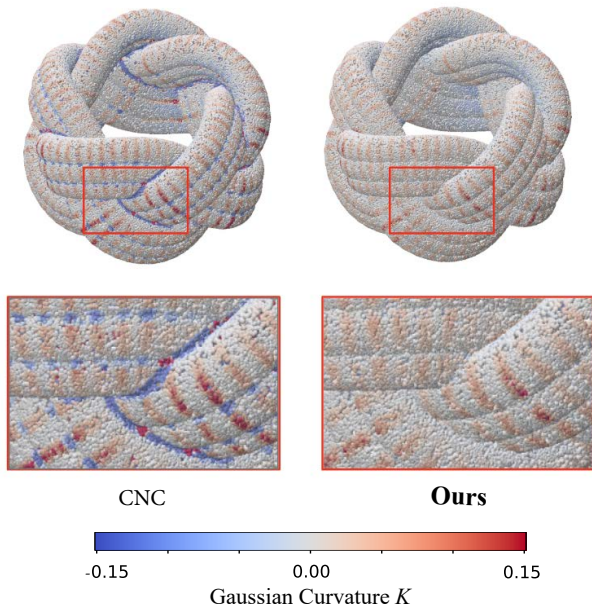


Figure 19: Gaussian curvature K on a self-intersecting torus knot. Bottom: inset zoom of the boxed crossing region. CNC produces negative curvature at the crossings. The blow-up maintains consistent banding on each knot segment through the intersections. Positive red banding is recovered from the underlying mesh.

Figure 21 shows squared mean curvature H^2 estimated on a point cloud sampled from two greyhound sculptures (ThreeD-Scans [Lar12]). Where the bodies touch, CNC reports elevated curvature along the intersection curves, visible as yellow streaks across the paws and chest in the close-up. The blow-up estimator attributes curvature to each sheet independently, so these ridges are absent. In practice this means per-component curvature fields can be recovered directly from the raw point cloud, with no prior segmentation or intersection removal.

Orientation Invariance. The curvature invariants we report are orientation-free by definition. Hence, any method that estimates them correctly should produce the same values regardless of a globally consistent normal field. Jet fitting, as seen in Figure 18, is an “orientation free” algorithm [CP05]. By contrast, CNC uses oriented normals, so inconsistent orientations corrupt the estimates [LCL*23]. Our algorithm avoids oriented normals entirely, since the projector $P = UU^T$, its derivative \mathcal{P}_v , and the regression for B_i all operate without choosing a consistent normal direction. Orientation enters only in the final step when extracting $B_x(v) = N^T \mathcal{P}_v U$, where we report orientation-free invariants. This makes the estimator applicable to nonorientable surfaces (Figure 3), but also means curvature is underestimated at thin shell edges where opposite faces carry nearly parallel tangent planes with opposite normals. The projector sees both “sides” of a thin sheet as having the same tangent plane, so it reports low curvature at thin edges

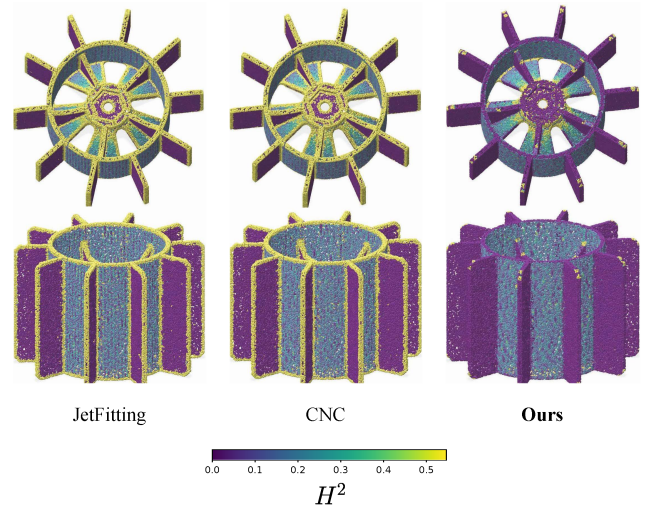


Figure 20: Mean squared curvature H^2 on a self-intersecting propeller model (Thingi10k [ZJ16]). Left: jet fitting. Center: CNC. Right: blow-up (ours). Jet fitting and CNC exhibit high-magnitude estimates (yellow) along intersections and edges. The discrete blow-up recovers consistent per-component curvature: the cylindrical base carries uniform magnitude, and the flat blade faces read near zero. No singularity detection or segmentation is applied as a preprocessing step.

whereas CNC detects the normal reversal, as seen in Figure 22. Whether this trade-off is desirable depends on the application.

8. Discussion

Our formulation assumes that a tangent plane is available at each sample. Future work could address automatically estimating multiple tangents at each point in the point cloud. For example, near singularities, PCA-based tangent estimation averages across intersecting components and can be sensitive to noise. This is illustrated in Figure 23, where the lifted heat method is robust to positional noise but sensitive to estimated tangents.

More specifically, we find that our method is generally robust to positional noise if one has reliable normals, as can be the case in data obtained by photometric scanning, but degrades with respect to increasingly noisy normals. Figure 24 illustrates this effect, where we have taken a point cloud sampled from a propeller mesh [ZJ16] and used local PCA to estimate normals. By adding noise to the positions and estimated tangents separately, we see how the lifted heat method is affected: lifted components remain well separated with respect to positional noise, but degrade more rapidly in correspondence to noisy normals. Developing noise-robust variants, especially for scanned data, is an important direction for future work.

A different strategy may be to estimate multiple tangents per point using local mixture-of-subspaces methods such as generalized PCA [VMS05] or polyvector fields [DVPS14; BS19]. Each estimated tangent would then generate a separate lifted copy of the same spatial sample, yielding a discrete approximation of the ex-

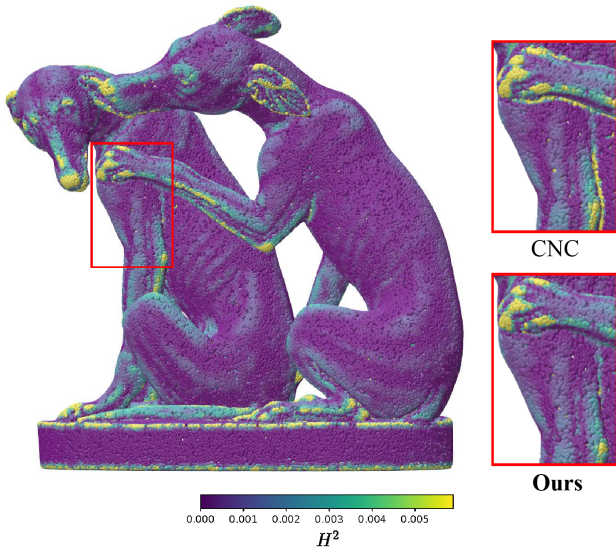


Figure 21: Squared mean curvature H^2 on two intersecting greyhound sculptures (ThreeDScans [Lar12]). Top: blow-up (ours). Bottom: CNC reports elevated curvature along intersection curves (visible as yellow ridges along the paws and legs); the blow-up attributes curvature to each greyhound independently.

ceptional fiber. The blow-up itself is agnostic to how these tangent planes are obtained.

Estimating multiple tangents could also improve the resolution of tangential intersections. The first lift can convert a tangential intersection into a transverse one, with sheets meeting at an angle determined by their curvature difference. Lifted k -NN neighborhoods may still include some cross-sheet samples near the singular point. In practice, this contamination is localized to a small neighborhood of the singularity that shrinks as the curvature difference between sheets grows. Figure 1 illustrates this, where the second lift achieves good separation despite the tangential contact. A more robust approach would involve handling multiple tangents at each blow-up level, which we leave to future work. Similarly, while the blow-up can be iterated, our experiments mainly validate the first lift and simple second-level examples. A systematic study of higher-order lifts on complex singularities is left to future work.

A second limitation is the growth of ambient dimension under iteration. For surfaces in \mathbb{R}^3 , the first lift lies in \mathbb{R}^{12} and the second in \mathbb{R}^{156} . Although the intrinsic dimension remains unchanged, the projector representation is redundant, since symmetric rank- d projectors have far fewer degrees of freedom than the ambient space. This overhead was still manageable in our experiments, but it appears in the timings. On the 200,000-point *Glykon* example in Figure 1, constructing the level-one and level-two lifts took 38.1 s and 76.9 s, respectively, while assembling the lifted affinity graph and linear systems for the lifted heat-method increased from about 4.4 s at level 1 to 30.7 s at level 2 (all timings measured on an Intel Core i9-9900K CPU with 8 physical cores and 16 logical cores, 48 GiB

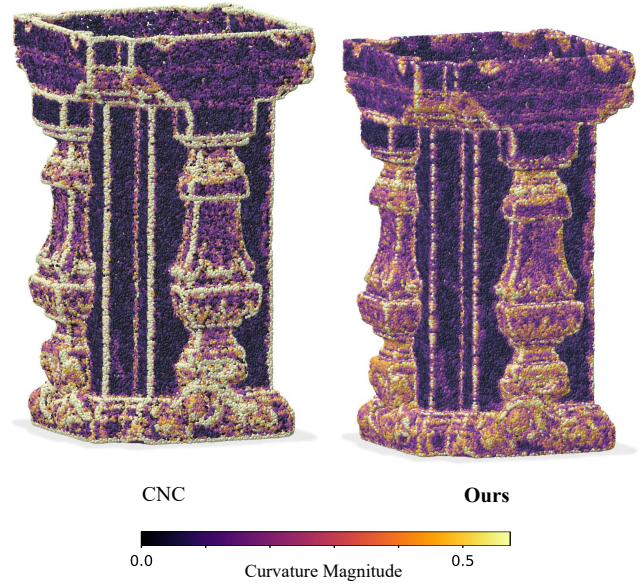


Figure 22: Curvature magnitude $\|II\|_F$ on a column model (ThreeDScans [Lar12]) with thin walls. Left: CNC reports high curvature along thin edges where oriented normals reverse between opposite faces. Right: the blow-up estimator reports low curvature at these edges because the tangent plane itself has little rotation.

of RAM, and an NVIDIA GeForce RTX 3070). In particular, exact kD -tree queries in the lifted ambient coordinates are likely to become a bottleneck as the dimension grows, since their effectiveness degrades in higher dimensions. For higher-order iterations, more compact Grassmannian coordinates, localized lifting near singular regions, or approximate nearest-neighbor methods may therefore be necessary. For oriented hypersurfaces, a cheaper alternative is to lift using the normal itself. This avoids the \mathbb{R}^{n+n^2} projector embedding and distinguishes opposite normal orientations. We use projectors because they are orientation-free and extend directly to arbitrary codimension via $\text{Gr}(d, n)$. However, if reliable oriented normals are available, the oriented lift can be substituted.

Beyond the operators explored here, the discrete tangent blow-up may also be useful for tasks such as interpolation and remeshing in the presence of singularities. Extending the framework to mixed-dimensional singularities, and clarifying its connection to higher-order jet constructions, are also natural next steps.

9. Conclusion

We introduced the discrete tangent blow-up, a lifted representation for point clouds with non-manifold geometry. By lifting each sample and its tangent plane to $\mathbb{R}^n \times \text{Gr}(d, n)$, the construction separates incident sheets in a metric space where standard Euclidean structures can be applied without combinatorial repair or explicit singularity detection. The projector formulation yields a simple embedding, supports iteration to incorporate higher-order geometric information, and leads to discretized differential operators that can

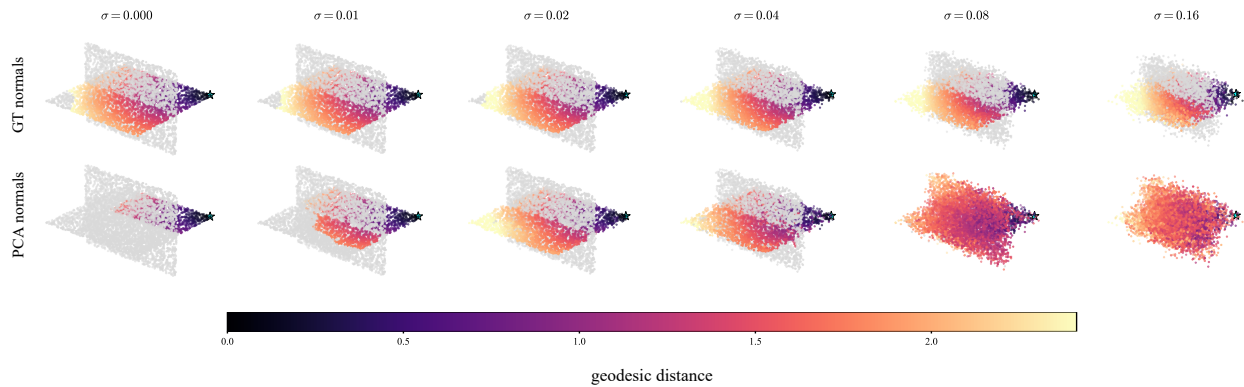


Figure 23: Geodesic distances on two intersecting planes ($N=6000$, $k=20$, $\alpha=1.0$, $\sigma_x=\sigma_u=0.5$). Top row: ground-truth tangent frames; bottom row: PCA-estimated tangents ($k_{\text{pca}}=30$). Columns show increasing positional noise $\sigma \in \{0, 0.01, 0.02, 0.04, 0.08, 0.16\}$. Grey points are unreachable (infinite geodesic distance), indicating correct sheet separation. The lifted heat method is robust to positional noise, but degrades when tangent estimates are corrupted, as PCA normals near the intersection line conflate intersecting components.

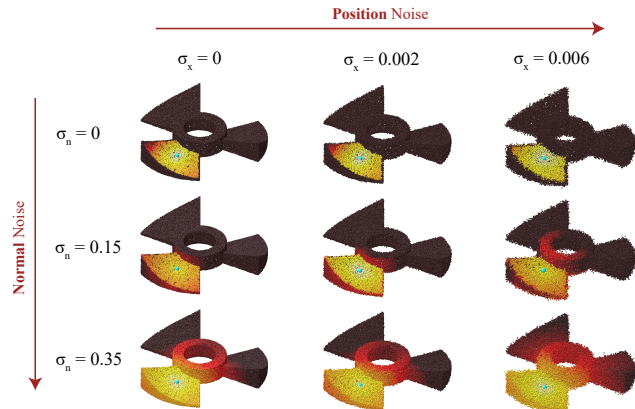


Figure 24: Effect of noise on lifted heat diffusion. Columns add increasing Gaussian noise to point positions, reported as fractions of the bounding-box diagonal. Rows add increasing Gaussian noise to normals. Normals are estimated by local PCA from the original point cloud with $k = 10$, and the lift uses $\alpha = 2.0$. Moderate perturbations preserve the qualitative diffusion pattern, while high combined noise causes visible heat leakage.

be defined directly in the lifted domain. We view this work as a step toward geometry processing methods that treat singularities as part of the data. We hope this perspective opens the door to a broader class of geometry processing algorithms that operate directly on non-manifold geometry.

Acknowledgements

The MIT Geometric Data Processing Group acknowledges the generous support of National Science Foundation grants IIS2335492 and OAC2403239, from the CSAIL Future of Data and FinTechAI programs, from the MIT-IBM Watson AI Laboratory, from the

Wistron Corporation, from the MIT Generative AI Impact Consortium, from the Toyota-CSAIL Joint Research Center, from the Natural Sciences and Engineering Research Council of Canada, and from Schmidt Sciences. Additional support was provided by SideFX Software. Finally, we thank Vidit Nanda for helpful discussions.

References

- [AB24] AAMARI, EDDIE and BERENFELD, CLÉMENT. “A theory of stratification learning”. *arXiv preprint arXiv:2405.20066* (2024) 2.
- [ACK13] ATTENE, MARCO, CAMPEN, MARCEL, and KOBBELT, LEIF. “Polygon mesh repairing: An application perspective”. *ACM Computing Surveys (CSUR)* 45.2 (2013), 1–33 3.
- [ALC*26] ARNAL-ANGER, LÉO, LEJEMBLE, THIBAUT, COEURJOLLY, DAVID, et al. “Survey on differential estimators for 3d point clouds”. *Computer Graphics Forum*. Wiley Online Library, 2026, e70394 3.
- [ARAC14] ANDREUX, MATHIEU, RODOLA, EMANUELE, AUBRY, MATHIEU, and CREMERS, DANIEL. “Anisotropic Laplace-Beltrami operators for shape analysis”. *European conference on computer vision*. Springer, 2014, 299–312 3.
- [ASG17] ABBENA, ELSA, SALAMON, SIMON, and GRAY, ALFRED. *Modern differential geometry of curves and surfaces with Mathematica*. Chapman and Hall/CRC, 2017 12.
- [Ban76] BANCHOFF, THOMAS F. “Minimal submanifolds of the bicylinder boundary”. *Boletim da Sociedade Brasileira de Matemática-Bulletin/Brazilian Mathematical Society* 7.1 (1976), 37–57 2, 12.
- [BG20] BEN-SHABAT, YIZHAK and GOULD, STEPHEN. “DeepFit: 3D surface fitting via neural network weighted least squares”. *European conference on computer vision*. Springer, 2020, 20–34 3.
- [BH11] BENDICH, PAUL and HARER, JOHN. “Persistent intersection homology”. *Foundations of Computational Mathematics* 11.3 (2011), 305–336 2, 3.
- [BLM17] BUET, BLANCHE, LEONARDI, GIAN PAOLO, and MASNOU, SIMON. “A varifold approach to surface approximation”. *Archive for Rational Mechanics and Analysis* 226.2 (2017), 639–694 3.
- [BLM22] BUET, BLANCHE, LEONARDI, GIAN PAOLO, and MASNOU, SIMON. “Weak and approximate curvatures of a measure: a varifold perspective”. *Nonlinear Analysis* 222 (2022), 112983 3.

- [BMR*16] BOSCAINI, DAVIDE, MASCI, JONATHAN, RODOLÀ, EMANUELE, et al. “Anisotropic diffusion descriptors”. *Computer Graphics Forum*. Vol. 35. 2. Wiley Online Library. 2016, 431–441 3.
- [BMW10] BENDICH, PAUL, MUKHERJEE, SAYAN, and WANG, BEI. “Stratification Learning through Homology Inference.” *AAAI Fall Symposium: Manifold Learning and Its Applications*. 2010 2.
- [BN03] BELKIN, MIKHAIL and NIYOGI, PARTHA. “Laplacian eigenmaps for dimensionality reduction and data representation”. *Neural computation* 15.6 (2003), 1373–1396 11.
- [BN06] BELKIN, MIKHAIL and NIYOGI, PARTHA. “Convergence of Laplacian eigenmaps”. *Advances in neural information processing systems* 19 (2006) 2, 8.
- [Bou23] BOUMAL, NICOLAS. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023 5.
- [BR22] BUET, BLANCHE and RUMPF, MARTIN. “Mean curvature motion of point cloud varifolds”. *ESAIM: Mathematical Modelling and Numerical Analysis* 56.5 (2022), 1773–1808 3.
- [BS19] BESSMELTSEV, MIKHAIL and SOLOMON, JUSTIN. “Vectorization of line drawings via polyvector fields”. *ACM Transactions on Graphics (TOG)* 38.1 (2019), 1–12 14.
- [BSW09] BELKIN, MIKHAIL, SUN, JIAN, and WANG, YUSU. “Constructing Laplace operator from point clouds in \mathbb{R}^d ”. *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*. SIAM. 2009, 1031–1040 1, 2, 8.
- [Bus12] BUSSEY, ALAIN. *Non-Orientable Surfaces: Parametric Representations*. <https://eumat.sourceforge.net/Programs/Examples/Non-Orientable%20Surfaces.html>. Accessed April 2026. 2012 2, 12.
- [BWM12] BENDICH, PAUL, WANG, BEI, and MUKHERJEE, SAYAN. “Local homology transfer and stratification learning”. *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 2012, 1355–1370 3.
- [BZA24] BENDOKAT, THOMAS, ZIMMERMANN, RALF, and ABSIL, P-A. “A Grassmann manifold handbook: Basic geometry and computational aspects”. *Advances in Computational Mathematics* 50.1 (2024), 6 4–7.
- [CBK21] CHARTON, JEROME, BAEK, STEPHEN, and KIM, YOUNGJUN. “Mesh repairing using topology graphs”. *Journal of Computational Design and Engineering* 8.1 (2021), 251–267 2, 3.
- [CDLL26] CASTILLO, FEDERICO, DUARTE, DANIEL, LEYTON-ÁLVAREZ, MAXIMILIANO, and LIENDO, ALVARO. “Nash blowup fails to resolve singularities in dimensions four and higher”. *Annals of Mathematics* 203.2 (2026), 677–694 3.
- [CG06] CANAS, GUILLERMO D and GORTLER, STEVEN J. “Surface remeshing in arbitrary codimensions”. Vol. 22. 9. Springer, 2006, 885–895 3.
- [CHS96] CONWAY, JOHN H, HARDIN, RONALD H, and SLOANE, NEIL JA. “Packing lines, planes, etc.: Packings in Grassmannian spaces”. *Experimental mathematics* 5.2 (1996), 139–159 6.
- [CL06] COIFMAN, RONALD R and LAFON, STÉPHANE. “Diffusion maps”. *Applied and computational harmonic analysis* 21.1 (2006), 5–30 1, 9.
- [CLS*21] CAO, YUEQI, LI, DIDONG, SUN, HUAFEI, et al. “Efficient Weingarten map and curvature estimation on manifolds”. *Machine Learning* 110.6 (2021), 1319–1344 3.
- [CP05] CAZALS, FRÉDÉRIC and POUGET, MARC. “Estimating differential quantities using polynomial fitting of osculating jets”. *Computer aided geometric design* 22.2 (2005), 121–146 1, 3, 12, 14.
- [CWW13] CRANE, KEENAN, WEISCHEDEL, CLARISSE, and WARDEZKY, MAX. “Geodesics in heat: A new approach to computing distance based on heat flow”. *ACM Transactions on Graphics (ToG)* 32.5 (2013), 1–11 10.
- [Do 16] DO CARMO, MANFREDO P. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016 10, 12.
- [DVPS14] DIAMANTI, OLGA, VAXMAN, AMIR, PANOZZO, DANIELE, and SORKINE-HORNUNG, OLGA. “Designing N-PolyVector fields with complex polynomials”. *Computer Graphics Forum*. Vol. 33. 5. Wiley Online Library. 2014, 1–11 14.
- [EAS98] EDELMAN, ALAN, ARIAS, TOMÁS A, and SMITH, STEVEN T. “The geometry of algorithms with orthogonality constraints”. *SIAM journal on Matrix Analysis and Applications* 20.2 (1998), 303–353 6.
- [EGO*20] ERLER, PHILIPP, GUERRERO, PAUL, OHRHALLINGER, STEFAN, et al. “Points2surf learning implicit surfaces from point clouds”. *European conference on computer vision*. Springer. 2020, 108–124 3.
- [EKSX*96] ESTER, MARTIN, KRIEGEL, HANS-PETER, SANDER, JÖRG, XU, XIAOWEI, et al. “A density-based algorithm for discovering clusters in large spatial databases with noise”. *kdd*. Vol. 96. 34. 1996, 226–231 11.
- [FDC03] FLEISHMAN, SHACHAR, DRORI, IDDO, and COHEN-OR, DANIEL. “Bilateral mesh denoising”. (2003), 950–953 3.
- [Fri20] FRIEDMAN, GREG. *Singular intersection homology*. Vol. 33. Cambridge University Press, 2020 4.
- [GKOM18] GUERRERO, PAUL, KLEIMAN, YANIR, OVSJANIKOV, MAK, and MITRA, NILOY J. “Pcpnet learning local shape properties from raw point clouds”. *Computer graphics forum*. Vol. 37. 2. Wiley Online Library. 2018, 75–85 3.
- [GM88] GORESKY, MARK and MACPHERSON, ROBERT. “Stratified morse theory”. *Stratified Morse Theory*. Springer. 1988, 3–22 4.
- [Gon77] GONZÁLEZ-SPRINBERG, GERARDO. “Éventails en dimension 2 et transformé de Nash”. 1977 2.
- [GV13] GOLUB, GENE H and VAN LOAN, CHARLES F. *Matrix computations*. JHU press, 2013 4.
- [HDD*92] HOPPE, HUGUES, DEROSE, TONY, DUCHAMP, TOM, et al. “Surface reconstruction from unorganized points”. *Proceedings of the 19th annual conference on computer graphics and interactive techniques*. 1992, 71–78 1.
- [HDD*94] HOPPE, HUGUES, DEROSE, TONY, DUCHAMP, TOM, et al. “Piecewise smooth surface reconstruction”. *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. 1994, 295–302 12.
- [Hir83] HIRONAKA, HEISUKE. “On Nash blowing-up”. *Arithmetic and Geometry: Papers Dedicated to IR Shafarevich on the Occasion of His Sixtieth Birthday. Volume II: Geometry*. Springer, 1983, 103–111 2.
- [HPN*16] HUA, BINH-SON, PHAM, QUANG-HIEU, NGUYEN, DUC THANH, et al. “Scenenn: A scene meshes dataset with annotations”. *2016 fourth international conference on 3D vision (3DV)*. Ieee. 2016, 92–101 10, 11.
- [HRS06] HARO, GLORIA, RANDALL, GREGORY, and SAPIRO, GUILLERMO. “Stratification learning: Detecting mixed density and dimensionality in high dimensional point clouds”. *Advances in Neural Information Processing Systems* 19 (2006) 3.
- [JDD03] JONES, THOUIS R, DURAND, FRÉDO, and DESBRUN, MATHIEU. “Non-iterative, feature-preserving mesh smoothing”. (2003), 943–949 3.
- [KL97] KAMBHATLA, NANDAKISHORE and LEEN, TODD K. “Dimension reduction by local principal component analysis”. *Neural computation* 9.7 (1997), 1493–1516 3.
- [KLAK25] KOHLBRENNER, MAXIMILIAN, LIU, HONGYI, ALEXA, MARC, and KAZHDAN, MISHA. “Symmetrized Poisson Reconstruction”. *Computer Graphics Forum*. Vol. 44. 5. Wiley Online Library. 2025, e70210 3.
- [Lar12] LARIC, OLIVER. *Three D Scans*. <http://threedscans.com/>. Accessed: 2026-04. 2012 10, 12, 14, 15.

- [LB13] LÉVY, BRUNO and BONNEEL, NICOLAS. “Variational anisotropic surface meshing with Voronoi parallel linear enumeration”. *Proceedings of the 21st international meshing roundtable*. Springer, 2013, 349–366 3.
- [LCL*23] LACHAUD, J-O, COEURJOLLY, DAVID, LABART, CÉLINE, et al. “Lightweight curvature estimation on point clouds with randomized corrected curvature measures”. 42.5 (2023), e14910 3, 12, 14.
- [Lee03] LEE, JOHN M. “Smooth manifolds”. *Introduction to smooth manifolds*. Springer, 2003, 1–29 4, 7.
- [LHW*25] LI, YUANQI, HUANG, JINGCHENG, WANG, HONGSHEN, et al. “High-quality Point Cloud Oriented Normal Estimation via Hybrid Angular and Euclidean Distance Encoding”. *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 1287–1296 3.
- [LLC*22] LI, QING, LIU, YU-SHEN, CHENG, JIN-SAN, et al. “HSurfNet: Normal estimation for 3D point clouds by learning hyper surfaces”. *Advances in Neural Information Processing Systems* 35 (2022), 4218–4230 3, 11.
- [LON25] LIM, UZU, OBERHAUSER, HARALD, and NANDA, VIDIT. “Hades: Fast singularity detection with local measure comparison”. *SIAM Journal on Mathematics of Data Science* 7.4 (2025), 1882–1903 3.
- [LRT22] LACHAUD, JACQUES-OLIVIER, ROMON, PASCAL, and THIBERT, BORIS. “Corrected curvature measures”. *Discrete & Computational Geometry* 68.2 (2022), 477–524 3, 12.
- [LZ13] LIANG, JIAN and ZHAO, HONGKAI. “Solving partial differential equations on point clouds”. *SIAM Journal on Scientific Computing* 35.3 (2013), A1461–A1486 9.
- [LZH*06] LAI, YU-KUN, ZHOU, QIAN-YI, HU, SHI-MIN, et al. “Robust feature classification and editing”. *IEEE Transactions on Visualization and Computer Graphics* 13.1 (2006), 34–45 3.
- [MCO08] MARSHALL, SCOTT T, COOKE, MICHELE L, and OWEN, SUSAN E. “Effects of nonplanar fault topology and mechanical interaction on fault-slip distributions in the Ventura Basin, California”. *Bulletin of the Seismological Society of America* 98.3 (2008), 1113–1127 1.
- [MHLZ21] MA, BAORUI, HAN, ZHIZHONG, LIU, YU-SHEN, and ZWICKER, MATTHIAS. “Neural-Pull: Learning Signed Distance Function from Point clouds by Learning to Pull Space onto Surface”. *International Conference on Machine Learning*. PMLR. 2021, 7246–7257 3.
- [Nan20] NANDA, VIDIT. “Local cohomology and stratification”. *Foundations of Computational Mathematics* 20.2 (2020), 195–222 2, 3.
- [Nas96] NASH JR, JOHN F. “Arc structure of singularities”. *Duke Math. J* 81.1 (1996), 31–38 2.
- [Nea04] NEALEN, ANDREW. “An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation”. URL: <http://www.nealen.com/projects> 130.150 (2004), 25 9.
- [NJW01] NG, ANDREW, JORDAN, MICHAEL, and WEISS, YAIR. “On spectral clustering: Analysis and an algorithm”. *Advances in neural information processing systems* 14 (2001) 11.
- [Nob75] NOBILE, AUGUSTO. “Some properties of the Nash blowing-up”. *Pacific Journal of Mathematics* 60.1 (1975), 297–305 2, 3.
- [PS81] PETIT, JEAN-PIERRE and SOURIAU, J. “Une représentation analytique de la surface de Boy”. *Compte Rendus de l’Académie des Sciences de Paris* 293 (1981), 5 12.
- [PSH*04] POTTMANN, HELMUT, STEINER, TIBOR, HOFER, MICHAEL, et al. “The isophotic metric and its application to feature sensitive morphology on surfaces”. *European Conference on Computer Vision*. Springer. 2004, 560–572 3.
- [Reb77] REBASSOO, VELLO. “Desingularisation properties of the Nash blowing-up process”. PhD thesis. University of Washington, 1977 3.
- [RS22] ROBBIN, JOEL W and SALAMON, DIETMAR A. *Introduction to differential geometry*. Springer Nature, 2022 1, 5, 12.
- [Rus04] RUSINKIEWICZ, SZYMON. “Estimating curvatures and their derivatives on triangle meshes”. *Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004*. IEEE. 2004, 486–493 3.
- [SC20] SHARP, NICHOLAS and CRANE, KEENAN. “A laplacian for non-manifold triangle meshes”. *Computer Graphics Forum*. Vol. 39. 5. Wiley Online Library. 2020, 69–80 1, 3, 11.
- [Sem54] SEMPLE, J GREENLEES. “Some investigations in the geometry of curve and surface elements”. *Proceedings of the London Mathematical Society* 3.1 (1954), 24–49 2.
- [STHN20] STOLZ, BERNADETTE J, TANNER, JARED, HARRINGTON, HEATHER A, and NANDA, VIDIT. “Geometric anomaly detection in data”. *Proceedings of the national academy of sciences* 117.33 (2020), 19664–19669 2, 3.
- [Tin23] TINARRAGE, RAPHAËL. “Recovering the homology of immersed manifolds”. *Discrete & Computational Geometry* 69.3 (2023), 659–744 3.
- [TM98] TOMASI, CARLO and MANDUCHI, ROBERTO. “Bilateral filtering for gray and color images”. *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE. 1998, 839–846 3.
- [VCD*16] VAXMAN, AMIR, CAMPEN, MARCEL, DIAMANTI, OLGA, et al. “Directional field synthesis, design, and processing”. *Computer graphics forum*. Vol. 35. 2. Wiley Online Library. 2016, 545–572 1.
- [VL25] VALQUE, LÉO and LAZARD, SYLVAIN. “Resolving self-intersections in 3D meshes while preserving floating-point coordinates”. *Computer Graphics Forum*. Vol. 44. 5. Wiley Online Library. 2025, e70197 3.
- [VMS05] VIDAL, RENE, MA, YI, and SASTRY, SHANKAR. “Generalized principal component analysis (GPCA)”. *IEEE transactions on pattern analysis and machine intelligence* 27.12 (2005), 1945–1959 3, 14.
- [Von07] VON LUXBURG, ULRIKE. “A tutorial on spectral clustering”. *Statistics and computing* 17.4 (2007), 395–416 9, 11.
- [VR23] VON ROHRSCHEIDT, JULIUS and RIECK, BASTIAN. “Topological singularity detection at multiple scales”. *International Conference on Machine Learning*. PMLR. 2023, 35175–35197 3.
- [WNEH22] WIERSMA, RUBEN, NASIKUN, AHMAD, EISEMANN, ELMAR, and HILDEBRANDT, KLAUS. “Deltaconv: anisotropic operators for geometric deep learning on point clouds”. *ACM Transactions on Graphics (ToG)* 41.4 (2022), 1–10 9.
- [YJL*25] YANG, XINRAN, JI, DONGHAO, LI, YUANQI, et al. “EdgeMovingNet: Edge-preserving Point Cloud Reconstruction via Joint Geometry Features”. *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 22150–22160 3.
- [ZGZJ16] ZHOU, QINGNAN, GRINSPUN, EITAN, ZORIN, DENIS, and JACOBSON, ALEC. “Mesh arrangements for solid geometry”. *ACM Transactions on Graphics (TOG)* 35.4 (2016), 1–15 3.
- [ZJ16] ZHOU, QINGNAN and JACOBSON, ALEC. “Thing10K: A Dataset of 10,000 3D-Printing Models”. *arXiv preprint arXiv:1605.04797* (2016) 10, 12–14.
- [ZP04] ZELNIK-MANOR, LIHI and PERONA, PIETRO. “Self-tuning spectral clustering”. *Advances in neural information processing systems* 17 (2004) 9, 10.

Appendix A: Proofs of Separation and Smoothness Theorems

Theorem 3 (Lifted Separation) Let $M_1, M_2 \subset \mathbb{R}^n$ be smooth d -dimensional submanifolds intersecting at a point x_0 with distinct tangent planes. Write $P_0^{(1)}, P_0^{(2)}$ for their respective tangent projectors at x_0 , and let

$$\delta = d_{\text{chord}}(T_{x_0}M_1, T_{x_0}M_2) > 0$$

be the chordal distance between the two tangent planes. Then there exists $\epsilon_0 > 0$ (depending on δ and the second fundamental forms of M_1, M_2 near x_0) such that for all $p \in M_1, q \in M_2$ with $\|p -$

$x_0\|, \|q - x_0\| < \epsilon_0$,

$$d_{\mathcal{L}}(\tilde{p}, \tilde{q}) \geq \frac{\sqrt{\alpha} \delta}{\sqrt{2}} > 0,$$

where $\tilde{p} = (p, T_p M_1)$, $\tilde{q} = (q, T_q M_2)$ are the lifted points and $d_{\mathcal{L}}$ is the lifted metric.

Proof Write $P_x^{(k)}$ for the tangent projector of M_k at $x \in M_k$, and $P_0^{(k)} = P_{x_0}^{(k)}$ for its value at x_0 , for $k = 1, 2$.

Step 1 (Projector Lipschitz bound). Since each M_k is smooth, the projector map $x \mapsto P_x^{(k)}$ is smooth on M_k . Its directional derivative along a tangent vector v is the projector derivative $\mathcal{P}_v^{(k)}$, which decomposes as $\mathcal{P}_v^{(k)} = N^{(k)} B^{(k)}(v) (U^{(k)})^\top + U^{(k)} B^{(k)}(v)^\top (N^{(k)})^\top$. The two summands have orthogonal column spaces ($\text{col}(N^{(k)})$ and $\text{col}(U^{(k)})$), so we can use $\|A + B\|_F^2 = \|A\|_F^2 + \|B\|_F^2$ (the trace $\text{tr}(A^\top B) = 0$). Multiplying by matrices with orthonormal columns preserves the Frobenius norm, so each summand contributes $\|B^{(k)}(v)\|_F$, giving

$$\|\mathcal{P}_v^{(k)}\|_F^2 = 2 \|B^{(k)}(v)\|_F^2 \leq 2 L_k^2 \|v\|^2,$$

where $L_k = \max\{\|B_x^{(k)}(v)\|_F : x \in M_k, \|x - x_0\| \leq \epsilon_0, \|v\| = 1\}$ is finite by smoothness and compactness. By the mean value inequality,

$$\|P_x^{(k)} - P_0^{(k)}\|_F \leq \sqrt{2} L_k d_{M_k}(x, x_0),$$

where d_{M_k} is the geodesic distance on M_k . For ϵ_0 sufficiently small, $d_{M_k}(x, x_0) \leq 2 \|x - x_0\|$, so setting $C_k = 2\sqrt{2} L_k$ gives $\|P_x^{(k)} - P_0^{(k)}\|_F \leq C_k \|x - x_0\|$.

Step 2 (Triangle inequality on projectors). For $p \in M_1$ and $q \in M_2$ with $\|p - x_0\|, \|q - x_0\| < \epsilon_0$, set $C = C_1 + C_2$:

$$\begin{aligned} \|P_p^{(1)} - P_q^{(2)}\|_F &\geq \|P_0^{(1)} - P_0^{(2)}\|_F - \|P_p^{(1)} - P_0^{(1)}\|_F - \|P_q^{(2)} - P_0^{(2)}\|_F \\ &\geq \sqrt{2} \delta - C \epsilon_0, \end{aligned}$$

where we used $\|P_0^{(1)} - P_0^{(2)}\|_F = \sqrt{2} \delta$, the standard identity relating the projector Frobenius distance to the chordal distance on the Grassmannian.

Step 3 (Lifted distance). The squared lifted distance satisfies

$$d_{\mathcal{L}}^2(\tilde{p}, \tilde{q}) = \|p - q\|^2 + \frac{\alpha}{2} \|P_p^{(1)} - P_q^{(2)}\|_F^2 \geq \frac{\alpha}{2} (\sqrt{2} \delta - C \epsilon_0)^2.$$

Choosing $\epsilon_0 < (\sqrt{2} - 1) \delta / C$ ensures $\sqrt{2} \delta - C \epsilon_0 > \delta$, so that

$$d_{\mathcal{L}}(\tilde{p}, \tilde{q}) \geq \sqrt{\frac{\alpha}{2}} \delta = \frac{\sqrt{\alpha} \delta}{\sqrt{2}}.$$

□

Theorem 4 (Regularity of the Lifted Manifold) Let $M \subset \mathbb{R}^n$ be a C^r submanifold of dimension d with $r \geq 2$. Then the lifted embedding $\Phi : M \rightarrow \mathbb{R}^{n+n^2}$, $\Phi(x) = (x, \sqrt{\alpha/2} \text{vec}(P_x))$, is a C^{r-1} embedding, and the lifted manifold $\tilde{M} = \Phi(M)$ is a C^{r-1} submanifold of dimension d .

Proof The map Φ is a section of the projection $\pi : \mathbb{R}^{n+n^2} \rightarrow \mathbb{R}^n$ onto the first factor: by construction, $\pi \circ \Phi = \text{id}_M$. For any section, injectivity ($\Phi(p) = \Phi(q) \Rightarrow p = \pi(\Phi(p)) = \pi(\Phi(q)) = q$), immersivity

($d\pi \circ d\Phi = \text{id}$ implies $d\Phi$ is injective), and topological embedding ($\Phi^{-1} = \pi|_{\tilde{M}}$ is continuous) are automatic. It therefore suffices to show that Φ is C^{r-1} .

The first component of Φ is the inclusion $M \hookrightarrow \mathbb{R}^n$, which is C^r . For the second component, let $\varphi : V \rightarrow M$ be a C^r local parametrization with Jacobian $J(x) = D\varphi_{\varphi^{-1}(x)} \in \mathbb{R}^{n \times d}$. The projector admits the parametrization-free expression

$$P_x = J(x) (J(x)^\top J(x))^{-1} J(x)^\top.$$

Since φ is C^r , the Jacobian J is C^{r-1} . The Gram matrix $J^\top J$ is C^{r-1} and invertible (since J has full column rank), and matrix inversion is smooth on the open set of invertible matrices. Therefore $x \mapsto P_x$ is C^{r-1} , and hence so is Φ . □