Computing the Bottleneck Distance Between Persistent Homology Transforms

Michael Kerber
Institute of Geometry, Graz University of Technology, Austria

Elena Xinyi Wang
Institute of Geometry, Graz University of Technology, Austria
University of Fribourg, Switzerland

Abstract

The Persistent Homology Transform (PHT) summarizes a shape in $\mathbb{R}^m$ by collecting persistence diagrams obtained from linear height filtrations in all directions on $\mathbb{S}^{m-1}$. It enjoys strong theoretical guarantees, including continuity, stability, and injectivity. A natural way to compare two PHTs is to use the bottleneck distance between their diagrams as the direction varies. Prior work has either compared PHTs by sampling directions or, in 2D, computed the exact *integral* of bottleneck distance over all angles via a kinetic data structure. We improve the integral objective to $\tilde{O}(n^5)$ in place of the earlier $\tilde{O}(n^6)$ bound, where n denotes the number of simplices. For the *max* objective, we give an $\tilde{O}(n^3)$ expected-time algorithm in $\mathbb{R}^2$ and an $\tilde{O}(n^5)$ expected-time algorithm in $\mathbb{R}^3$.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology; Theory of computation → Computational geometry

Keywords and phrases Kinetic data structure, bottleneck distance, persistent homology transform, vineyards

Digital Object Identifier 10.4230/LIPIcs.SoCG.2026.62

Related Version *Full Version*: https://arxiv.org/pdf/2512.00821

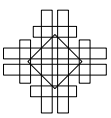
Funding *Michael Kerber*: Supported by the Austrian Science Fund (FWF) 10.55776/P33765.
Elena Xinyi Wang: Supported by the Austrian Science Fund (FWF) 10.55776/P33765.

1 Introduction

Persistent homology captures the multiscale evolution of homology classes along a filtration induced by a real-valued function on a space. It has proved useful across fields such as materials science [9, 14], computational biology [12, 19], as well as machine learning [3, 13], providing robust descriptors for high-dimensional data. A persistence diagram is a concise multiset summary of persistent homology, recording the birth–death times of features. Similarity between objects is then assessed by comparing their diagrams, most commonly via the bottleneck distance. Computationally, this reduces to a minimum-bottleneck perfect matching on a bipartite graph with diagonal copies, and algorithms for this problem have been studied extensively in both theory and practice [2, 6, 10, 15, 16].

Since a single diagram depends on the chosen filter and may miss geometry, the Persistent Homology Transform (PHT) collects diagrams from all directions to retain fuller information [20]. It assigns to a shape in $\mathbb{R}^m$ the collection of persistence diagrams obtained from linear height filtrations over all directions on $\mathbb{S}^{m-1}$. It possesses many desirable theoretical properties such as continuity, stability, and injectivity [5, 11, 20]. These properties make the PHT an effective topological descriptor for applications. However, unlike other directional-transform–based signatures such as the Euler Characteristic Transform, the PHT lacks efficient comparison algorithms, limiting its practical use. A natural way to compare PHTs is to sample directions and compute the bottleneck distance only at those angles. This

© Michael Kerber and Elena Xinyi Wang;
licensed under Creative Commons License CC-BY 4.0



inevitably leads to information loss. To integrate the bottleneck distance over the parameter domain (e.g., $\mathbb{S}^{m-1}$), prior work by Munch, Wang, and Wenk introduced a kinetic "hourglass" approach that maintains matchings as the parameter moves. It computes this integral with an $\tilde{O}(n^6)$ bound for data in $\mathbb{R}^2$ [18]. In this work, we focus on the max variant, which we define as the supremum of the bottleneck distance over all directions.

Our Contributions. We provide an $\tilde{O}(n^3)$ expected-time algorithm to compute the max bottleneck distance between PHTs of shapes in $\mathbb{R}^2$ and an $\tilde{O}(n^5)$ expected-time algorithm in $\mathbb{R}^3$, where $\tilde{O}$ means the logarithmic factor is ingored. Additionally, we improve the computation of the integral distance from the previous $\tilde{O}(n^6)$ bound to $\tilde{O}(n^5)$ in $\mathbb{R}^2$.

For the max bottleneck distance, we use techniques from computational geometry and topology and inspired by the strategy of Bjerkevik and Kerber for computing the matching distance for two-parameter modules [1]. At a high level, the solution for 2D and 3D are similar. We first reformulate the problem as a decision problem: given a threshold λ, we determine whether the max bottleneck distance is at most λ. To solve this, we sweep the direction parameter around the sphere while maintaining the bottleneck matching between the two persistence diagrams. The key insight is that this matching changes only at discrete "events": points where persistence diagrams change combinatorially or where edge weights cross the threshold λ. At each event, we update the matching using a single augmenting path search in the geometric bipartite graph, exploiting the fact that only local changes occur.

We then identify that the maximum bottleneck distance must occur at one of a finite set of candidate values, corresponding to critical points and intersections of the difference functions between simplices. While there are $O(n^4)$ candidates in 2D and $O(n^6)$ in 3D, we avoid exhaustive enumeration through a randomized incremental construction. We maintain a band containing the true maximum and iteratively refine it by processing simplices in random order, testing whether their associated candidates lie within the current band. The randomization ensures that we expect only $O(\log n)$ refinements, reducing the expected complexity to $\tilde{O}(n^3)$ in 2D and $\tilde{O}(n^5)$ in 3D.

Outline. We begin in Section 2 with a brief overview of persistent homology, persistence diagrams, and the PHT, and we formalize the problem. Section 3 presents our $\mathbb{R}^2$ algorithm: first at a high level, then with full technical details. The improvement for the integral distance is shown in Section 4. Section 5 extends the approach to $\mathbb{R}^3$ before concluding in Section 6.

2 Preliminary

2.1 Persistent homology

Persistent homology is a multi-scale summary of the connectivity of objects in a nested sequence of subspaces; see [7, 8] for an introduction. For the purposes of this section, we define a *persistence diagram* to be a finite collection of points $\{(b_i, d_i)\}_i$ with $d_i \geq b_i$ for all i.

Given two persistence diagrams X and Y, a partial matching is a bijection $\eta : X' \to Y'$ on a subset of the points $X' \subseteq X$ and $Y' \subseteq Y$; we denote this by $\eta : X \rightleftharpoons Y$. The cost of a partial matching is the maximum over the L_∞-norms of all pairs of matched points and the distance between the unmatched points to the diagonal:

$$c(\eta) = \max\left(\{\|x - \eta(x)\|_\infty \mid x \in X'\} \cup \{\tfrac{1}{2}|z_2 - z_1| \mid (z_1, z_2) \in (X \setminus X') \cup (Y \setminus Y')\}\right) \quad (1)$$

and the bottleneck distance is defined as $d_B(X, Y) = \inf_{\eta : X \rightleftharpoons Y} c(\eta)$.

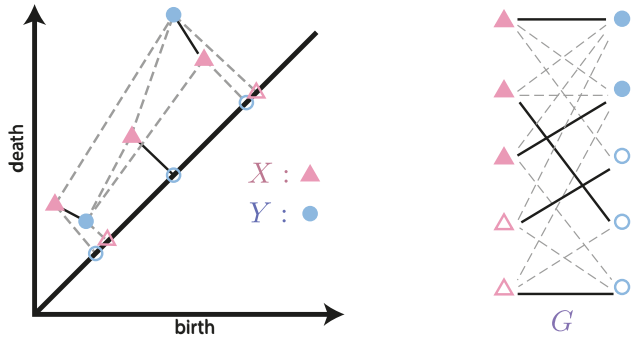


Figure 1 Construction of the bipartite graph G based on the persistence diagrams X and Y.

We can reduce finding the bottleneck distance between persistence diagrams to the problem of finding the bottleneck cost of a bipartite graph. Let X and Y be two persistence diagrams given as finite lists of off-diagonal points. For any off-diagonal point $z = (z_1, z_2)$, the orthogonal projection to the diagonal is $z' = ((z_1 + z_2)/2, (z_1 + z_2)/2)$. Let $\overline{X}$ (resp. $\overline{Y}$) be the set of orthogonal projections of the points in X (resp. Y). Set $U = X \sqcup \overline{Y}$ and $V = Y \sqcup \overline{X}$. We define the complete bipartite graph $G = (U \sqcup V, U \times V, \mathrm{c})$, where for $u \in U$ and $v \in V$, the weight function c is given by

$$\mathrm{c}(uv) = \begin{cases} \|u - v\|_\infty & \text{if } u \in X \text{ or } v \in Y \\ 0 & \text{if } u \in \overline{X} \text{ and } v \in \overline{Y}. \end{cases}$$

An example of the bipartite graph construction is shown in Figure 1. This graph can be used to compute the bottleneck distance of the input diagrams because of the following lemma.

▶ **Lemma 1** (Reduction Lemma [8]). *For the above construction of G, $d_B(G) = d_B(X, Y)$.*

2.2 Persistent homology transform

Let K be a finite simplicial complex embedded in $\mathbb{R}^m$. For any direction $\omega \in \mathbb{S}^{m-1}$, the height function $h_\omega : |K| \to \mathbb{R}$ is defined by $h_\omega(x) = \langle x, \omega \rangle$. The lower-star filtration induced on the abstract complex K is given by:

$$h_\omega(\sigma) = \max\{h_\omega(v) \mid v \in \sigma\},$$

and the sublevelset at height a is the full subcomplex induced by vertices $\{v \in K \mid h_\omega(v) \leq a\}$. Filtering K by h_ω produces a persistence diagram $\mathrm{Dgm}_k(h_\omega^K)$. We focus on $k = 0, 1$ for $m = 2$, and $k = 0, 1, 2$ for $m = 3$.

The *persistent homology transform (PHT)* is defined as:

$$\mathrm{PHT}(K) : \mathbb{S}^{m-1} \to \mathcal{D}, \quad \omega \mapsto \mathrm{Dgm}(h_\omega^K),$$

where $\mathcal{D}$ is the space of persistence diagrams. The PHT is injective [20, 5], making it a faithful shape representation.

2.3 Maximum bottleneck distance

In this work, we are interested in comparing two finite embedded simplicial complexes, $K, K' \in \mathbb{R}^m$, $m = 2$ or 3, by comparing their PHTs. We define the *maximum bottleneck distance* between K and K' to be:

$$d_\infty(\mathrm{PHT}(K), \mathrm{PHT}(K')) = \max_{\omega \in \mathbb{S}^{m-1}} d_B\left(\mathrm{Dgm}(h_\omega^K), \mathrm{Dgm}(h_\omega^{K'})\right).$$

This distance is a variant of a similar integral distance that has been studied previously by Munch, Wang, and Wenk [18]:

$$d_1(\mathrm{PHT}(K), \mathrm{PHT}(K')) = \int_{\mathbb{S}^{m-1}} d_B\left(\mathrm{Dgm}(h_\omega^K), \mathrm{Dgm}(h_\omega^{K'})\right) d\omega.$$

Both d_∞ and d_1 inherit stability from the bottleneck distance. However, they differ from a computational perspective. Computing d_1 requires full knowledge of the whole domain, which is inevitably expensive. Our d_∞ metric seeks the maximum value, which allows us to reformulate the problem as finding the supremum over a finite set of candidate values rather than tracking the bottleneck distance continuously.

A key property that makes our algorithmic approach feasible is that the bottleneck distance function is Lipschitz continuous, which is a direct consequence of the stability of the bottleneck distance (see the full version for a complete proof):

▶ **Lemma 2.** *Let $H(\omega) := d_B\big(\mathrm{Dgm}(h_\omega^K), \mathrm{Dgm}(h_\omega^{K'})\big)$ denote the bottleneck distance between persistence diagrams. Then the function $H : \mathbb{S}^{m-1} \to \mathbb{R}$ is Lipschitz continuous with constant $R_K + R_{K'}$, where $R_K = \max_{v \in K^{(0)}} \|v\|$ and $R_{K'} = \max_{v \in (K')^{(0)}} \|v\|$.*

3 Computing d_∞ in $\mathbb{R}^2$

Our setup in this section is as follows: we are given two geometric simplicial complexes K, K' in $\mathbb{R}^2$ with at most n simplices each. The goal is to compute d_∞: we first introduce the necessary notation, then give a high-level overview, and finally present the technical details.

3.1 Events

For each simplex $\alpha \in K \cup K'$, we define an insertion curve $I_\alpha(\omega)$. These curves record the height at which each simplex enters the sublevel-set filtration induced by direction $\omega \in S^1$. Explicitly, if the k-simplex α has vertices $\{v_0, \ldots, v_k\} \subseteq V(K)$, its insertion curve is

$$I_\alpha(\omega) := \max_{0 \le i \le k} h_\omega(v_i), \quad h_\omega(x) = \langle x, \omega \rangle.$$

Each I_α is a piecewise-trigonometric function on S^1.

There are at most $2n$ such curves. We now define *difference curves* that capture both matched and unmatched contributions to the bottleneck distance. For distinct simplices α, σ:

$$\Delta_{\alpha,\sigma}(\omega) := \begin{cases} |I_\alpha(\omega) - I_\sigma(\omega)| & \text{if } \alpha, \sigma \text{ are from different complexes} \\ \frac{1}{2}|I_\alpha(\omega) - I_\sigma(\omega)| & \text{if } \alpha, \sigma \text{ are from the same complex} \end{cases}$$

These are real-valued functions on S^1; geometrically, one can picture their graphs as curves in $S^1 \times \mathbb{R}_{\ge 0}$. The factor of $\frac{1}{2}$ corresponds to the contribution of unmatched points to the bottleneck distance, see Equation 1. There are $O(n^2)$ such curves for $|K|, |K'| = O(n)$. We call the vertex v_i achieving $\max_{0 \le j \le k} \langle v_j, \omega \rangle$ the *active vertex* of α at ω. As ω varies, the active vertex changes at discrete directions where two vertices achieve the same height. Within any interval where the active vertices remain fixed, the difference curve simplifies to either $|\langle u - w, \omega \rangle|$ (for different complexes) or $\frac{1}{2}|\langle u - w, \omega \rangle|$ (for same complex), which is a simple trigonometric function. See Figure 2 for an example. The insertion and difference curves are both well-behaved. See Figure 3 for an example.

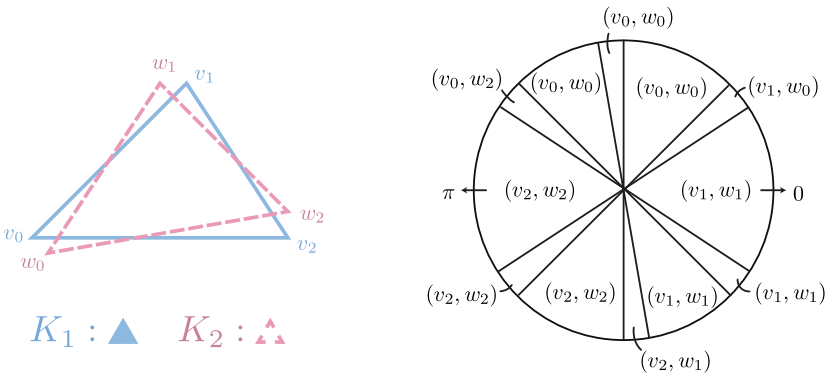


Figure 2 On the left are two simplices with vertices $v_{0,1,2}$ and $w_{0,1,2}$ respectively. On the right is $\mathbb{S}^1$ partitioned into regions with the same active vertex pairs (v_i, w_j).

▶ **Lemma 3.** *Each difference curve $\Delta_{\alpha,\sigma}$ crosses any horizontal line $y = \lambda$ at most $O(1)$ times as ω varies over $[0, 2\pi]$. Since there are $O(n^2)$ difference curves in total, this yields $O(n^2)$ crossings with any horizontal line. In particular, there are $O(n^2)$ insertion-curve intersections, as these correspond to difference curves crossing the line $y = 0$.*

The bound on insertion-curve intersections follows immediately since we have $O(n)$ curves total, giving at most $\binom{O(n)}{2} = O(n^2)$ pairwise intersections. For threshold crossings, each difference curve $\Delta_{\alpha,\sigma}$ is piecewise trigonometric with $O(1)$ pieces (as simplices in $\mathbb{R}^2$ have at most 3 vertices), and crosses any threshold λ at most $O(1)$ times. With $O(n^2)$ pairs total, this yields $O(n^2)$ threshold crossings. The detailed geometric analysis is in the full version.

In each direction ω, the bottleneck distance is realized by the value of a difference curve $\Delta_{\alpha,\sigma}(\omega)$. When this maximum is unique in a neighborhood of ω, the continuity of the bottleneck function $H(\omega) := d_B\left(\mathrm{Dgm}(h^K_\omega), \mathrm{Dgm}(h^{K'}_\omega)\right)$ ensures that the same curve or persistence value realizes the bottleneck distance in an ε-neighborhood. We call the points in S^1 where the realizing curve or value changes locally *events*, and the values H attains at events are the *candidate values*.

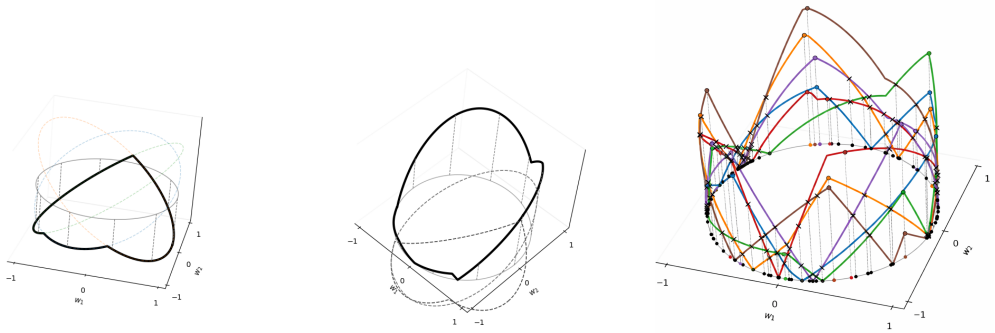
▶ **Theorem 4** (Events for d_∞ in $\mathbb{R}^2$)**.** *The direction ω^* attaining d_∞ occurs at either*

1. *a local maximum of a difference curve $\Delta_{\alpha,\sigma}$, or*
2. *an intersection of two difference curves. See Figure 3 for an illustration.*

Proof. Let ω^* be the direction that attains d_∞. By Lemma 2, if the bottleneck distance at ω^* is realized by a unique difference curve in a neighborhood, then ω^* must be a local maximum of that difference curve, giving case (1). This includes both matched pairs (when $\alpha \in K, \sigma \in K'$) and unmatched point persistence (when α, σ are from the same complex). Otherwise, at least two curves achieve the same maximum value at ω^*, giving case (2). Note that intersections of insertion curves (where the persistence diagram changes combinatorially) form a special case of (2), as when two insertion curves I_α, I_β from the same complex coincide, all difference curves of the form $\Delta_{\alpha,\sigma}$ and $\Delta_{\beta,\sigma}$ are equal. ◀

▶ Remark 5. When we refer to "intersections" of difference curves, we mean isolated transverse crossings, not intervals of coincidence. Endpoints of overlaps also count as intersections, since the realizing curve may change at those boundaries.

▶ **Corollary 6.** *There are $O(n^4)$ candidate values where d_∞ may be attained.*



(a) (b) (c)

Figure 3 (a) An insertion curve $I_\alpha(\omega)$. (b) A difference curve $\Delta_{\alpha,\sigma}(\omega)$. (c) Arrangement of four difference curves; local maxima and pairwise intersections mark the candidate values of Theorem 4.

Proof. There are $O(n^2)$ difference curves: $O(n^2)$ from pairs $(\alpha, \sigma) \in K \times K'$, plus $O(n^2)$ from pairs within the same complex. Each has $O(1)$ local maxima (see the full version for the explicit bound), contributing $O(n^2)$ type-(1) candidates. For type-(2) candidates, each pair of difference curves intersects at $O(1)$ heights, giving $\binom{O(n^2)}{2} = O(n^4)$ intersections total. Thus the total number of candidates is $O(n^4)$. ◀

▶ Remark 7. When simplices share vertices, their difference curves may partially coincide. For instance, if $\alpha, \beta \in K$ share vertex v and $\sigma, \tau \in K'$ share vertex w, then $\Delta_{\alpha,\sigma}$ and $\Delta_{\beta,\tau}$ coincide in regions where v and w are simultaneously active. Despite these overlaps, each difference curve retains a unique identity at the combinatorial level through its defining simplex pair. In particular, our candidate enumeration remains valid: if the maximum bottleneck distance is achieved at direction ω^*, then at least one combinatorially distinct difference curve must realize this maximum, and this curve's critical points and intersections are in our candidate set. The full technical detail is in the full version.

3.2 Overview of the algorithm

We describe the high-level structure of our algorithm next. On this level, our approach is similar to the algorithm of Bjerkevik and Kerber to compute the matching distance [1]. However, the approaches differ in the technical realization of the predicates which are described in the subsequent subsections.

Decision procedure. Given a threshold $\lambda \geq 0$, we decide whether $d_\infty \leq \lambda$ by sweeping ω around $\mathbb{S}^1$ and maintaining a bottleneck matching. As ω rotates, we update the persistence diagrams at $O(n^2)$ insertion-curve intersections and perform single augmenting-path updates whenever an edge weight crosses the threshold. This decision procedure runs in $\tilde{O}(n^3)$ time.

Basic algorithm. By Theorem 4, d_∞ is attained at one of $O(n^4)$ candidate values corresponding to intersections of difference curves and their local maxima. These candidates can be enumerated and sorted in $O(n^4 \log n)$ time. Combined with binary search using our decision procedure, this yields an $O(n^4 \log n + n^3 \operatorname{polylog}(n)) = \tilde{O}(n^4)$ algorithm.

Improved algorithm via band refinement. By Theorem 4, there are $O(n^4)$ total candidates where d_∞ could be attained. To achieve $\tilde{O}(n^3)$ time, we avoid enumerating all candidates.

We call a candidate an α-candidate if it arises from a difference curve $\Delta_{\alpha,\sigma}$ for some σ (when $\alpha \in K$), from $\Delta_{\tau,\alpha}$ for some τ (when $\alpha \in K'$), or from $|I_\alpha(\omega) - I_\beta(\omega)|/2$ for some β in the same complex as α.

We maintain a height band $(\lambda_1, \lambda_2]$ satisfying $d_\infty > \lambda_1$ and $d_\infty \leq \lambda_2$. We process simplices from both complexes in random order. For each simplex α, we apply:

1. Existence predicate: Tests if any α-candidate lies in $(\lambda_1, \lambda_2]$ in $O(n^2 \operatorname{polylog}(n))$ time. If no candidate exists in the band, we skip to the next simplex.
2. Band-refinement predicate: If the existence test is positive, we enumerate all $O(n^3)$ α-candidates in the band and refine via binary search in $O(n^3 \operatorname{polylog}(n))$ time.

The resulting total running time is $O(n \cdot n^2 \operatorname{polylog}(n) + C \cdot n^3 \operatorname{polylog}(n))$, where C is the number of band-refinement calls. As we will show, processing the simplices in a random order ensures $\mathbb{E}[C] = O(\log n)$, yielding expected time $\tilde{O}(n^3)$.

3.3 Decision problem

We now describe how to decide whether $d_\infty \leq \lambda$ for a given threshold $\lambda \geq 0$. The key idea is to sweep the direction ω around S^1 and check if the bottleneck distance ever exceeds λ.

By Lemma 1, at each direction ω, the bottleneck distance d_B equals the cost of a minimum-bottleneck matching in a bipartite graph $G(\omega)$ constructed from the two persistence diagrams $X = \operatorname{Dgm}(h^K_\omega)$ and $Y = \operatorname{Dgm}(h^{K'}_\omega)$ with diagonal copies. Define the subgraph $G_\lambda(\omega)$ to contain only those edges of $G(\omega)$ whose weights are $\leq \lambda$. Then $d_B(X, Y) \leq \lambda$ if and only if $G_\lambda(\omega)$ admits a perfect matching.

Our strategy is to maintain a perfect matching in $G_\lambda(\omega)$ as ω sweeps from 0 to 2π. If at any point no perfect matching exists, we conclude $d_\infty > \lambda$; otherwise, if the sweep completes with a valid matching at every direction, then $d_\infty \leq \lambda$.

Sweep algorithm. We initialize at $\omega = 0$ by computing a perfect matching in $G_\lambda(0)$ using Hopcroft-Karp in $O(n^{2.5})$ [15]; if none exists, $d_\infty > \lambda$.

As ω increases from 0 to 2π, the bipartite graph $G_\lambda(\omega)$ changes only at discrete events:

- Diagram changes: At intersections of insertion curves, points move to or from the diagonal in the persistence diagrams, updating the bipartite graph structure.
- Threshold crossings: When a difference curve $\Delta_{\alpha,\sigma}(\omega)$ crosses the threshold λ, the corresponding edge enters or leaves $G_\lambda(\omega)$.

Maintaining the matching. A matching M is maintained across the sweep. At each event:

- If a new edge enters $G_\lambda(\omega)$ (weight decreases below λ), M remains valid.
- If an edge leaves $G_\lambda(\omega)$ (weight exceeds λ) and is not in M, then M remains valid.
- If an edge $(u, v) \in M$ becomes invalid, Hall's theorem guarantees that a new perfect matching exists if and only if there is an augmenting path from the newly exposed vertices. We search for such a path in $O(n \log n)$ time using the geometric structure [10]. If found, we augment along this path to restore a perfect matching. If no augmenting path exists, then $G_\lambda(\omega)$ has no perfect matching, so $d_B > \lambda$ at this ω, implying $d_\infty > \lambda$.

At diagram change events (insertion curve intersections), we update the persistence diagrams using the vineyard algorithm [4]. When two insertion curves I_α and I_β from the same complex intersect, the relative order of exactly two simplices changes in the filtration. Following Bjerkevik and Kerber [1], this corresponds to a single transposition in the barcode pairing, which can be updated in $O(n)$ time.

If the sweep completes with a valid perfect matching at every ω, then $d_\infty \leq \lambda$.

▶ **Lemma 8.** *The decision procedure runs in $\tilde{O}(n^3)$ time.*

Proof. At each event, we either update the diagram structure in $O(n)$ time (vineyard update for insertion curve intersections) or search for a single augmenting path in $O(n \log n)$ time (for threshold crossings). With $O(n^2)$ total events from Lemma 3, the total time is $\tilde{O}(n^3)$. ◀

3.4 Candidate pruning via band refinement

Recall that we have $O(n^4)$ total candidates where d_∞ may be attained. To find d_∞ efficiently, we maintain a height band $(\lambda_1, \lambda_2]$ containing d_∞ and iteratively shrink it by processing simplices from both complexes in random order.

We distinguish between two types of intersections between difference curves. *Intra-intersections* occur when two difference curves share a common simplex: either Δ_{α,σ_1} and Δ_{α,σ_2} sharing simplex α, or $\Delta_{\beta_1,\tau}$ and $\Delta_{\beta_2,\tau}$ sharing simplex τ. *Inter-intersections* occur when $\Delta_{\alpha,\sigma}$ and $\Delta_{\beta,\tau}$ meet with all four simplices distinct: $\alpha \neq \beta$ and $\sigma \neq \tau$.

Pre-refinement. Before processing individual simplices, we refine the band to exclude certain events from its interior: let S be the set containing all local maxima of difference curves $\Delta_{\alpha,\sigma}$ for all pairs (including both different-complex and same-complex pairs), all intra-α intersections where $\Delta_{\alpha,\sigma_1}(\omega) = \Delta_{\alpha,\sigma_2}(\omega)$ for some $\omega \in \mathbb{S}^1$ and distinct σ_1, σ_2, and all intra-τ intersections where $\Delta_{\beta_1,\tau}(\omega) = \Delta_{\beta_2,\tau}(\omega)$ for distinct β_1, β_2.

Since there are $O(n^2)$ difference curves, each with $O(1)$ local maxima, we have $O(n^2)$ maxima total. For intra-intersections, each of the $O(n)$ simplices in K contributes $\binom{O(n)}{2} = O(n^2)$ pairs of curves that could intersect, with $O(1)$ intersection points each, giving $O(n^3)$ intra-α intersections. Similarly for intra-τ intersections. Thus $|S| = O(n^3)$.

We binary search with the decision procedure to find the largest value in S below d_∞ and the smallest at or above it, setting these as λ_1 and λ_2. The resulting band $(\lambda_1, \lambda_2]$ contains no intra-events in its interior. This takes $O(n^3 \operatorname{polylog} n)$ time.

3.4.1 Predicate 1: Existence test

After pre-refinement, we process simplices to detect inter-intersections.

▶ **Lemma 9.** *For fixed $\alpha \in K$, determining whether any inter-intersection involving α lies in $(\lambda_1, \lambda_2]$ takes $O(n^2 \log n)$ time.*

Proof. We seek intersections of the form $\Delta_{\alpha,\sigma} \cap \Delta_{\beta,\tau}$ where $\beta \neq \alpha$. For each $\tau \in K'$, we define the *red curves* $R_\alpha = \{\Delta_{\alpha,\sigma} : \sigma \neq \alpha\}$ as all curves involving simplex α, and the *blue curves* $B_\tau = \{\Delta_{\beta,\tau} : \beta \neq \tau\}$ as all curves involving simplex τ. (See Figure 2 for the geometric setting; the red and blue curves are real-valued functions on S^1, and we detect their intersection as real-valued crossing events while sweeping around S^1.)

The key insight from pre-refinement is that within the band $(\lambda_1, \lambda_2]$, no two red curves can intersect (these would be intra-α intersections), and no two blue curves can intersect (these would be intra-τ intersections).

Each difference curve $\Delta_{\alpha,\sigma}(\omega)$ consists of $O(1)$ trigonometric pieces, switching between different expressions at directions where the active vertex changes. Since simplices in $\mathbb{R}^2$ have at most 3 vertices, there are $O(1)$ such transition points. When we clip to the band $[\lambda_1, \lambda_2]$, each curve crosses the boundary at $O(1)$ points, subdividing into $O(1)$ monotone arcs within the band.

We apply the standard Bentley-Ottmann sweep-line algorithm to all $O(n)$ arcs. Since red arcs do not intersect each other and blue arcs do not intersect each other (by pre-refinement), any intersection must be between a red and blue arc. We terminate immediately upon detecting the first such intersection. This takes $O(n \log n)$ time per choice of τ. Testing all $O(n)$ choices of $\tau \in K'$ takes $O(n^2 \log n)$ total time for fixed α. ◀

3.4.2 Predicate 2: Band refinement

When the existence test detects an inter-intersection, we must enumerate all candidates involving α to refine the band.

▶ **Lemma 10.** *When an inter-intersection involving α is detected, we can enumerate all $O(n^3)$ α-candidates in $(\lambda_1, \lambda_2]$ and compute a refined band with no α-candidates in its interior in $O(n^3 \operatorname{polylog} n)$ time.*

Proof. After pre-refinement, only inter-intersections remain in the band's interior. For simplex α, there are $O(n^3)$ such intersections (choosing σ, β, τ with $O(n)$ choices each). We enumerate these candidates, sort them, and refine the band using binary search with the decision procedure, taking $O(n^3 \operatorname{polylog} n)$ time. ◀

▶ **Theorem 11.** *After global pre-refinement, processing each simplex α takes $O(n^2 \log n + \mathbb{1}_\alpha n^3 \operatorname{polylog} n)$ total time, where $\mathbb{1}_\alpha$ is the indicator function of the existence test.*

Proof. This follows directly from Lemma 9 and 10. ◀

3.4.3 Randomized analysis

▶ **Theorem 12.** *Processing simplices from both K and K' in uniformly random order yields $\mathbb{E}[C] = O(\log n)$ refinements, where C counts positive existence tests.*

Proof. For each simplex α, define x_α as the smallest α-candidate that is $\geq d_\infty$ and y_α as the largest α-candidate that is $< d_\infty$. Since d_∞ is realized by at least one candidate, for some simplex α^* we have $x_{\alpha^*} = d_\infty$. Every interval $(y_\alpha, x_\alpha]$ contains d_∞, so our maintained band always has the form $(y_{\alpha_i}, x_{\alpha_j}]$ for some processed simplices α_i, α_j.

When processing simplices in random order, we obtain random permutations of the sequences $\{x_\alpha\}$ and $\{y_\alpha\}$. Simplex α passes the existence test and the refinement is triggered if and only if there exists an α-candidate in the current band. This occurs when x_α is smaller than all previously seen $x_{\alpha'}$ values (making it a left-to-right minimum), or y_α is larger than all previously seen $y_{\alpha'}$ values (making it a left-to-right maximum).

Indeed, if x_α is not a minimum, then some previously processed α' satisfies $x_{\alpha'} \leq x_\alpha$, so the band's upper bound is already at most x_α, excluding all α-candidates with value $\geq x_\alpha$. Likewise, if y_α is not a maximum, the band's lower bound already exceeds y_α, excluding all α-candidates with value $\leq y_\alpha$. Thus, when neither x_α nor y_α improves the band boundaries, every α-candidate lies outside (y_α, x_α), and none can fall in the current band.

By a classical result on random permutations [17]: in a uniformly random ordering of n distinct values, the i-th element is a left-to-right minimum with probability $1/i$ (being the smallest of the first i elements). By linearity of expectation, the expected number of left-to-right minima is $H_n = \sum_{i=1}^{n} \frac{1}{i} = O(\log n)$. The same bound holds for maxima. Therefore, the expected number of refinements is at most the expected number of minima in $\{x_\alpha\}$ plus the expected number of maxima in $\{y_\alpha\}$, giving $\mathbb{E}[C] \leq O(\log n) + O(\log n) = O(\log n)$. ◀

▶ **Corollary 13.** *The candidate pruning algorithm runs in expected time*

$$O(n \cdot n^2 \log n + \log n \cdot n^3 \operatorname{polylog}(n)) = \tilde{O}(n^3).$$

Proof. We process $O(n)$ simplices from both complexes. Each requires an $O(n^2 \log n)$ existence test. In expectation, only $O(\log n)$ trigger the $O(n^3 \operatorname{polylog}(n))$ band refinement. The refinement cost dominates, giving $\tilde{O}(n^3)$ expected time. ◀

4 Computing d_1 in $\mathbb{R}^2$

Our framework also improves the computation of the integral bottleneck distance d_1 from the previous $\tilde{O}(n^6)$ bound [18] to $\tilde{O}(n^5)$.

The integral requires maintaining the bottleneck matching as ω sweeps around $\mathbb{S}^1$. As ω varies, the bottleneck distance $H(\omega)$ changes only at the $O(n^4)$ difference curve intersection events identified in Section 3. At each event, we update the matching in $O(n \log n)$ time via an augmenting path search to identify which difference curve realizes the new bottleneck. Between consecutive events at angles ω_i and ω_{i+1}, the bottleneck distance is given by a single difference curve that can be integrated analytically.

Thus the integral distance can be computed in $\tilde{O}(n^5)$ total time: $O(n^4)$ events, each requiring $O(n \log n)$ processing. This improves upon the hourglass data structure approach of [18], which maintained two kinetic heaps and required $\tilde{O}(n^6)$ time.

5 Computing d_∞ in $\mathbb{R}^3$

We now consider embedded complexes in $\mathbb{R}^3$. Directions range over the sphere $\mathbb{S}^2$, difference curves become surfaces, and intersections occur along curves rather than at points. The algorithm follows the same framework as 2D, but requires traversing an arrangement on $\mathbb{S}^2$ via an Euler tour rather than sweeping a circle. This yields expected complexity $\tilde{O}(n^5)$.

5.1 Events

Let $K, K' \subset \mathbb{R}^3$ be finite simplicial complexes with at most n simplices each. For each simplex $\alpha \in K \cup K'$, $\omega \in S^2$, the *insertion surface* is:

$$I_\alpha : S^2 \to \mathbb{R}, \quad I_\alpha(\omega) := \max_{v \in \alpha} \langle v, \omega \rangle$$

Each I_α is the upper envelope of at most 4 trigonometric forms on S^2. The loci where the maximizing vertex changes are great circles $\{\omega \in S^2 : \langle v_i - v_j, \omega \rangle = 0\}$.

For distinct simplices α, σ, the *difference surface* is a real-valued function on $\mathbb{S}^2$:

$$\Delta_{\alpha,\sigma}(\omega) := \begin{cases} |I_\alpha(\omega) - I_\sigma(\omega)| & \text{if } \alpha, \sigma \text{ are from different complexes} \\ \frac{1}{2}|I_\alpha(\omega) - I_\sigma(\omega)| & \text{if } \alpha, \sigma \text{ are from the same complex} \end{cases}$$

Within each region where the active vertices are fixed, $\Delta_{\alpha,\sigma}$ simplifies to $|\langle u - w, \omega \rangle|$ for active vertices $u \in \alpha$ and $w \in \sigma$. Two difference surfaces $\Delta_{\alpha,\sigma}$ and $\Delta_{\beta,\tau}$ intersect along curves on $\mathbb{S}^2$ where $\Delta_{\alpha,\sigma}(\omega) = \Delta_{\beta,\tau}(\omega)$.

▶ **Lemma 14.** *For simplicial complexes in $\mathbb{R}^3$:*

1. *Each pair of insertion surfaces intersects along $O(1)$ great circle arcs on $\mathbb{S}^2$.*
2. *Each pair of difference surfaces intersects along $O(1)$ great circle arcs on $\mathbb{S}^2$.*
3. *Each intersection curve between difference surfaces has $O(1)$ local maxima.*
4. *Each difference surface crosses any threshold λ along $O(1)$ curves on $\mathbb{S}^2$.*

The proof follows from the piecewise structure: since simplices in $\mathbb{R}^3$ have at most 4 vertices, each insertion surface has $O(1)$ regions of fixed active vertex. Within each region, the surfaces are trigonometric forms and their intersections are great circles. The detailed geometric analysis is in the full version.

▶ **Theorem 15** (Events for d_∞ in $\mathbb{R}^3$). *The direction ω^* attaining d_∞ occurs at one of:*

1. *A local maximum of a difference surface $\Delta_{\alpha,\sigma}$*
2. *A local maximum along an intersection curve on $\mathbb{S}^2$ where two difference surfaces meet*
3. *A triple intersection point where three difference surfaces meet*

Proof. The bottleneck distance function $H(\omega)$ is continuous on S^2. At any $\omega^* \in S^2$ where d_∞ is attained, consider the difference surfaces achieving the maximum value $H(\omega^*)$.

If a unique surface $\Delta_{\alpha,\sigma}$ achieves this maximum in a neighborhood of ω^*, then by continuity, ω^* must be a local maximum of that surface (case 1). This covers both matched pairs and unmatched persistence. If exactly two surfaces achieve the maximum at ω^* but no other surface does so nearby, then ω^* lies on their intersection curve on $\mathbb{S}^2$. Since H is locally determined by these two surfaces along the curve, ω^* must be a local maximum along this curve (case 2). If three or more surfaces achieve the maximum at ω^*, then generically three surfaces meet at isolated points, giving case 3. ◀

▶ **Corollary 16.** *There are $O(n^6)$ candidate values where d_∞ may be attained.*

Proof. We count candidates by type. Type 1 has $O(n^2)$ difference surfaces (including both between-complex and within-complex pairs), each having $O(1)$ maxima, contributing $O(n^2)$ candidates. Type 2 consists of $\binom{O(n^2)}{2} = O(n^4)$ pairs with $O(1)$ maxima per intersection curve, giving $O(n^4)$ candidates. Type 3 is $\binom{O(n^2)}{3} = O(n^6)$ triple intersection points. This gives us the total of $O(n^6)$ candidates. ◀

5.2 Overview of algorithm

Decision procedure. Given threshold λ, we decide whether $d_\infty \leq \lambda$ by traversing an arrangement on $\mathbb{S}^2$. Similar to the 2D case, we track two types of events:

- Diagram changes: $O(n^2)$ intersection curves between insertion surfaces where the persistence diagrams change combinatorially;
- Threshold crossings: $O(n^2)$ curves on $\mathbb{S}^2$ where difference surfaces cross the threshold λ.

Overlaying these curves yields an arrangement of $O(n^4)$ complexity on $\mathbb{S}^2$. We traverse the dual of this arrangement via an Euler tour by traversing each edge twice, maintaining a bottleneck matching throughout. Each diagram change requires $O(n)$ time to update the barcode pairing using the vineyard algorithm, while each threshold crossing may require $O(n \log n)$ time for augmenting path search. The total complexity is $\tilde{O}(n^5)$.

Improved algorithm via band refinement. We maintain a band $(\lambda_1, \lambda_2]$ containing d_∞ and process simplices in random order. For each simplex α:

1. Existence predicate: Test if any α-candidate lies in the band. Following the approach from 2D adapted to surfaces: for fixed α and $\sigma \in K'$, consider the surface $\Delta_{\alpha,\sigma}$. For each $\beta \in K \setminus \{\alpha\}$, compute intersection curves with all $\Delta_{\beta,\tau}$ surfaces. Within the band, these $O(n)$ curves form disjoint arcs on $\mathbb{S}^2$ (no intra-β intersections after pre-refinement). Similarly get $O(n)$ disjoint arcs from surfaces involving $\gamma \in K \setminus \{\alpha, \beta\}$. Check for red-blue intersections in $O(n \log n)$ time. Testing all combinations takes $O(n^4 \log n)$.

2. Band refinement: If positive, enumerate all α-candidates in the band: $O(n^5)$ triple intersections, $O(n^3)$ curve maxima, $O(n)$ surface maxima. Then refine via binary search in $\tilde{O}(n^5)$ time.

Using the same randomization analysis as 2D, we expect $O(\log n)$ refinements. This results in total expected complexity of $O(n \cdot n^4 \log n + \log n \cdot n^5 \operatorname{polylog} n) = \tilde{O}(n^5)$.

5.3 Decision problem

Given threshold $\lambda \geq 0$, we decide whether $d_\infty \leq \lambda$ by constructing and traversing an arrangement on $\mathbb{S}^2$. Unlike the 2D case where we sweep around a circle, here we must systematically explore the entire sphere.

We first construct an arrangement $\mathcal{A}$ on $\mathbb{S}^2$ consisting of:

- $O(n^2)$ curves from insertion surface intersections (where persistence diagrams change)
- $O(n^2)$ curves where difference surfaces equal λ (threshold crossings)

Each insertion surface is piecewise trigonometric with $O(1)$ pieces, so pairs intersect along $O(1)$ curves each. Similarly, each difference surface crosses the threshold λ along $O(1)$ curves. The overlay of all these curves forms an arrangement with $O(n^4)$ vertices, edges, and faces.

▶ **Theorem 17.** *Whether $d_\infty \leq \lambda$ can be decided in $\tilde{O}(n^5)$ time.*

Proof. We construct the arrangement $\mathcal{A}$ and compute an Euler tour of its dual graph, treated as a planar graph embedded on $\mathbb{S}^2$. The tour visits each edge twice, once in each direction.

Starting from an arbitrary face with a perfect matching in $G_\lambda(\omega)$, we traverse the tour and maintain the matching throughout. At insertion curve crossings where the persistence diagram changes, we update the barcode pairing via a single transposition in $O(n)$ time using the vineyard algorithm. At threshold crossings where an edge weight exceeds λ, we search for an augmenting path in $O(n \log n)$ time. If no augmenting path exists, we conclude $d_\infty > \lambda$ and terminate. If the tour completes with valid matchings maintained throughout, then $d_B(\omega) \leq \lambda$ for all $\omega \in \mathbb{S}^2$, hence $d_\infty \leq \lambda$.

The arrangement has $O(n^4)$ edges in total. The $O(n^4)$ diagram changes contribute $O(n^5)$ total time at $O(n)$ each, while the $O(n^2)$ threshold crossings contribute $\tilde{O}(n^3)$ at $O(n \log n)$ each. However, the bottleneck arises from traversing all $O(n^4)$ edges of the arrangement, where at each edge crossing we may need to update the matching. Since each edge crossing potentially requires $O(n \log n)$ time for augmenting path search, the total complexity is $O(n^4 \cdot n \log n) = \tilde{O}(n^5)$. ◀

5.4 Candidate pruning via band refinement and complexity analysis

As in the 2D case, we maintain a band $(\lambda_1, \lambda_2]$ containing d_∞ and process simplices in random order. We first apply a pre-refinement to exclude all intra-events from the band.

Pre-refinement. Before processing individual simplices, we refine the band to exclude all intra-events from its interior. We call a triple intersection point $\Delta_{\alpha,\sigma} \cap \Delta_{\beta,\tau} \cap \Delta_{\gamma,\rho}$ an *intra-event* if any two of the six involved simplices are equal. Let S be the set containing all $O(n^2)$ local maxima of difference surfaces, all $O(n^4)$ local maxima along surface intersection curves, and all intra-event triple intersections. Since there are $O(n^5)$ ways to choose 5 distinct simplices and $O(1)$ ways to make two of them equal, we have $|S| = O(n^5)$. We use binary search with the decision procedure to find the largest value in S that is $< d_\infty$ (setting λ_1) and the smallest value in S that is $\geq d_\infty$ (setting λ_2), yielding the band $(\lambda_1, \lambda_2]$ that contains d_∞ but excludes all values of S from its interior. This takes $O(n^5 \operatorname{polylog} n)$ time.

▶ **Lemma 18.** *For fixed $\alpha \in K$, we can test whether any inter-intersection involving α lies in the band $(\lambda_1, \lambda_2]$ in $O(n^4 \log n)$ time.*

Proof. Fix $\alpha \in K$ and $\sigma \in K'$, giving the difference surface $\Delta_{\alpha,\sigma}$. We seek triple intersection points of the form $\Delta_{\alpha,\sigma} \cap \Delta_{\beta,\tau} \cap \Delta_{\gamma,\rho}$ where $\beta, \gamma \in K \setminus \{\alpha\}$ and $\tau, \rho \in K'$, with the intersection value lying in $(\lambda_1, \lambda_2]$.

For fixed $\beta \in K \setminus \{\alpha\}$, compute the intersection curves between $\Delta_{\alpha,\sigma}$ and all surfaces $\Delta_{\beta,\tau}$ for $\tau \in K'$. Each pair of surfaces intersects along great circle arcs on $\mathbb{S}^2$ where $\Delta_{\alpha,\sigma}(\omega) = \Delta_{\beta,\tau}(\omega)$. We obtain $O(n)$ intersection curves on $\mathbb{S}^2$.

The crucial observation is that when we restrict to the band $\{\omega : \lambda_1 < \Delta_{\alpha,\sigma}(\omega) \leq \lambda_2\}$, these $O(n)$ curves become disjoint arcs. Any intersection between curves $\Delta_{\alpha,\sigma} \cap \Delta_{\beta,\tau_1}$ and $\Delta_{\alpha,\sigma} \cap \Delta_{\beta,\tau_2}$ would yield a triple point where $\Delta_{\alpha,\sigma} = \Delta_{\beta,\tau_1} = \Delta_{\beta,\tau_2}$. This is an intra-event (two surfaces share β), excluded from the band's interior via pre-refinement.

Similarly, for fixed $\gamma \in K \setminus \{\alpha, \beta\}$, the intersection curves between $\Delta_{\alpha,\sigma}$ and all $\Delta_{\gamma,\rho}$ surfaces yield another set of $O(n)$ disjoint arcs within the band. We now have two collections of arcs on the restricted domain of $\Delta_{\alpha,\sigma}$: the "red" arcs from β-surfaces and the "blue" arcs from γ-surfaces. A red-blue intersection corresponds to a triple point $\Delta_{\alpha,\sigma} \cap \Delta_{\beta,\tau} \cap \Delta_{\gamma,\rho}$, which is an inter-intersection involving α.

To detect red-blue intersections, we apply stereographic projection from the north pole of $\mathbb{S}^2$ to map the band region to a planar domain and use a sweep-line algorithm. As shown in the full version, the projection preserves the disjointness of red and blue arc families while mapping great circle arcs to well-behaved curves (circles and lines) that intersect $O(1)$ times. The sweep-line algorithm thus detects all red-blue intersections in $O(n \log n)$ time.

Iterating over all choices of $\sigma \in K'$, $\beta \in K \setminus \{\alpha\}$, and $\gamma \in K \setminus \{\alpha, \beta\}$ gives total complexity $O(n \cdot n \cdot n \cdot n \log n) = O(n^4 \log n)$. ◀

▶ **Lemma 19.** *When the existence test is positive for simplex α, we can enumerate all $O(n^5)$ candidates involving α that lie in $(\lambda_1, \lambda_2]$.*

Proof. For each α, the pre-refinement step excludes the $O(n)$ local maxima of difference surfaces and $O(n^2)$ local maxima of intersection curves. We only need to consider triple intersection points $\Delta_{\alpha,\sigma} \cap \Delta_{\beta,\tau} \cap \Delta_{\gamma,\rho}$: with $O(n)$ choices each for $\sigma, \beta, \tau, \gamma, \rho$, and $O(1)$ intersection points per triple, we get $O(n^5)$ candidates. Total enumeration takes $O(n^5)$ time. Binary search with the decision procedure then refines the band in $\tilde{O}(n^5)$ time. ◀

▶ **Theorem 20** (Total complexity in 3D)**.** *The algorithm runs in expected time $\tilde{O}(n^5)$.*

Proof. We process $O(n)$ simplices from both complexes. For each simplex α, we perform the existence test in $O(n^4 \log n)$ time. If positive, we enumerate and refine in $\tilde{O}(n^5)$ time. We expect $O(\log n)$ refinements using the same analysis as in 2D. The total expected complexity is therefore $O(n \cdot n^4 \log n + \log n \cdot n^5 \operatorname{polylog} n) = \tilde{O}(n^5)$. ◀

6 Conclusion

We have presented the first algorithms for computing the maximum bottleneck distance between Persistent Homology Transforms, achieving expected time complexity of $\tilde{O}(n^3)$ in $\mathbb{R}^2$ and $\tilde{O}(n^5)$ in $\mathbb{R}^3$. As a byproduct, our framework also improves the integral bottleneck distance computation from $\tilde{O}(n^6)$ to $\tilde{O}(n^5)$ in 2D. Using similar techniques, the integral version in $\mathbb{R}^3$ can be computed in $\tilde{O}(n^7)$ time by maintaining matchings across all $O(n^6)$ events, though we leave the technical details for future work.

Our adaptation of the framework introduced by Bjerkevik and Kerber [1] reveals a broader principle: directional transforms that vary continuously over parameter spaces can often be optimized through event-driven algorithms and randomized candidate pruning. This suggests potential applications beyond the PHT to other parametric problems in computational topology, such as the Euler Characteristic Transform or persistence-based shape signatures.

From a practical perspective, the cubic complexity in 2D makes exact PHT comparison feasible for moderate-sized datasets, potentially enabling new applications in shape matching and retrieval where sampling-based approximations were previously necessary. However, significant challenges remain: extending to the Wasserstein distance would require fundamentally different techniques due to its non-combinatorial nature, and handling higher-dimensional ambient spaces faces steep complexity barriers.

References

1 Havard Bakke Bjerkevik and Michael Kerber. Asymptotic improvements on the exact matching distance for 2-parameter persistence. *Journal of Computational Geometry*, 14(1):309–342, December 2023. doi:10.20382/jocg.v14i1a12.

2 Sergio Cabello, Siu-Wing Cheng, Otfried Cheong, and Christian Knauer. Geometric Matching and Bottleneck Problems. In Wolfgang Mulzer and Jeff M. Phillips, editors, *40th International Symposium on Computational Geometry (SoCG 2024)*, volume 293 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:15, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2024.31.

3 Mathieu Carrière, Frederic Chazal, Yuichi Ike, Theo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2786–2796. PMLR, 26–28 August 2020. URL: https://proceedings.mlr.press/v108/carriere20a.html.

4 David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry*, SCG '06, pages 119–126, New York, NY, USA, 2006. Association for Computing Machinery. doi:10.1145/1137856.1137877.

5 Justin Curry, Sayan Mukherjee, and Katharine Turner. How many directions determine a shape and other sufficiency results for two topological transforms. *Transactions of the American Mathematical Society, Series B*, 2018.

6 Tamal K. Dey and Cheng Xin. Computing Bottleneck Distance for 2-D Interval Decomposable Modules. In Bettina Speckmann and Csaba D. Tóth, editors, *34th International Symposium on Computational Geometry (SoCG 2018)*, volume 99 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2018.32.

7 Tamal Krishna Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 1 edition, 2017. arXiv:1708.09359v2.

8 Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction*. American Mathematical Society, 2010.

9 Herbert Edelsbrunner and Teresa Heiss. Merge trees of periodic filtrations, 2024. doi:10.48550/arXiv.2408.16575.

10 Alon Efrat, Alon Itai, and Matthew J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, September 2001. doi:10.1007/s00453-001-0016-8.

11 Robert Ghrist, Rachel Levanger, and Huy Mai. Persistent homology and Euler integral transforms. *Journal of Applied and Computational Topology*, 2(1):55–60, October 2018. doi:10.1007/s41468-018-0017-1.

12 Chad Giusti, Eva Pastalkova, Carina Curto, and Vladimir Itskov. Clique topology reveals intrinsic geometric structure in neural correlations. *Proceedings of the National Academy of Sciences*, 112(44):13455–13460, 2015. doi:10.1073/pnas.1506407112.

13 Felix Hensel, Michael Moor, and Bastian Rieck. A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, Volume 4 - 2021, 2021. doi:10.3389/frai.2021.681108.

14 Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G. Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040, 2016. doi:10.1073/pnas.1520877113.

15 John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, December 1973. doi:10.1137/0202019.

16 Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov. Geometry helps to compare persistence diagrams. *ACM J. Exp. Algorithmics*, 22, September 2017. doi:10.1145/3064175.

17 Donald E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, 2nd edition, 1998. Section 5.1.1.

18 Elizabeth Munch, Elena Xinyi Wang, and Carola Wenk. The kinetic hourglass data structure for computing the bottleneck distance of dynamic data, 2025. doi:10.48550/arXiv.2505.04048.

19 Michael W. Reimann, Max Nolte, Martina Scolamiero, Katharine Turner, Rodrigo Perin, Giuseppe Chindemi, Paweł Dłotko, Ran Levi, Kathryn Hess, and Henry Markram. Cliques of neurons bound into cavities provide a missing link between structure and function. *Frontiers in Computational Neuroscience*, Volume 11 - 2017, 2017. doi:10.3389/fncom.2017.00048.

20 Katharine Turner, Sayan Mukherjee, and Doug M. Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, December 2014. doi:10.1093/imaiai/iau011.