Persistence Homology of Proximity Hyper-Graphs for Higher Dimensional Big Data

Rohit P. Singh and Philip A. Wilsey
University of Cincinnati, Cincinnati, OH 45221, USA
Email: singh2ro@mail.uc.edu, wilseypa@gmail.comn

***Abstract*—Persistent Homology (PH) is a method of *Topological Data Analysis* that analyzes the topological structure of data to help data scientists infer relationships in the data to assist in informed decision- making. A significant component in the computation of PH is the construction and use of a *complex* that represents the topological structure of the data. Some complex types are fast to construct but space inefficient whereas others are costly to construct and space efficient. Unfortunately, existing complex types are not both fast to construct and compact.**

This paper works to increase the scope of PH to support the computation of low dimensional homologies (H_0–H_{10}) in high-dimension, big data. In particular, this paper exploits the desirable properties of the *Vietoris–Rips* Complex (*VR*-Complex) and the *Delaunay* Complex in order to construct a *sparsified complex*. The *VR*-Complex uses a distance matrix to quickly generate a complex up to the desired homology dimension. In contrast, the Delaunay Complex works at the dimensionality of the data to generate a sparsified complex. While construction of the *VR*-Complex is fast, its size grows exponentially by the size *and* dimension of the data set; in contrast, the Delaunay complex is significantly smaller for any given data dimension. However, its construction requires the computation of a Delaunay Triangulation that has high computational complexity. As a result, it is difficult to construct a Delaunay Complex for data in dimensions $d > 6$ that contains more than a few hundred points. The techniques in this paper enable the computation of topological preserving sparsification of k-Simplices (where $k \ll d$) to quickly generate a reduced sparsified complex sufficient to compute homologies up to k-subspace, irrespective of the data dimensionality d.

***Index Terms*—Proximity Hyper-graphs, Simplicial Complex, Persistent Homology, Data Mining**

I. Introduction

Topological Data Analysis (TDA) aims to develop tools for studying the qualitative features of data using results and ideas from geometry and topology. TDA techniques and specifically *Persistent Homology (PH)* provide data scientists a tool to understand and comprehend the higher dimensional landscapes of data [1]–[3]. PH provides a precise and robust definition of the qualitative features (and a systematic way to compute them) of high dimensional data [4]. PH computes homological features of the space with 0-dimensional features represented by connected components, 1-dimensional features by loops, 2-dimensional features by voids, and so on in higher dimensions. These features remain unperturbed under geometric deformations such as stretching, bending, expanding, shrinking, and rotation [5], [6]. These properties provide a versatile and resilient description of the data sets and are sometimes termed the *characteristic signature* of the data [7].

TDA analysis techniques (especially PH) have been successfully utilized in various application fields to infer higher dimensional relationships in complex data. It has provided valuable insight into research problems including the structural and functional analysis of proteins [8]–[10], cell development and differentiation trajectories [11]–[13], natural language processing [14] and statistical inference [15]–[17]. For proteins, the PH barcode signature helped in distinguishing the alpha-helix and beta-sheets based structural classes of proteins. For cell differentiation in the embryonic developmental stages, TDA can help identify cell transition signifying major events during prenatal and embryonic stages [18]–[21]. The application of TDA techniques has seen an unprecedented surge with recent developments in materials research [22], dynamic and recurrent systems [23], and other domains.

Many important data analysis problems involve complex, high-dimensional big data. Therefore it is desirable that TDA and PH be expanded to support the processing of big data. The computation of PH requires a construction of topological *complex* representing the topology of the space characterized by the data [24]; in general, the complex constructed is a *simplicial complex*. Unfortunately, the space complexity of the simplicial complex increases exponentially with the size and dimension of the data and it can quickly become prohibitively large [2]. While there are a number of types of simplicial complexes [25]–[32], the *Vietoris–Rips (*VR*)* complex is the most widely used. The *VR*-complex enjoys a fast construction time, but its memory complexity prevents its use on big data or high-dimensional data [32]. The sparsified *Delaunay complex* is much smaller in size with respect to the fully expanded *VR*-complex but it requires the computation of Delaunay triangulation in the dimension of the data [27]. Unfortunately, the computation of Delaunay triangulation in higher dimensional spaces is prohibited due to the size of exploration space even for moderate size data sets [33].

This paper addresses the construction of simplicial complexes for higher dimensional data sets and enables triangulation of the subspace by inducing a lower dimensional hyper-graph mesh on the point cloud. This subspace hyper-graph mesh generation overcomes the Delaunay triangulation requirement for constructing *compact sparsified complexes* in higher dimensions. This technique extends the edge based β-skeleton proximity graph induction technique to the general dimension by inducing k-uniform hyper-graph (k-simplex) mesh for point cloud in $\mathbb{R}^n$ for $k \ll d$ [34], [35]. The

978-1-6654-8045-1/22/$31.00

sparsification factor $0 \leq \beta^s \leq \infty$ controls the degree and density of hyper-graph with no sparsification at $\beta^s = 0$ and sparsification increases with increase in β^s. The scalable sparsification parameter can reduce or increase the density of the local structure to compute the overall topology in lower dimensional space. This approach provides ways to construct sparsified complexes similar to the Delaunay complex at topological sub-spaces but without computation costs of Delaunay triangulation in higher data dimensions. The approach is unable to provide the sparsification at the highest dimension and is different from β-sparsification discussed in [36]. The key advantage of this approach is that its exploration space is dependent only on point cloud size for a given homology dimension and is independent of the dimension of the data dimension. This enables the computation of lower dimensional homologies for functional genomics big high dimensional data sets with thousands of points in $\mathbb{R}^{1000} - \mathbb{R}^{30000}$.

The remainder of this paper is organized as follows: Section II introduces the *VR*-complex, *Delaunay complex* and *Clique complexes*. In addition, the *homology* and *co-homology* computation are described. Section III discusses the existing sparsification sub-sampling and dimensionality reduction techniques for PH. Section IV details the construction of the sparsified lower dimensional simplicial complex. Section V contains an experimental analysis of sub-spatial sparsified complexes. Finally, Section VI provides concluding remarks about the merits of the proposed work and future directions.

II. Background

Simplicial complexes: The computation of persistent homology require construction of a *simplicial complex* from the original point cloud data. The simplicial complex represents vertices as 0-simplex, edges as 1-simplices, triangles as 2-simplices, tetrahedrons as 3-simplices, and so on for the higher dimensional counterparts. The *Vietoris-Rips*, *Delaunay*, and *Clique* complex are of interest to the study of this paper.

VR-complexes are a purely combinatorial simplicial complex that explores all possible vertex combinations to form simplices. Generally, the size of *VR*-complex is restricted with threshold parameter ϵ, where every simplex with maximum edge weight greater than ϵ is removed from the simplicial complex. The size of *VR*-complex grows exponentially with homological dimension. The maximum number of k-simplices in *VR*-complex are $\frac{n!}{n!(n-k)!}$, where n is the total number of points in point cloud P for unbounded ϵ-threshold [37].

In contrast, Delaunay complexes are a sparsified complex with k-simplices derived from the faces of highest order Delaunay simplices for point cloud P in $\mathbb{R}^d$. The computation of Delaunay triangulation for $d > 6$ becomes extremely expensive even for small point clouds. The maximum number of simplices in Delaunay complex is given by [26], [27] as:

$$\binom{n - \lfloor \frac{d+1}{2} \rfloor}{n-d} + \binom{n - \lfloor \frac{d+2}{2} \rfloor}{n-d} = O(n^{\lceil \frac{d}{2} \rceil}). \tag{1}$$

The impracticability of Delaunay triangulation for point clouds in higher dimensions makes Delaunay complex an unfavorable simplicial complex for high dimensional data sets. Interestingly, the sparsified complexes including the *graph induced complex* [38] and *clique* [39] complex plays important role in restricting the complex size to reasonable space complexity. The work in this paper generates a sparsified 1-skeleton from sparsified hyper-graph and expands the underlying clique complex [30].

Homology: Homological features represents *holes* and *cycles* in $\mathbb{R}^2$, *voids* and *tunnels* in $\mathbb{R}^3$ and their counterparts in higher dimensional spaces. Topological features are the homology groups of a space. Homological features represented by simplicial homology can be defined using a simplicial chain complex. A chain complex is defined as a sequence of chain groups C_n together with boundary homomorphisms δ_i between the chain groups. That is:

$$\cdots \xrightarrow{\delta_{n+2}} C_{n+1} \xrightarrow{\delta_{n+1}} C_n \xrightarrow{\delta_n} C_{n-1} \xrightarrow{\delta_{n-1}} \cdots \tag{2}$$

The chain complex has the property that $\delta_i \circ \delta_{i+1} = 0$ and thus $im(\delta_{n+1}) \subset ker(\delta_n)$. That is, the image ($im$) of the homomorphism is included in the kernel (ker) of the next homomorphism. Elements of the image are boundaries, and elements of the kernel are cycles. The n^{th} homology group in the chain complex is defined to be the quotient [40].

$$H_n(C) = \frac{ker(\delta_n)}{im(\delta_{n+1})}. \tag{3}$$

Co-homology: Co-homology groups are dual of homological groups [41]. Co-homology groups are sequence of groups with origins in algebraic topology rather than geometry. Co-homology groups are important in practice as chain complex can grow from lower order simplices to higher order simplices. Thus, given the chain complex of Equation 2 and a group G, the co-chains C_n^* are defined to be the respective groups of all homomorphisms from C_n to G:

$$C_n^* = Hom(C_n, G). \tag{4}$$

The co-boundary map $\partial_n : C_{n-1}^* \rightarrow C_n^*$ is then the dual to the δ_n homomorphism as the mapping of ϕ as $\partial_n(\phi) = \delta_n^*(\phi)$. For an element $c \in C_n$ and a homomorphism $\phi \in C_{n-1}^*$, we have

$$\partial_n \phi(c) = \phi(\delta_n c). \tag{5}$$

Because $\delta_n \circ \delta_{n+1} = 0$, it is easily seen that $\partial_{n+1} \circ \partial_n = 0$. In other words, $im(\partial_n) \subset ker(\partial_{n+1})$. With this fact we can define the n^{th} co-homology group as the quotient [42], [43]:

$$H_n(C; G) = \frac{ker(\partial_{n+1})}{im(\partial_n)}. \tag{6}$$

Topological features in this paper are computed using the co-homology computation framework of LHF [44].

III. Related Work

Dimensional reduction techniques are commonly used to compute the PH for lower dimensional topological features of higher dimensional data sets [45]. Most of the data reduction techniques used are based on density and spectral analysis

(*e.g.,* IsoMap [46], Laplacian eigenmaps [47] and kernelPCA [48]). Several non-linear dimensionality reduction techniques have also been studied to extract and visualize topological features in higher dimensions. These works compute the local structure by using probability, spectral, and density based estimates. The dimensionality reduction technique for the lower dimensional PH computation has been known to alter the manifold homology [49].

k-nearest neighbor based proximity graphs have also been analyzed by *Takahashi et al* [50]. Density distribution based topological analysis has been performed by sampling on the Gabriel graphs of the points cloud [51]–[53]. The Gabriel graphs are proximity graphs embedded on a point cloud and are obtained from β-skeleton neighborhood graphs for $\beta = 1$, among other interesting embeddings for $\beta \neq 1$ [54]. Three dimensional generalization of β-skeletons have been attempted by *Hiyoshi et al* [55]. The approach of this paper is fundamentally different from the previous dimensionality reduction approaches. Moreover, the approach overcomes the limitation of linear edge based graph neighborhood embedding by using hyper-graph embedding.

In this paper, the proximity hyper-graphs are induced using neighborhood relationships based on coverage region (Section IV). The hyper-graph induction requires d-Ball based neighborhood evaluation. Several data structures exist to efficiently compute neighboring points including kd-trees [56], [57], ball-tree [58], [59], k-NN based nearest neighbors [60] and Locality Sensitive Hashing (LSH) [61]. Tree based neighborhood evaluation suffers in higher dimensions as minimum number of points required for efficient partitioning should be $\gg 2^d$. This requirement make brute force approaches outperform tree based algorithms in higher dimensions [62]. For higher dimensional data sets machine learning and hashing based approaches are often preferred [63]. For the purpose of this paper kd-tree based evaluation is performed with a scope to be expanded with machine learning and LSH based approaches.

IV. Overview of Approach

This section explains the subspace-based generation of a *reduced sparsified simplicial complex*. When computing homologies up to $H(k\text{-}1)$ features, it is sufficient to generate simplicial complex up to k-simplices. The approach in this section induces k-uniform hyper-graph, where $k < d$, on d-dimensional data to compute lower dimensional $H(k\text{-}1)$-homologies. The maximum number of ordered k-simplices that can exist is $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, where n in the size of point cloud. This count is same as number of simplices in *VR*-complex expanded to k-dimensions with $\epsilon = \infty$. The number of Delaunay k-simplices for data in d-dimension require computation of Delaunay triangulation in ambient dimension d and is extremely prohibitive for $d > 6$. To efficiently identify lower dimensional homologies its is thus required to have an efficient mesh induction algorithm on point cloud that can preserve underlying topology. In this section, a simple but scalable generalized approach is described to induce lower simplicial mesh on high dimensional point cloud.

Throughout the remainder of this paper the following terms will be used without further definition:

- P, the input point cloud in $\mathbb{R}^n$,
- β^s represents a subspace sparsification intensity,
- k-simplex and k-hyperedge are equivalent, and
- S_k, set of all k-dimensional simplices of P for $k < d$.

A. Definition: Hyper-graph Induction

There are two possibilities for hyper-graph induction, namely: d-Ball based, and d-Lune based (the d-Lune based technique is valid only if the sparsification factor $\beta^s > 1$). Both techniques define a region R that cause the k-hyperedge to be removed from the graph if any point of the k-hyperedge lie in R. The region R is defined as either the intersection or union of a set of d-Balls (intersection or union is based on β^s and the specific method used). The radii of the d-balls and the intersecting rules to define R are presented below; their centers have complex definitions that are best described by the pseudo-code in Algorithms 1 and 2.

There are two basic techniques for defining the region R, namely: d-Ball based, and d-Lune based; which to use is determined by the user. Given a point cloud P in $\mathbb{R}^d$ and a sparsification parameter $0 < \beta^s < \infty$, a k-hyper-edge with $k+1$-points $(v_0, v_1 \cdots v_k) \in P$ and circum-center HE_{CC} is valid, iff no point from $P \backslash (v_0, v_1 \cdots v_k)$ belongs to the region R$((v_0, v_1 \cdots v_k),\beta^s)$ defined as:

- ***d*-Ball Based:**
  - For $\beta^s \leq 1$, the intersection of 2 d-Balls with radii HE_{CC}/β^s
  - For $\beta^s > 1$, the union of 2 d-Balls with radii $HE_{CC} * \beta^s$
- ***d*-Lune Based**: (valid only if $\beta^s > 1$)
  - The intersection of k d-Balls with radii $\beta^s * HE_{CC}$.

Algorithm 1 computes the d-Ball based β^s-centers for given hyper-edge and a sparsification factor β^s. The d-Balls centers and radius remains same for β^s and its reciprocal $\frac{1}{\beta^s}$, the difference occur during evaluation of the hyper-edge validity (Algorithm 3). Thus, if $\beta^s \leq 1$, Line 3 of Algorithm 1 redefines β^s as its reciprocal. With that change, the remaining computation is independent of the value of β^s. The remainder of the algorithm proceeds as follows. Line 4–6, computes the k-hyperplane coefficients ($hpCoff$), circum-center(HE_{CC}) and circum-radius (HE_{CR}) for the k-hyper-edge. Line 7, computes the perpendicular distance from the k-hyper-edge circum-center to the $beta$-centers corresponding to β^s-radius equal to $\beta^s \cdot HE_{CR}$. Lines 8–9 compute the hyper-planes parallel to the k-hyper-edge hyperplane passing through the upper and lower β^s-centers ($\cdot$ computes a vector dot-product and $(||\vec{a}||)$ is the magnitude of a). Lines 13–15, computes the upper and lower β^s-center coordinates using the variables computed in line 10–11. Finally at line 16, the algorithm return the β^s-centers and β^s-radius. The computation of β^s-centers requires the intermediate sub-space embedding of k-simplex in k-dimensional space. This step requires PCA and inverse PCA transforms that are not shown in Algorithm 1.

Algorithm 1 d-Ball Based Beta-Centers Computation.

Input: β^s, Hyper-Edge$(v_0, v_1 \cdots v_k)$
Output: upper and lower β^s-$Centers$, β^s-radius

1: **function** β^s_{dBall}CENTERS(β^s, $Hyper\text{-}Edge(v_0, \cdots v_k)$)
2: **if** $\beta^s \leq 1$ **then**
3: $\beta^s \leftarrow \frac{1}{\beta^s}$
4: hpCoff $\leftarrow$ HyperPlane(Hyper-Edge$(v_0, \cdots v_k)$)
5: $\mathrm{HE_{CC}} \leftarrow$ circumCenter(Hyper-Edge$(v_0, \cdots v_k)$)
6: $\mathrm{HE_{CR}} \leftarrow$ circumRadius(Hyper-Edge$(v_0, \cdots v_k)$)
7: d $\leftarrow \sqrt{((\beta^s * \mathrm{HE_{CR}})^2 - \mathrm{HE^2_{CR}})}$
8: upperHP $\leftarrow$ -(hp$\tilde{\text{C}}$off $\cdot$ $\tilde{\mathrm{HE}}_{\mathrm{CC}}$) + d $\cdot$ ||hp$\tilde{\text{C}}$off||
9: lowerHP $\leftarrow$ -(hp$\tilde{\text{C}}$off $\cdot$ $\tilde{\mathrm{HE}}_{\mathrm{CC}}$)-d $\cdot$ ||hp$\tilde{\text{C}}$off||
10: var1 $\leftarrow \frac{(\text{-(hp}\tilde{\text{C}}\text{off}\cdot\tilde{\mathrm{HE}}_{\mathrm{CC}}\text{)-upperHP)}}{||\text{hp}\tilde{\text{C}}\text{off}||^2}$
11: var2 $\leftarrow \frac{(\text{-(hp}\tilde{\text{C}}\text{off}\cdot\tilde{\mathrm{HE}}_{\mathrm{CC}}\text{)-lowerHP)}}{||\text{hp}\tilde{\text{C}}\text{off}||^2}$
12: $\beta^s_{\text{centerlower}}, \beta^s_{\text{centerupper}} \leftarrow \emptyset$
13: **for each** coff $x, y \in hpCoff, HE_{CC}$ **do**
14: $\beta^s_{\text{centerupper}}$.insert(x $\cdot$ (var1) + y)
15: $\beta^s_{\text{centerlower}}$.insert(x $\cdot$ (var2) + y)
16: **return** $(\beta^s_{\text{centerlower}}, \beta^s_{\text{centerupper}})$, $\beta^s \cdot \mathrm{HE_{CR}}$

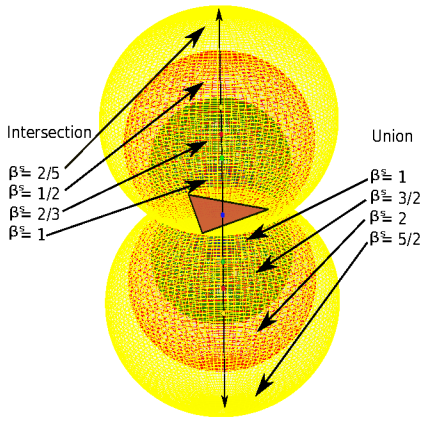


Fig. 1. The Coverage Region R grows with increasing β^s. The R d-Ball based coverage region for $0 < \beta^s \leq 1$ is computed as the intersection of two balls orthogonal to the k-simplex. For $1 < \beta^s$ the union of two balls orthogonal to the k-simplex defines R.

Figure 1, shows the two intersecting balls up and below the hyper-edge plane for 8 values of β^s. The blue, green, red and yellow balls corresponds to $\beta^s = 1, \frac{1}{1}$, $\beta^s = 1.5, \frac{2}{3}$,$\beta^s = 2, \frac{1}{2}$ and $\beta^s = 2.5, \frac{2}{5}$ respectively. For $\beta^s = 1$, the two balls (upper and lower) overlaps and represents the circum-sphere of the 2-simplex. At this β^s value the intersection and union of two balls is identical. As sparsification grows (reduces) further away from 1, the corresponding union (intersection) of the two balls increases (decreases). This strategy provides a continues spectrum of induced k-uniform hyper-graphs on point cloud P in $\mathbb{R}^d$, where $k \ll d$. The k-hyper-graph becomes complete when β^s becomes 0 and results in completely disconnected hyper-graph as β^s approaches ∞.

Algorithm 2 computes d-Lune based β^s-centers for a given hyper-edge and a sparsification factor β^s. The d-Lune based validation is only defined for $\beta^s > 1$ and algorithm stops at Line 4 otherwise. Line 5–6, computes the circum-center(HE_{CC}) and circum-radius (HE_{CR}) corresponding to k-hyper-edge. Line 9–10, computes the β^s-center and β^s-radius corresponding to each vertex of the k-hyper-edge. The lines 11–12, collects the β^s-centers. Finally at line 13, the

Algorithm 2 d-Lune Based Beta-Centers Computation.

Input: β^s, hyper-edge$(v_0, v_1 \cdots v_k)$
Output: k β^s-$Centers$ and radii for each hyper-edge vertices

1: **function** β^s_{dLune}CENTERS(β^s, $Hyper\text{-}Edge(v_0, \cdots v_k)$)
2: **if** $\beta^s < 1$ **then**
3: *only defined for* $\beta^s > 1$
4: **return**
5: $\mathrm{HE_{CC}} \leftarrow$ circumCenter(Hyper-Edge$(v_0, \cdots v_k)$)
6: $\mathrm{HE_{CR}} \leftarrow$ circumRadius(Hyper-Edge$(v_0, \cdots v_k)$)
7: $\beta^s_{\text{centers}}, \beta^s_{\text{radii}} \leftarrow \emptyset$ ▷ Circumcenter and Radius
8: **for each vertex** $v_i \in$ Hyper−Edge$(v_0, v_1 \cdots v_k)$ **do**
9: $\beta^s_c \leftarrow (\beta^s \cdot \mathrm{HE_{CR}}\text{-}\mathrm{HE_{CR}} + 1) \cdot (\mathrm{HE_{CC}}\text{-v}_i) + \mathrm{v}_i$
10: $\beta^s_r \leftarrow$ distance(v_i, β^s_c)
11: β^s_{centers}.insert(β^s_c)
12: β^s_{radii}.insert(β^s_r)
13: **return** $\beta^s_{\text{centers}}, \beta^s_{\text{radii}}$

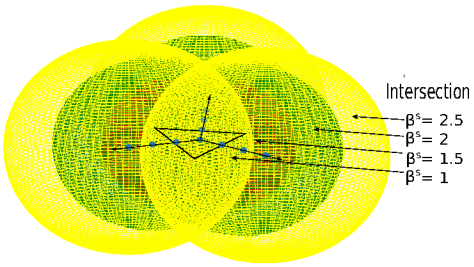


Fig. 2. Coverage Region increases with increase in the sparsification factor β^s. k-Lune Based coverage region $1 \leq \beta^s < \infty$ is shown as intersection of k-Balls centered on k-simplex hyperplane

algorithm return the β^s-centers and β^s-radii. For $\beta^s = 1$, all the β^s-centers coincides with k-hyper-edge circum-center and their intersection corresponds to k-hyper-edge circum-sphere. As β^s grows the d-Lune region grows in hyper-volume orthogonal to its hyperplane. As β^s grows away from 1, the k-hyper-graph edges experience reduction and eventually saturates; the complex does not necessary collapse all hyper-edges (the d-Ball technique always will).

Figure 2, shows the k-intersecting balls based d-Lune for 4 values of β^s. The blue d-Balls (not visible due to overlapping) corresponds to $\beta^s = 1$ and represents k-hyper-edge circum-sphere. The intersection of three red, green and yellow balls represents the d-Lune for $\beta^s = 1.5, \beta^s = 2$ and $\beta^s = 2.5$.

B. Hyper-edge Validation

Utilizing the functions of Algorithms 1 and 2, the evaluation criteria for hyper-edge validity is described in Algorithm 3 (a valid hyper-edge is preserved, an invalid one is removed). As in Algorithm 2, the k-Lune based criteria is not defined when $\beta^s < 1$ (and Line 4 terminates the algorithm accordingly). For $\beta^s > 1$ the k-Lune based validation is evaluated in Lines 6–10. The β^s-centers and β^s-radii from Algorithm 2 are used at Line 7. The intersection of all k-hyper-balls corresponding to hyper-edge vertices is than evaluated. The algorithm at Line 10 return *true* if no point of the point cloud(P) is present in k-Lune region; otherwise it returns *false*.

The algorithm evaluates the d-Ball based validation in Lines 11–17. The algorithm computes β^s-centers and β^s-radius for the upper and lower β^s-hyper-sphere at Line 12. The intersection of the two hyper-balls is checked for $\beta^s \leq 1$ and

Algorithm 3 Hyper-edge validity.

Input: P in $\mathbb{R}^d$, β^s, Hyper-Edge($v_0, v_1 \cdots v_k$), Rule R
Output: true if valid otherwise false

1: **function** β^s - VALIDATION(P, β^s,Hyper-Edge($v_0, \cdots v_k$),R)
2: **if** R **is** d-Lune and $\beta^s < 1$ **then**
3: *d-Lune based rule only defined for*$\beta^s > 1$
4: **return**
5: $points \leftarrow \emptyset$
6: **if** R **is** d-Lune **then**
7: $C, R \leftarrow \beta^s_{lune}$CENTERS($\beta^s$,Hyper-Edge($v_0, \cdots v_k$))
8: **for each** $c_i, r_i \in C,R$ **do**
9: $points \leftarrow points \cap P.Ball(c_i, r_i)$
10: **return** $points \in \emptyset$? **true** : **false**
11: **else if** R **is** d-Ball **then**
12: $c, r \leftarrow \beta^s_{d-Ball}$CENTERS($\beta^s$,Hyper-Edge($v_0, \cdots v_k$))
13: **if then**$\beta^s \leq 1$
14: $points \leftarrow P.Ball(c.up, r) \cap P.Ball(c.low, r)$
15: **else**
16: $points \leftarrow P.Ball(c.up, r) \cup P.Ball(c.low, r)$
17: **return** $points \in \emptyset$? **true** : **false**

union is checked for $\beta^s > 1$ at Line 14 and 16 respectively. The algorithm then checks for the presence of point in the β^s-region and return *true* if no point is present; otherwise return *false*. For the sake of simplicity and testing convenience, the neighborhood computation for function $P.Ball$ is obtained using kd-tree at Lines 9, 14 and 16. Other, more efficient, solutions would be used in practice at higher dimensions.

C. Lower Dimensional Enumeration for $k \ll d$ and Sparsification

For a point cloud P of size n in $\mathbb{R}^d$ Euclidean space, the maximum possible 1-hyper-edges (1-simplices) and 2-hyper-edges (2-simplices) are $\frac{n(n+1)}{2}$ and $\frac{n(n-1)(n-2)}{6}$ respectively. The number of these k-hyper-edges increases with the degree (k) and can be generalized by combination $\binom{n}{k}$ = $\frac{n!}{k!(n-k)!}$ for k-hyper-edge. To compute simplicial complex up to dimension k, it is not require to conquer the data at dimension d to compute lower dimensional homological feature. For k-dimensional Betti Numbers (and the corresponding homological features), it is sufficient to conquer the data at dimension $k \ll d$. This approach is utilized by *VR*-complex as it adds all possible simplices of dimension less than k to compute homological features up to dimension k. Since the *VR*-complex adds all possible simplices its size grows exponentially with dimension. In contrast, a Delaunay complex is small for compared to *VR*-complex; however, it requires the computation of a Delaunay triangulation at the dimension of the data. This makes it infeasible to use a Delaunay complex for the computation of lower dimensional homological features for point cloud in higher dimensions. The approach of this paper provides a topology sensitive reduction technique that reduces the number of simplices at the desired subspace while preserving topology.

The enumeration technique as described in Algorithm 4 inputs a point cloud P in $\mathbb{R}^d$, a sparsification factor β^s, and an edge-degree k for the hyper-graph induction. At Line 3, the Algorithm receives user input to select between the k-Ball or k-Lune methods. The algorithm generate all the possible unique combinations for k-hyper-edges at Line 4 and checks

Algorithm 4 Enumeration and Sparsification of k-hyper-edges

Input: P in $\mathbb{R}^d$,β^s Sparsification factor, k edge-degree
Output: Sparsified k Hyper-Edge set

1: **function** k-ENUMERATION(β^s, k, P)
2: $\beta^s(HE_k) \leftarrow \emptyset$
3: R $\leftarrow$ Rule-UserInput($k - Ball, k - Lune$)
4: **for** $\forall seq((p_i, \ldots p_k) \in$ P **do**
5: $HE_i \leftarrow [P_i, \cdots, P_k]_i$
6: **if** β^s-VALIDATION(P, β^s, HE_i, R) **then**
7: $\beta^s(HE_k).append(HE_i)$
8: **return** $\beta^s(HE_k)$

them for β^s-validation at Line 6. The valid hyper-edges are appended to the sparsified k-hyper-edge set and is returned at Line 8. In general, the size of the simplicial complex can be further reduced with a user-specified ϵ_{max}. However, a study of sparsified hyper-graphs coupled with ϵ_{max} filtration is beyond the scope of this paper.

V. EXPERIMENTS

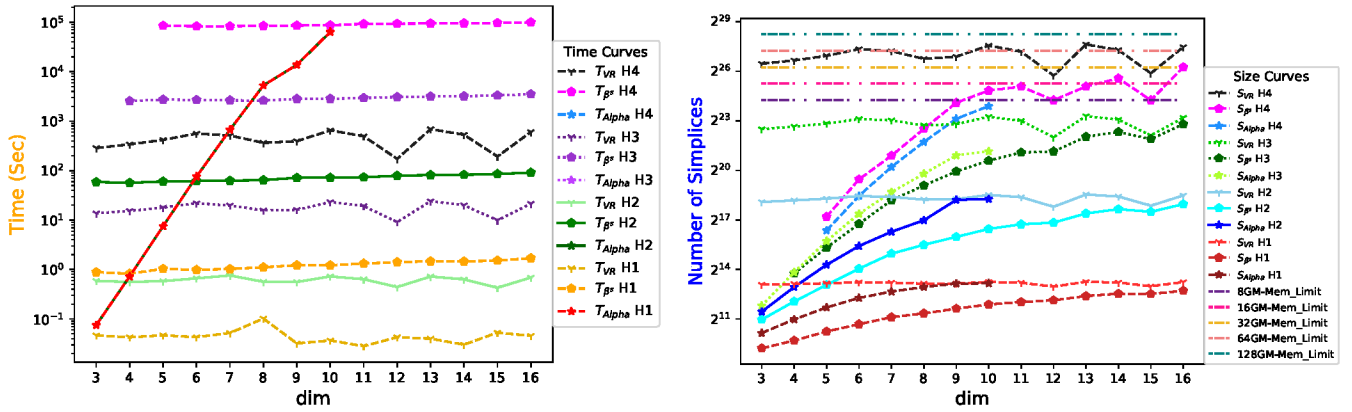
This section examines the performance (complex size, and PH run time costs) of Alpha, VR, and β^s-sparsified complexes for synthetic and real-world data sets. In particular, the following synthetic data sets are examined: (a) $d$spheres of 150 points in $\mathbb{R}^3 - \mathbb{R}^{16}$, (b) a *Permutahedron* (120 points) embedded in $\mathbb{R}^4$, and (c) a *tetraSphere* (210 points) and a *cubicSphere* (192 points) both embedded in $\mathbb{R}^5$. The real world data sets used in this study are: (a) a *forest fire* data set (517 points in $\mathbb{R}^{11}$) [64] and (b) a *concrete compressive strength* data set (1030 points in $\mathbb{R}^9$) [65]. The *forest fire* data set is a multivariate data with *temperature, wind-speed, humidity, rain* among others as factors to predict burned area during forest fire. The *concrete compressive strength* data set (CCSDS) analyze factors like *cement quantity, blast furnace slag, fly ash, water* among others to infer concrete strength. The *Permutahedron*, *tetraSphere* and *cubicSphere* data sets are generated by distributing points on their edges and projecting them on to unit d-sphere. Table I, shows the Betti numbers for each of these data sets.

TABLE I
BETTI COUNT (b_n) FOR THREE DIFFERENT SYNTHETIC DATA SETS.

Data Set	b_0	b_1	b_2	b_3	b_4	b_5
5-Tetra-Sphere	6	15	20	15	6	1
5-Cubic-Sphere	32	80	80	40	10	1
4-Permutahedron	120	240	150	30	1	-

In addition to the study of complex size and run times, this paper also presents results that the generated complexes have on the PH intervals (to give a sense of how well the various complexes preserve the topological features in the point cloud). The output of PH is analyzed using the Sliced Wasserstein distance metric to generate a dissimilarity scores [66]. The test data set in the experimental section may seem small but, given the capabilities of contemporary PH tools[1] and

[1]For example state-of-the art Delaunay computation algorithm will struggle for 1030 points in $\mathbb{R}^9$ [67].



(a) Computation time comparisons of *Alpha*, β^s, *VR*-complexes (b) Comparison of complex sizes between *Alpha*, β^s, and *VR*-complexes

Fig. 3. Time and Space comparisons with a dSphere point cloud of 150 points and noise of 0.2 in $\mathbb{R}^3 - \mathbb{R}^{16}$ for $\beta^s = 1$ for homology groups H_1 and H_4.

the higher dimensional nature of the data, they should serve as adequate tests. For the Sliced Wasserstein comparisons, the *VR*-complex PH output is used as the ground truth. The experiments including PH computations are performed using the LHF persistent homology tool chain [68] and testing was performed on an Intel(R) Xeon(R) CPU E5-1620 @ 3.70GHz with 128GB of RAM.

A. Space and Time Comparison to VR and Alpha-complex

The first study compares the PH run time and complex size differences between the β^s-sparsified complex at $\beta^s = 1$, the *VR*-complex and the *Alpha*-complex against the d-sphere test data. Figure 3a compares the computational run time of computing PH for the H_1, H_2, H_3 and H_4 homology groups. The run time for *VR*-complex is the fastest for computing $H_1 - H_4$ homology groups. In contrast, the Alpha complex run times increase exponential with dimension and fails at all dimensions $\geq \mathbb{R}^{11}$. The *VR*-complex is fast but the resulting complexes are quite large. In contrast, the run time for the β^s-sparsified complex remains proportional to the homology group being computed ($H_1 - H_4$) irrespective of the dimension of the data. Contrasting the *Alpha* and β^s-sparsified complexes, the later provides a capability to provide sparsification for high dimensional data at lower subspaces thereby overcoming the *Alpha*-complex failures in higher dimensional spaces.

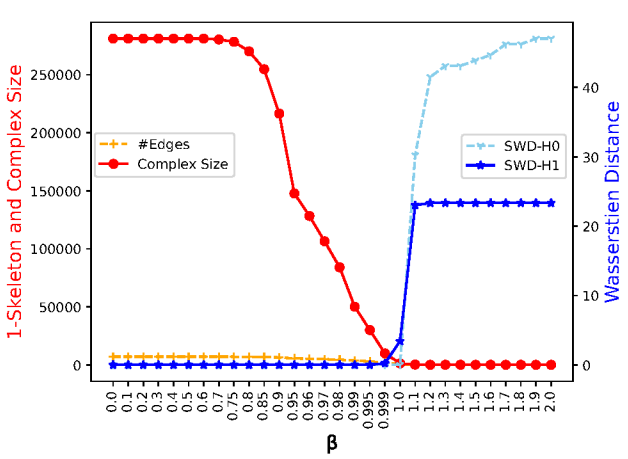
The β^s-sparsified complex allows one to develop a sparsified complexes at subspaces with sizes similar to an *Alpha*-complex with less time complexity Figure 3. In this implementation the dominant run time costs for constructing a β^s-sparsified in higher dimensions is the enumerated simplicial space and the use of the kd-tree based neighborhood. The enumerated space can be constricted by capping the ϵ_{max} value similar to that of the *VR*-complex. The kd-tree behaves worst than brute force for point clouds containing fewer than 2^d points (for P in $\mathbb{R}^d$). For kd-trees to work efficiently the size of the point cloud P should be greater than 2^d ($|P| \gg 2^d$). More efficient neighborhood search algorithms such as k-NN and LSH can keep a check on the build time.

The maximum complex size limits on 8GB, 16GB, 32GB, 64GB and 128GB RAM capacities is also shown in Figure 3b. For H_4 homology groups, the VR-complex size grows above the 64GB limit for the 150-point $d$sphere data in $\mathbb{R}^6$. In contrast, the corresponding *Alpha* and β^s-sparsified complex size remain significantly smaller with their complex size consuming less than 2GB of RAM capacity. Interesting, for $d$sphere data sets in dimension $\geq \mathbb{R}^{10}$, the *Alpha*-Complex construction run time costs prohibit its construction in reasonable time. The β^s-sparsified complex can support this computation. Furthermore, the β^s-sparsified complex construction algorithm is embarrassingly parallel and its run time complexity can be significantly reduced with multi-threading and distributed computing solutions.

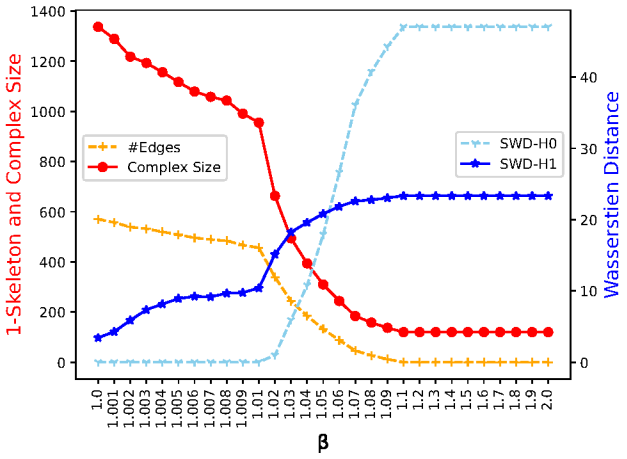
B. Complex Size and Accuracy

1) Permutahedron, Tetra-dSphere and Cubic-dSphere: As discussed in Section IV-C, the complexity of the β^s-sparsification is $O(n^k)$ for simplicial complex up to k-simplices. *Permutahedron* in $\mathbb{R}^4$ has a 4-simplex as a highest degree hyper-edge. To obtain H_1 and H_2 homology features only 2-simplex and 3-simplices are required.

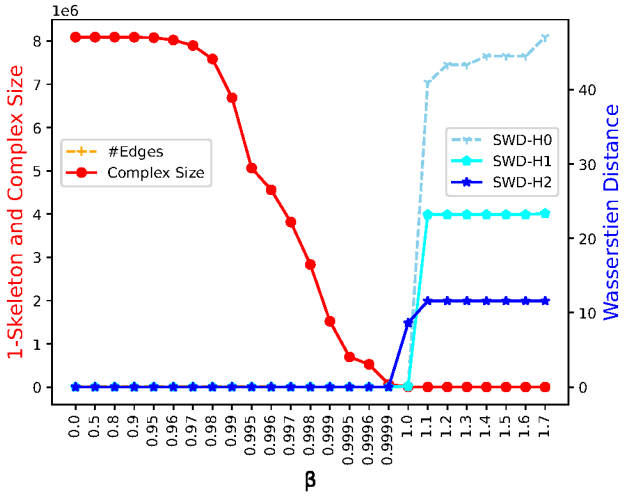
The complex reduction curves and the corresponding Sliced Wasserstein results for H_1 homologies are shown in Figure 4a and 4b. Since the d-Lune based reduction is defined only for $\beta^s > 1$, the β^s values in Figure 4b starts from 1. Similarly, for homology up to H_2 the reduction is shown in Figure 4c and 4d for d-Ball and d-Lune based proximity rules. Simplicial Complex sizes experience a monotonic reduction as β^s increases. The reduction from $0 \leq \beta^s \leq 0.8$ is somewhat slow, but accelerates quickly for $0.8 \leq \beta^s \leq 1.0$. Similar, but slightly different reduction curves are experienced for homologies computed to H_2 (Figures 4c and 4d, the Red and Orange curves). The d-Ball based sparsification criteria quickly lose its edges and becomes completely disconnected (Figures 4a and 4c). The d-Lune based sparsification experiences a steady reduction curve and does not become completely disconnected even for larger values of β^s (Figures 4b and 4d). The ad-



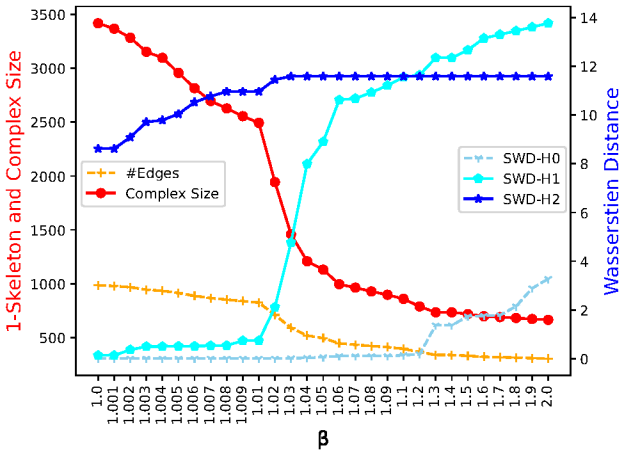
(a) Permutahedron d-Ball up to H_1 ($0 \leq \beta^s \leq 2$)



(b) Permutahedron d-Lune up to H_1 ($1 \leq \beta^s \leq 2$)



(c) Permutahedron d-Ball up to H_2 ($0 \leq \beta^s \leq 2$)



(d) Permutahedron d-Lune up to H_2 ($1 \leq \beta^s \leq 2$)

Fig. 4. Sliced-Wasserstein distance for Permutahedron in $\mathbb{R}^4$ with 120 points and noise of 0.1.

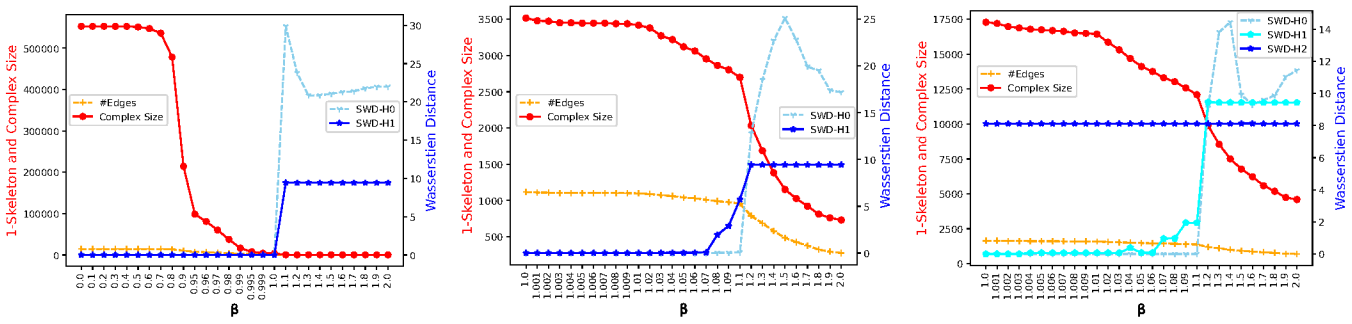
vantage of the approach for lower dimensional homologies computation is the reduced complex size without much effect in the PH intervals.

Sliced Wasserstein Distance remains close to 0 for values of $\beta^s < 1.0$. For $\beta^s > 1.0$, the d-Ball based SW distance increase quickly compared to d-Lune based reduction (Figures 4a and 4b, the Blue and Sky Blue curves). Similar SW distance curve are experienced for H_2 homology groups (Figures 4c and 4d, Blue,Sky Blue and Cyan curves).

Results for the *TetraSphere* in $\mathbb{R}^5$ are shown in Figure 5. For 2D case, the complex experienced a significant reduction until $\beta^s = 1$ without affecting the persistence intervals. For $\beta^s > 1$, the d-Ball based approach quickly sheds most of the complex size and SW distance signifies homology collapse (Figure 5a). For d-Lune based approach, the gradual reduction is experienced until $\beta^s = 1.1$ with gradual increase in the SW distance (Figure 5b). The d-Lune based results shows that the H_2 homology features are lost even at $\beta^s = 1$ (Figure 5c); these results are similar to permutahedron results where H_2 are lost just before $\beta^s = 1$ (Figure 4c).

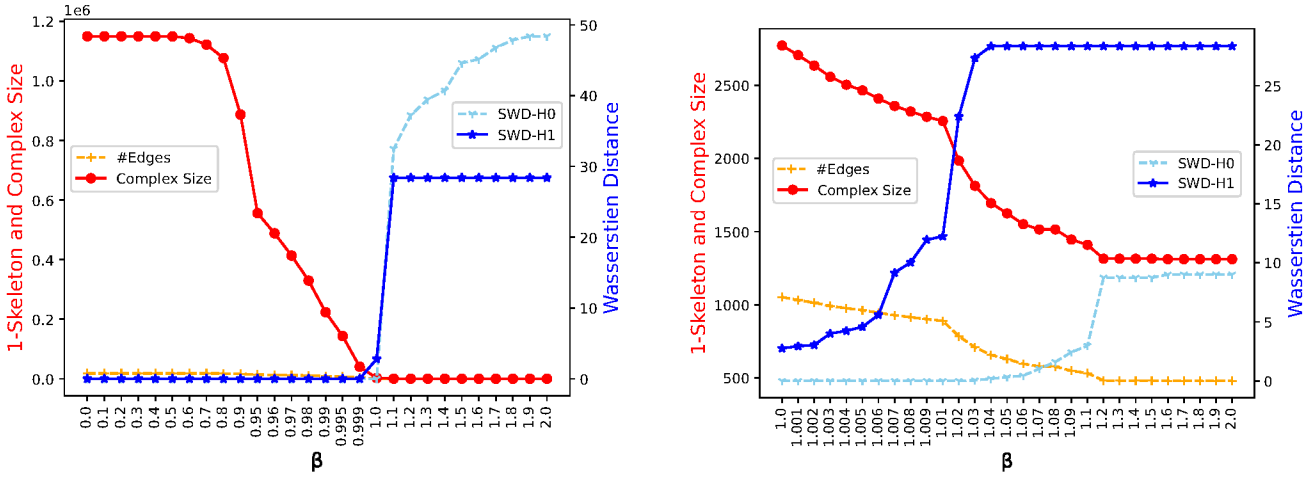
Results for the *cubicSphere* are shown in Figure 6 for homology groups to H_1. The results for *permutahedron*, *tetraSphere* and *cubicSphere* synthetic data sets shows that the sparsification approach is stable and follows a topology preserving reduction.

2) Forest Fire and Concrete Compressive Strength Data: Moving to the real world test data sets, Figure 7 shows a steady reduction in the complex size for $0 \leq \beta^s \leq 2$. The SW distance scores show an unperturbed H_0 score but considerable fluctuations in the H_1 results (Figure 7a, Blue line). An alternative comparisons using Betti numbers is shown in Table II. The counts show that the Betti numbers are preserved until $\beta^s = 1$ and quickly reduce for d-Ball based sparsification (Figure 7a), but a steady reduction for d-Lune based sparsification (Figure 7b). The SW distance increases suddenly at $\beta^s = 1.4$ and then drops again at $\beta^s = 1.5$, this variation is supported by the Betti counts where the H_1 features increases to 150 suddenly and then drops to 143.



(a) TetraSphere d-Ball up to H_1 $(0 \leq \beta^s \leq 2)$ (b) TetraSphere d-Lune up to H_1 $(1 \leq \beta^s \leq 2)$ (c) TetraSphere d-Lune up to H_2 $(1 \leq \beta^s \leq 2)$

Fig. 5. Sliced-Wasserstein distance for TetraSphere in $\mathbb{R}^5$ with 201 points and noise of 0.1.



(a) CubicSphere d-Ball up to H_1 $(0 \leq \beta^s \leq 2)$ (b) CubicSphere d-Lune up to H_1 $(1 \leq \beta^s \leq 2)$

Fig. 6. Sliced-Wasserstein distance for CubicSphere in $\mathbb{R}^5$ with 192 points and noise of 0.1 .

TABLE II
TOTAL BETTI COUNT (b_1) COMPARISON TO *VR*-COMPLEX 134 COUNT FOR FOREST FIRE DATA SET

β^s	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
b_1(d-Ball)	135	135	137	138	140	142	138	142	14	1	0	0	0	0	0	0	0	0
b_1 (d-Lune)	-	-	-	-	-	-	-	142	145	147	147	150	143	114	93	77	54	36

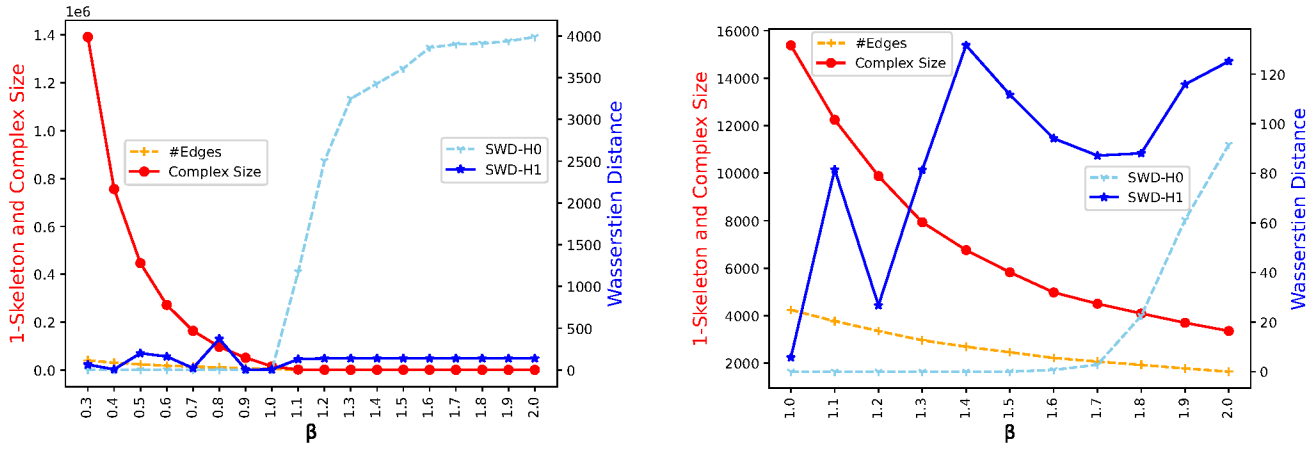
The *CCSDS* real world test data in $\mathbb{R}^9$ has been evaluated for its H_1 homology features. The results of Figure 8 show that the homology is preserved until $\beta^s = 1$ with both H_0 and H_1 SW-distance close to 0. The H_1 SW distance increases as β^s becomes greater than one and signify disconnected loops. The H_0 homology results shows a very gradual increase in SW distance indicating that the minimum spanning tree edges are well preserved.

VI. CONCLUSION AND FUTURE WORK

The computation of persistent homology is challenging for big data sets in higher dimensional spaces. This paper presents a technique to embed lower dimensional hyper-graphs on higher dimensional point clouds to compute lower dimensional homologies. The complex size is reduced at the subspace to compute lower homologies rather that conquering the data in the ambient space. The proposed approach overcomes the infeasible Delaunay triangulation computation in higher dimensional spaces to generate a sparsified complex. The proximity hyper-graphs can be embedded at a subspace sufficient to compute lower homologies. The experimental results show that the lower dimensional homologies are well preserved under subspace sparsification. The approach will enable the computation of PH for moderately big data sets in higher dimensions.

REFERENCES

[1] F. Chazal and B. Michel, "An introduction to topological data analysis: Fundamental and practical aspects for data scientists," Oct. 2017.



(a) Forest Fire d-Ball up to H_1 ($0 \leq \beta^s \leq 2$)

(b) Forest Fire$d$-Lune up to H_1 ($1 \leq \beta^s \leq 2$)

Fig. 7. Sliced-Wasserstein distance for Forest Fire Data Set in $\mathbb{R}^{11}$ with 517 points.

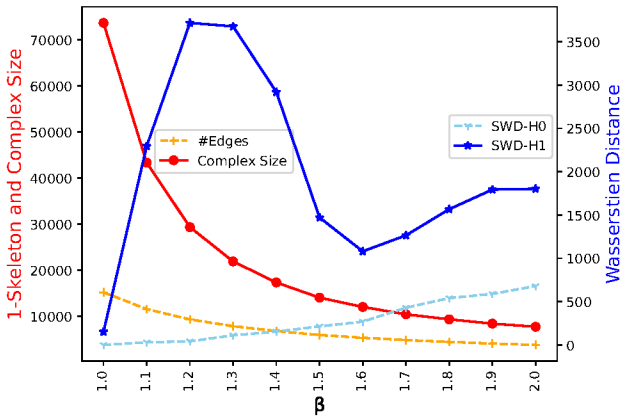


Fig. 8. Sliced-Wasserstein distance for Concrete Compressive Strength Data Set in $\mathbb{R}^9$ with 1030 points d-Lune ($1 \leq \beta^s \leq 2$) up to H_1 homologies.

[2] H. Edelsbrunner and J. Harer, "Persistent homology — a survey," *Surveys on Discrete and Computational Geometry*, vol. 453, pp. 257–282, 2008.

[3] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, "A roadmap for the computation of persistent homology," *EPJ Data Science*, vol. 6, no. 1, Aug. 2017.

[4] U. Fugacci, S. Scaramuccia, F. Iuricich, and L. D. Floriani, "Persistent homology: a step-by-step introduction for newcomers," in *Smart Tools and Apps for Graphics – Eurographics Italian Chapter Conference*, G. Pintore and F. Stanco, Eds. The Eurographics Association, 2016, pp. 1–10.

[5] H. Edelsbrunner and J. Harer, *Computational Topology, An Introduction*. American Mathematical Society, 2010.

[6] A. Zomorodian and G. Carlsson, "Computing persistent homology," *Discrete Comput Geom*, vol. 33, no. 2, pp. 249–274, Feb. 2005.

[7] I. Chevyrev, V. Nanda, and H. Oberhauser, "Persistence paths and signature features in topological data analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 1, pp. 192–202, Jan. 2020.

[8] V. Kovacev-Nikolic, P. Bubenik, D. Nikolić, and G. Heo, "Using persistent homology and dynamical distances to analyze protein binding," *Statistical applications in genetics and molecular biology*, vol. 15, no. 1, pp. 19–38, 2016.

[9] K. Xia and G.-W. Wei, "Persistent homology analysis of protein structure, flexibility, and folding," *Int Journal for Numerical Methods in Biomedical Engineering*, vol. 30, no. 8, pp. 814–844, 2014.

[10] ——, "A review of geometric, topological and graph theory apparatuses for the modeling and analysis of biomolecular data," 2016.

[11] P. G. Camara, D. I. S. Rosenbloom, K. J. Emmett, A. J. Levine, and R. Rabadan, "Topological data analysis generates high-resolution, genome-wide maps of human recombination," *Cell systems*, vol. 3, no. 1, pp. 83–94, 2016.

[12] P. G. Cámara, "Topological methods for genomics: present and future directions," *Current Opinion in Systems Biology*, vol. 1, pp. 95–101, 2017.

[13] A. H. Rizvi, P. G. Camara, E. K. Kandror, T. J. Roberts, I. Schieren, T. Maniatis, and R. Rabadan, "Single-cell topological rna-seq analysis reveals insights into cellular differentiation and development," *Nature biotechnology*, vol. 35, no. 6, pp. 551–560, 2017.

[14] X. Zhu, "Persistent homology: An introduction and a new text representation for natural language processing," in *Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI '13. AAAI Press, Aug. 2013, pp. 1953–1959.

[15] E. Berry, Y.-C. Chen, J. Cisewski-Kehe, and B. T. Fasy, "Functional summaries of persistence diagrams," *Journal of Applied and Computational Topology*, vol. 4, pp. 211–262, Mar. 2020.

[16] C. Moon, S. A. Mitchell, J. E. Heath, and M. Andrew, "Statistical inference over persistent homology predicts fluid flow in porous media," *Water Resources Research*, vol. 55, no. 11, pp. 9592–9603, 2019.

[17] V. Maroulas, F. Nasrin, and C. Oballe, "A bayesian framework for persistent homology," *SIAM Journal on Mathematics of Data Science*, vol. 2, no. 1, pp. 48–74, Feb. 2020.

[18] A. Bukkuri, N. Andor, and I. K. Darcy, "Applications of topological data analysis in oncology," *Frontiers in Artificial Intelligence*, vol. 4, p. 38, 2021.

[19] M. Joshi and D. Joshi, "A survey of topological data analysis methods for big data in healthcare intelligence," *International Journal of Applied Engineering Research*, vol. 14, no. 2, pp. 584–588, 2019.

[20] S. Mandal, A. Guzmán-Sáenz, N. Haiminen, S. Basu, and L. Parida, "A topological data analysis approach on predicting phenotypes from gene expression data," in *Algorithms for Computational Biology*, ser. Lecture Notes in Computer Scince, C. Martín-Vide, M. A. Vega-Rodríguez, and T. Wheeler, Eds., vol. 12099. Springer International Publishing, 2020, pp. 178–187.

[21] N. Sauerwald, Y. Shen, and C. Kingsford, "Topological data analysis reveals principles of chromosome structure in cellular differentiation," in *19th International Workshop on Algorithms in Bioinformatics (WABI 2019)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), K. T. Huber and D. Gusfield, Eds., vol. 143. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 23:1–23:16.

[22] I. Obayashi, T. Nakamura, and Y. Hiraoka, "Persistent homology analysis for materials research and persistent homology software: Homcloud," *Journal of the Physical Society of Japan*, vol. 91, no. 9, p. 091013, 2022.

[23] P. Skraba, V. De Silva, and M. Vejdemo-Johansson, "Topological analy-

sis of recurrent systems," in *NIPS 2012 Workshop on Algebraic Topology and Machine Learning, December 8th, Lake Tahoe, Nevada*, 2012, pp. 1–5.

[24] J.-D. Boissonnat and C. Maria, "The simplex tree: An efficient data structure for general simplicial complexes," *Algorithmica*, vol. 70, no. 3, pp. 406–427, Nov. 2014.

[25] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seideln, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. 29, no. 4, pp. 551–559, 1983.

[26] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Transactions on Graphics*, vol. 13, no. 1, pp. 43–72, Jan. 1994.

[27] H. Edelsbrunner, "Shape reconstruction with delaunay complex," in *Latin American Symposium on Theoretical Informatics*, C. L. Lucchesi and A. V. Moura, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 119–132.

[28] J. F. Espinoza, R. Hernández-Amador, H. A. Hernández-Hernández, and B. Ramonetti-Valencia, "A numerical approach for the filtered generalized čech complex," *Algorithms*, vol. 13, no. 1, 2020.

[29] M. Kerber and R. Sharathkumar, "Approximate čech complex in low and high dimensions," in *International Symposium on Algorithms and Computation*, ser. Lecture Notes in Computer Science, L. Cai, S.-W. Cheng, and T.-W. Lam, Eds., vol. 8283. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 666–676.

[30] M. Lavrov, "Complexity of constructing the clique complex of the graph — mathematics stack exchange," https://math.stackexchange.com/questions/2466075/complexity-of-constructing-the-clique-complex-of-the-graph, Oct. 2017.

[31] V. de Silva and G. Carlsson, "Topological estimation using witness complexes," in *Eurographics Symposium on Point-Based Graphics*, ser. SPBG '04, M. Gross, H. Pfister, M. Alexa, and S. Rusinkiewicz, Eds. Goslar, DEU: The Eurographics Association, 2004, pp. 157–166.

[32] A. Zomorodian, "Fast construction of the vietoris–rips complex," *Computer and Graphics*, vol. 34, pp. 263–271, Jun. 2010.

[33] S. Hornus and J.-D. Boissonnat, "An efficient implementation of delaunay triangulations in medium dimensions," INRA, Tech. Rep. RR-6743, Nov. 2008, (HAL Id: inria-00343188). [Online]. Available: https://hal.inria.fr/inria-00343188

[34] L. Alonso, J. A. Méndez-Bermúdez, and E. Estrada, "Geometrical and spectral study of β-skeleton graphs," *Phys. Rev. E*, vol. 100, Dec. 2019.

[35] N. Amenta, M. Bern, and D. Eppstein, "The crust and the β-skeleton: Combinatorial curve reconstruction," *Graphical Models and Image Processing*, vol. 60, no. 2, pp. 125–135, Mar. 1998.

[36] R. P. Singh, N. O. Malott, and P. A. Wilsey, "Topological study of β-sparsified d-uniform hypergraph based simplicial complexes," *IEEE Transactions on Knowledge and Data Engineering*, 2022, (submitted).

[37] U. Bauer, "Ripser: efficient computation of vietoris-rips persistence barcodes," 2019.

[38] T. K. Dey, F. Fan, and Y. Wang, "Graph induced complex on point data," in *Proceedings of the Twenty-Ninth Annual Symposium on Computational Geometry*, ser. SoCG '13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 107–116.

[39] A. Zomorodian, "The tidy set: A minimal simplicial set for computing homology of clique complexes," in *Twenty-Sixth Annual Symposium on Computational Geometry*, 2010, pp. 257–266.

[40] A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2002, (available online: pi.math.cornell.edu/~hatcher/AT/ATpage.html).

[41] V. de Silva, D. Morozov, and M. Vejdemo-Johansson, "Dualities in persistent (co)homology," *Inverse Problems*, vol. 27, no. 12, p. 124003, 2011.

[42] J.-D. Boissonnat, T. K. Dey, and C. Maria, "The compressed annotation matrix: an efficient data structure for computing persistent cohomology," pp. 1–12, Jan. 2020.

[43] ——, "The compressed annotation matrix: an efficient data structure for computing persistent cohomology," Apr. 2013. [Online]. Available: http://arxiv.org/abs/1304.6813

[44] "Lightweight framework for homology." [Online]. Available: https://github.com/wilseypa/lhf

[45] B. Wang, B. Summa, V. Pascucci, and M. Vejdemo-Johansson, "Branching and circular features in high dimensional data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1902–1911, 2011.

[46] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[47] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

[48] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.

[49] B. Rieck and H. Leitte, "Persistent homology for the evaluation of dimensionality reduction schemes," *Computer Graphics Forum*, 2015.

[50] S. Takahashi, I. Fujishiro, and M. Okada, "Applying manifold learning to plotting approximate contour trees," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1185–1192, 2009.

[51] K. R. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Biology*, vol. 18, no. 3, pp. 259–278, Sep. 1969.

[52] P. Oesterling, C. Heine, H. Janicke, and G. Scheuermann, "Visual analysis of high dimensional point clouds using topological landscapes," in *2010 IEEE Pacific Visualization Symposium*, ser. (PacificVis 2010). Los Alamitos, CA, USA: IEEE Computer Society, Mar. 2010, pp. 113–120.

[53] P. Oesterling, G. Scheuermann, S. Teresniak, G. Heyer, S. Koch, T. Ertl, and G. H. Weber, "Two-stage framework for a topology-based projection and visualization of classified document collections," in *2010 IEEE Symposium on Visual Analytics Science and Technology*, 2010, pp. 91–98.

[54] J. W. Jaromooczyk and G. T. Toussaint, "Relative neighborhood graphs and their relatives," *Proceedings of the IEEE*, vol. 80, no. 9, pp. 1502–1517, Sep. 1992.

[55] H. Hiyoshi, "Greedy beta-skeleton in three dimensions," in *4th International Symposium on Voronoi Diagrams in Science and Engineering*, ser. ISVD 2007. IEEE, Jul. 2007, pp. 101–109.

[56] R. A. Brown, "Building a balanced k-d tree in o(kn log n) time," Mar. 2020.

[57] K. Zhou, Q. Hou, R. Wang, and B. Guo, "Real-time KD-tree construction on graphics hardware," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 126–136, Dec. 2008.

[58] S. M. Omohundro, "Five balltree construction algorithms," *International Computer Science Institute Technical Report*, 1989.

[59] T. Liu, A. W. Moore, and A. Gray, "New algorithms for efficient high-dimensional nonparametric classification," *Journal of Machine Learning Research*, vol. 7, no. 6, pp. 1135–1158, Jun. 2006.

[60] S. Ougiaroglou and G. Evangelidis, "A simple noise-tolerant abstraction algorithm for fast k-nn classification," in *Hybrid Artificial Intelligent Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 210–221.

[61] M. Slaney and M. Casey, "Locality-sensitive hashing for finding nearest neighbors," *IEEE Signal processing magazine*, vol. 25, no. 2, p. 128, 2008.

[62] P. Ram and K. Sinha, "Revisiting kd-tree for nearest neighbor search," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 1378–1388.

[63] J. Ji and Y. Chung, "k-nn join based on lsh in big data environment," *Journal of information and communication convergence engineering*, vol. 16, no. 2, pp. 99–105, 2018.

[64] P. Cortez and A. Morais, "A data mining approach to predict forest fires using meteorological data," in *New Trends in Artificial Intelligence, 13th Portuguese Conference on Artificial Intelligence*, J. Neves, M. F. Santos, and J. Machado, Eds. Associação Portuguesa para a Inteligência Artificial, Dec. 2007. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Forest+Fires

[65] I.-C. Yeh, "Modeling of strength of high-performance concrete using artificial neural networks," *Cement and Concrete Research*, vol. 28, no. 12, pp. 1797–1808, 1998.

[66] M. Carriere, M. Cuturi, and S. Oudot, "Sliced wasserstein kernel for persistence diagrams," Nov. 2017.

[67] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, Dec. 1996.

[68] Researchers at The High Performance Computing Laboratory, "LHF: Lightweight homology framework," The University of Cincinnati, 2020. [Online]. Available: https://github.com/wilseypa/lhf