A Class of Topological Pseudodistances for Fast Comparison of Persistence Diagrams

Rolando Kindelan Nuñez[1], Mircea Petrache[2], Mauricio Cerda[1*], Nancy Hitschfeld[1*]

[1]Universidad de Chile
[2]UC Chile
{rkindela, nancy}@dcc.uchile.cl, mpetrache@mat.uc.cl, mauricio.cerda@uchile.cl

Abstract

Persistence diagrams (PD)s play a central role in topological data analysis, and are used in an ever increasing variety of applications. The comparison of PD data requires computing comparison metrics among large sets of PDs, with metrics which are accurate, theoretically sound, and fast to compute. Especially for denser multi-dimensional PDs, such comparison metrics are lacking. While on the one hand, Wasserstein-type distances have high accuracy and theoretical guarantees, they incur high computational cost. On the other hand, distances between vectorizations such as Persistence Statistics (PS)s have lower computational cost, but lack the accuracy guarantees and in general they are not guaranteed to distinguish PDs (i.e. the two PS vectors of different PDs may be equal). In this work we introduce a class of pseudodistances called Extended Topological Pseudodistances (ETD)s, which have tunable complexity, and can approximate Sliced and classical Wasserstein distances at the high-complexity extreme, while being computationally lighter and close to Persistence Statistics at the lower complexity extreme, and thus allow users to interpolate between the two metrics. We build theoretical comparisons to show how to fit our new distances at an intermediate level between persistence vectorizations and Wasserstein distances. We also experimentally verify that ETDs outperform PSs in terms of accuracy and outperform Wasserstein and Sliced Wasserstein distances in terms of computational complexity.

Introduction

The processing and extraction of information from large datasets has become increasingly challenging due to the high dimensionality and noisiness of datasets. An important toolbox for describing the shape of complex data with noise robustness bounds is offered by the emerging research field of Topological Data Analysis (TDA) (Carlsson 2009; Edelsbrunner and Harer 2010; Cohen-Steiner, Edelsbrunner, and Harer 2007; Cohen-Steiner et al. 2010), which focuses on quantifying topological and geometric shape statistics of point clouds and other datasets.

An important advantage of TDA compared to other methods is the improved interpretability, based on insights from algebraic topology. The principal approach to encoding topological information in TDA are Persistence Diagrams (PDs) (Edelsbrunner and Harer 2010; Zomorodian 2009) or Persistence Barcodes (Ghrist 2008; Carlsson 2009). TDA methods are being applied in a growing variety of fields, including time-series analysis (Seversky, Davis, and Berger 2016; Carr, Garth, and Weinkauf 2017) (Venkataraman, Ramamurthy, and Turaga 2016; Umeda 2017), text data analysis (Wagner and Dłotko 2014; Rawson et al. 2022), molecular chemistry (Carr, Garth, and Weinkauf 2017), climate understanding (Carr, Garth, and Weinkauf 2017), atmospheric data analysis (Kuhn et al. 2017; Carr, Garth, and Weinkauf 2017), scientific visualization (Carr, Garth, and Weinkauf 2017), cosmology (Carr, Garth, and Weinkauf 2017), combustion simulations (Carr, Garth, and Weinkauf 2017), computational fluid dynamics (Carr, Garth, and Weinkauf 2017), neurosciences (Sizemore et al. 2019), human motion understanding (Lamar et al. 2016; Hossny et al. 2016; Venkataraman, Ramamurthy, and Turaga 2016), medical applications (Garside et al. 2019), volcanic eruption analysis (Kuhn et al. 2017; Carr, Garth, and Weinkauf 2017). In the above TDA applications, TDA has been used as a preprocessing stage for conventional Machine Learning (ML) algorithms (Ŝkraba 2018), preserving interpretability, or, more rarely, as a tool to interpret the shape of clouds manipulated via Deep Learning algorithms. The overall idea is to apply persistent homology for each sample and obtain its persistence diagram. Then the space of persistence diagrams, endowed with a suitable metric or pseudometric is used as a replacement, or as an enrichment, of the original dataset. A problem with comparison metrics between PDs is that they are computationally expensive, especially for PDs coming from H^j-homology with $j > 0$, which are a special type of point clouds in $\mathbb{R}^2$. Computing the distance between two such PDs is treated as a matching problem between points in the plane, with stability and theoretical bounds based on the link with optimal transport distances like Wasserstein Distances (WDs) (Dobrushin 1970; Cohen-Steiner et al. 2010; Panaretos and Zemel 2019; Bubenik and Elchesen 2022). Since WD computation for point clouds in dimension $d \geq 2$ have $O(n^3)$-complexity (Munkres 1957), where n is the number of points, PD comparison is often a bottleneck in ML data processing pipelines.

*These authors contributed equally.

Related Work and Main Contributions

Several approaches have arisen to make PD comparison computationally cheaper. A first direction is to optimize the precise computation of Wasserstein distance between PD's via optimization of matching problems (Dey and Wang 2022; Kerber, Morozov, and Nigmetov 2017; Chen and Wang 2021; Backurs et al. 2020), or by resorting to computationally simpler distances, such as Sliced Wasserstein Distances (SWDs) (Carrière, Cuturi, and Oudot 2017; Rabin et al. 2012; Bonneel et al. 2015; Paty and Cuturi 2019; Bayraktar and Guo 2021) which roughly speaking use as distance an average of distances of 1-dimensional projections, or to approximations of Wasserstein Distance such as Sinkhorn Distances (Cuturi 2013; Chakrabarty and Khanna 2021). Some methods use the particular geometry of PDs specifically for computing Wasserstein distances (Kerber, Morozov, and Nigmetov 2017; Khrulkov and Oseledets 2018; Dey and Zhang 2022). The work (Carrière, Cuturi, and Oudot 2017) applies SWD for PD comparison, but without optimizing the method towards optimum computational gains. A second direction to overcome PD comparison difficulties, is that of introducing simplified statistics via vectorization methods, with a variety of so-called Persistent Statistics (PS) (Adams et al. 2017; Ali et al. 2023; Chung and Lawson 2022). Then distances between PS vectorizations induce pseudodistances on the originating PDs, which while computationally faster, are not guaranteed to distinguish between distinct PD's (Fasy et al. 2020).

In view of the above challenges of Wasserstein distance and vectorization statistics, we introduce here a class of Extended Topology pseudodistances (ETDs) between PDs which are strictly richer than PS comparison, inspired from, but much faster to compute than SWD, and which also have significant computational gains with respect to previous WD-based approximate distances. Our main contributions are the following:

1. We introduce a new class of "enhanced topology pseudodistances" (ETDs) of increasing complexity (fixable by the user), which interpolate between simple PS vectorizations and the complexity of distances such as SWD and WD. Furthermore, we verify experimentally that for real data sets the loss is minimal at low complexity, and the distinguishing power of such ETDs is comparable to the one of Wasserstein distance between PD's in applications.
2. We build the basis for a rigorous theoretical comparison of our ETD distances to present methods for computing SWD and to commonly used PS vectorization. We also prove theoretical guarantees for stability under perturbations for our distance.
3. We test our ETDs for classification applications and experimentally compare to classical methods in terms of accuracy and of computation time.

It is worth emphasizing that, while a theoretical framework on metric comparison for PDs is well established (Cohen-Steiner, Edelsbrunner, and Harer 2007; Cohen-Steiner et al. 2010; Bubenik and Elchesen 2022; Ali et al. 2023; Chung and Lawson 2022), the PD construction already discards a lot of geometric and topological information about the datasets. The question of distinguishing what tasks are suited or not suited to be tackled through PD statistics is complex, and not fully settled. In the current work, we do some steps in this direction, and we hope that more research in this direction will come in the near future.

Background on PDs and Their Metrics

A Fast Reminder on Persistence Diagrams

We recall basic facts about PDs, see (Edelsbrunner and Harer 2010) for details. For a dataset encoding as a topological space X, we consider a filtration $\mathcal{F} = \{X_t\}_{t\in[0,T]}$ in which $X_0 = X$, $X_t \subseteq X_s$ for all pairs $t \leq s$ and $X_T = X$. This filtration encodes a strategy of inspection of X, where the precise construction algorithms for the X_t depend on the task at hand and are not relevant for us. As filtration parameter t increases, topological characteristics such as connected components, loops, voids, etc. appear, disappear, split or coalesce, as determined by homology classes of increasing dimension $j = 0, 1, 2, \ldots$. For each value of j the increasing set of j-th homology groups $H_j(X_t)$ associated to $\mathcal{F}$ can be encoded in the so-called *persistence module* of the filtration, which in high generality (via ad-hoc structure theorems) is decomposed in a direct sum of *persistence intervals*, each of which allows to determine the values of time parameter t at which a given homology class appears or disappears, named *birth time* b and *death time* $d \geq b$ of the corresponding feature. The set of pairs $(b, d) \subseteq \{(x, y) : x \leq y\}$ for j-dimensional homology classes form the *j-dimensional Persistence Diagram (PD)* $PD^j(X)$ of the space X. For this work we will consider a fixed dimension bound k, and we work with the Extended Persistence Diagram (EPD) $PD(X) = \{PD^0(X), PD^1(X), \ldots, PD^k(X)\}$, in which we reiterate that $PD^j(X)$ is a collection of points in $\mathbb{R}^2$ for all $0 \leq j \leq k$.

Wasserstein-Type Distances Between PDs

Here we recall the important metrics of interest for comparing PDs, namely geometric distances such as WD and SWD, and distances between vectorization summaries of PDs, such as PS. Consider a fixed dimension $j \geq 0$ and the PDs for dimension j, denoted $\mathsf{P}_1 = PD^j(X_1), \mathsf{P}_2 = PD^j(X_2) \subseteq \mathbb{R}^2$, corresponding to two datasets X_1, X_2. Standard comparison and theoretical guarantees such as stability under small perturbations between PDs is uses the Bottleneck Distance (BD) (cf. (Edelsbrunner and Harer 2010) and (Chazal et al. 2009; Chazal, de Silva, and Oudot 2014)), which is the $p \to \infty$ limit case of p-Wasserstein distances:

Definition 1 (Wasserstein distances)**.** *Let $\mathsf{P}_1, \mathsf{P}_2 \subseteq \mathbb{R}^2$ as above, set $\Delta := \{(x, x) : x \in \mathbb{R}\}$ and let Γ be the set of bijections between $\mathsf{P}_1 \cup \Delta$ and $\mathsf{P}_2 \cup \Delta$. Then for $p \in [1, \infty)$, the **p-Wasserstein distance** between $\mathsf{P}_1, \mathsf{P}_2$ is given by*

$$W_p(\mathsf{P}_1, \mathsf{P}_2) := \left[\inf_{\gamma\in\Gamma} \sum_{d\in\mathbb{P}_1\cup\Delta} \|d - \gamma(d)\|_\infty^p\right]^{\frac{1}{p}}, \quad (1)$$

*and the **Bottleneck distance** between* P_1 *and* P_2 *is given by*

$$W_\infty(\mathsf{P}_1, \mathsf{P}_2) := \inf_{\gamma \in \Gamma} \sup_{d \in \mathsf{P}_1 \cup \Delta} \|d - \gamma(d)\|_\infty. \quad (2)$$

The optimal algorithm for computing W_p for point clouds in dimension $d \geq 2$ is the Hungarian algorithm (see (Kuhn 1955) and Ch. 3 of (Peyré, Cuturi et al. 2019)) with complexity $O(N^3)$ if N is the number of points in $\mathsf{P}_1 \cup \mathsf{P}_2$. See the below discussion on time-complexity comparison for recent approximate algorithms with lower complexity. An important observation is that things improve consistently for 1-dimensional point clouds:

Proposition 1. *For two multisets* $\mathsf{P}_1, \mathsf{P}_2 \subseteq \mathbb{R}$ *the distances* $W_p(\mathsf{P}_1, \mathsf{P}_2)$ *can be computed in* $O(N \log N)$ *time.*

Proof sketch:. For distributions over $\mathbb{R}$ we have (see e.g. (Santambrogio 2015, Prop. 2.16)) $W_p(\mathsf{P}_1, \mathsf{P}_2) = \|\mathsf{sort}(\mathsf{P}_1) - \mathsf{sort}(\mathsf{P}_2)\|_p$, where $\mathsf{sort}(\mathsf{P})$ is the vector of coordinates of points from P, in non-increasing order. Assuming that the sorting operation has complexity $O(N \log N)$ and the ℓ_p-norm is computed with $O(N)$ operations, this gives the claimed complexity bound. □

Note that for 0-dimensional homology, in important cases such as for filtrations $\mathcal{F}$ coming from Čech or Vietoris-Rips complexes (Dey and Wang 2022, Ch. 6), we have birth times $b = 0$ by definition, and thus PD^0 is a 1-dimensional point cloud. In (Horak, Yu, and Salimi-Khorshidi 2021) a *topology distance (TD)* was proposed for comparing the 0-dimensional part of PD's, and it improves upon earlier statistics such as Geometry Score (Khrulkov and Oseledets 2018) for GAN comparison. The main difference between W_p and TD is that the latter is not invariant to relabelings of the points from the PD, whereas W_p is.

Unlike dimension 0, PD point clouds corresponding to homology groups of dimension $j > 1$ are "truly 2-dimensional", as birth times and death times both contain nontrivial informations about the features. As explained in the below discussion on time complexity, even considering the recent improvements on approximate Wasserstein distance computation, the cost for computing geometric distances between such PDs with good approximation, is larger than the bound from Prop. 1.

PD Vectorizations and Persistence Statistics

Vectorization is the dimension reduction of PDs from point clouds in 2 dimensions to vector data[1]. As the projection operation loses geometric information, vectorizations inherently face the tradeoff between simplicity and informativity. For a comprehensive survey of PD vectorizations see (Ali et al. 2023), in which a series of vectorizations are compared in benchmark ML tasks. We focus on the best performant statistic determined in the cited paper, which turns out to be the **Persistence Statistics (PS)**. For $PD^j(X) = \{(b_i, d_i) : i \in I_j\}$ PS includes quantile, average and variance statistics about the following collections of nonnegative numbers:

$$\{b_i\}_{i \in I_j},\ \{d_i\}_{i \in I_j},\ \{(b_i + d_i)/2\}_{i \in I_j},\ \{d_i - b_i\}_{i \in I_j}, \quad (3)$$

interpreted as, respectively, the set of birth, death, midpoints and lifetime lengths of the topological features indexed by I_j. Besides the above, PS includes the total number of (b_i, d_i) such that $d_i > b_i$, and the entropy of the multiplicity function, whose interpretation is given in (Chintakunta et al. 2015).

Extended Topology Pseudodistance

Our new Extended Topology Pseudodistances (ETD) are defined by projecting the PDs relative to each separate dimension j over a finite set of directions, and summing the 1-dimensional W_p-distances of the projections. We will apply to elements (b_i, d_i) from point clouds in $\mathbb{R}^2$ the projection onto the θ-direction defined as follows, for $\theta \in [0, 2\pi)$:

$$\pi_\theta : \mathbb{R}^2 \to \mathbb{R}, \quad \pi_\theta(x, y) := x \cos\theta + y \sin\theta.$$

Remark 1. *As before, for* $\mathsf{S} \subset \mathbb{R}^2$*, we treat* $\pi_\theta(\mathsf{S})$ *as a **multiset** and retain the multiplicity of repeated projections.*

Remark 2. *We have* $\pi_{\theta+\pi}(x) = -\pi_\theta(x)$ *thus the same information is encoded in the* π_θ*-projections restricted to just half of the available directions, e.g. restricting to* $\theta \in [0, \pi)$*.*

The point cloud obtained by orthogonal projection of a PD $\mathsf{P}^j \subset \mathbb{R}^2$ onto the diagonal is the following:

$$\widetilde{P}^j := \{((b+d)/2, (b+d)/2) :\ (b, d) \in \mathsf{P}^j\}. \quad (4)$$

Definition 2 (Extended Topology Pseudodistances)**.** *Let* $A \subset [0, \pi)$ *be a finite set of projection angles and* $p \in [0, \infty]$*, and consider two PDs* $\mathsf{P}_1 = PD(X_1), \mathsf{P}_2 = PD(X_2)$ *with* $PD(X)$ *defined as in the previous sections. For* $0 \leq j \leq k$*, define the auxiliary distances*

$$D_j^A(\mathsf{P}_1, \mathsf{P}_2) := \left(\sum_{\theta \in A} W_p\left(\pi_\theta(\mathsf{P}_1^j \cup \widetilde{P}_2^j), \pi_\theta(\mathsf{P}_2^j \cup \widetilde{P}_1^j)\right)^p\right)^{\frac{1}{p}},$$

where for finite sets $\mathsf{S}_1, \mathsf{S}_2 \subset \mathbb{R}$ *of equal cardinality, we set*

$$W_p(\mathsf{S}_1, \mathsf{S}_2) := \|\mathsf{sort}(S_1) - \mathsf{sort}(S_2)\|_p.$$

*Then the **p-Extended Topology Pseudodistance (ETD) with projection set** A is defined as:*

$$\mathsf{ETD}_A(\mathsf{P}_1, \mathsf{P}_2) := \left(\sum_{j=0}^{k} D_j^A(\mathsf{P}_1, \mathsf{P}_2)^p\right)^{1/p}.$$

We write $\mathsf{ETD} := \mathsf{ETD}_{A_1}$ *with* $A_1 = \{3\pi/4\}$ *and will call this distance the **basic p-ETD**.*

The reason why we add the sets of the form $\widetilde{P}_i^j$ in the definition of D_j^A, is for balancing: in general $\mathsf{P}_1^j, \mathsf{P}_2^j$ do not have the same cardinality, and the correct analogue of (1) requires including diagonal sets

Note that if the filtrations producing the PDs are such that birth values are equal to zero by construction for P_i^0, then these point clouds are 1-dimensional: we then replace D_0^A by $D_0(\mathsf{P}_1, \mathsf{P}_2) := \#A \cdot W_p(\pi_{\pi/2}(\mathsf{P}_1), \pi_{\pi/2}(\mathsf{P}_2))$, i.e. consider

[1]Note that the entries of a vector are an equivalent information to point clouds over the real line, each vector entry being identified with a coordinate, thus PD vectorizations are conceptually analogous to dimension reduction from 2 to 1 dimensional point clouds.

only the "death" coordinates, with no information loss (factor $\#A$ being introduced for normalization reasons). Also note that for $\theta = 3\pi/4$ we have that $\pi_\theta(\widetilde{P}_i^j) = \{0\}$.

We observe that PS-distances give a strictly less informative distance than ETD_A due to the observation contained in the following result, whose proof is a direct computation:

Lemma 1. *Let $j \geq 0$ and $\mathsf{P}^j = \{(b_1, d_1), \ldots, (b_{I_j}, d_{I_j})\}$ be the j-dimensional PD of a dataset. Then the sets from* (3) *are equal to, respectively:*

$$\pi_0(\mathsf{P}^j),\ \pi_{\pi/2}(\mathsf{P}^j),\ \tfrac{\sqrt{2}}{2}\,\pi_{\pi/4}(\mathsf{P}^j),\ \tfrac{2}{\sqrt{2}}\,\pi_{3\pi/4}(\mathsf{P}^j).$$

In particular, the above lemma implies that ETD_{A_4} distance is strictly stronger than PS for $A_4 := \{0, \pi/4, \pi/2, 3\pi/4\}$. Natural choices for $A \subset [0, \pi)$ with increasing numbers of elements are

$$A_n := \left\{ \frac{3\pi}{4} - \frac{i}{n}\pi \ (\mathsf{mod}\pi) :\ i \in \{0, \ldots, n-1\} \right\}. \quad (5)$$

In the above the "$(\mathsf{mod}\pi)$" notation means that if the number $\theta_i := 3\pi/4 - i\pi/n$ becomes negative, we replace it by $\pi - \theta_i$ instead. In Appendix B we mention a more extensive list of modifications to ETD_A which may be useful in applications. As noted in the proof of Prop. 1, we may explicitly compute W_p-distances from the above definition by sorting the corresponding vectors and taking ℓ_p-norm.

Remark 3 (invariance properties of ETD_A)**.** *In the above definition, the input of ETD_A are (unordered) collections of points encoded in $\mathsf{P}_1^j, \mathsf{P}_2^j, j = 0, \ldots, k$. In practice, we are necessarily given the point clouds in some order, and ML tasks are required to be* invariant *under reordering of the points of P_i^j, for all $(i, j) \in \{1, 2\} \times \{0, \ldots, k\}$. A wished for property of distances, adapted to ML tasks, is to actually implement this invariance, so that successive ML processing of such distances can be done without further symmetry constraints. This invariance is automatical for $\mathsf{ETD}_A(\mathsf{P}_1, \mathsf{P}_2)$ due to invariance (under relabeling of P_1^j and of P_2^j) of the intermediate quantities like $W_p(\pi_\theta(\mathsf{P}_1^j), \pi_\theta(\mathsf{P}_2^j))$.*

The following computational cost bounds for ETD are proved in Appendix A:

Theorem 1 (Computational cost of ETD_A)**.** *Let $\mathsf{P}_1, \mathsf{P}_2$ be two PDs corresponding to homology dimensions $0, \ldots, k$, and let $A \subset [0, 2\pi)$ be a set of cardinality a. Then the cost of calculating $\mathsf{ETD}_A(\mathsf{P}_1, \mathsf{P}_2)$ is*

$$a(T_1 + (k+1)M(3 + T_2 + logM)) = O(akM \log M),$$

assuming unit cost for sum or product of real numbers, and where T_1 is the cost to evaluate $\sin, \cos$, T_2 is the cost to take p-th powers, and $M := \max_{0 \leq j \leq k}(\#\mathsf{P}_1^j + \#\mathsf{P}_2^j)$.

The $M \log M$ factor in the above estimates the complexity of sorting algorithms for M real numbers. Note that implementing the sorting stage with the Trimsort algorithm allows lower complexity of $O(M)$. Trimsort is a hybrid algorithm that seamlessly blends merge sort with insertion sort. It takes advantage of the inherent structure within the data to be merged, identifying sequences of pre-sorted data and integrating them into the final list, minimizing redundant comparisons. While on the one hand ETDs can be considered as enrichments of PS-based distances (see Prop. 1), the distances ETD_A are theoretically connected to the Sliced Wasserstein Distance (SWD). The following is a reformulation of (Bonnotte 2013, Def. 5.1.1) in our setting (see also (Carrière, Cuturi, and Oudot 2017, Def. 3.1) which is specific for PD applications and (Nadjahi 2021) for more recent advances on SWD in general):

Definition 3 (Sliced Wasserstein Distance)**.** *Let $\mathsf{S}_1, \mathsf{S}_2 \subseteq \mathbb{R}^2$ be two finite point clouds, and let $\widetilde{S}_i$ be the projections as in* (4)*. Then for $p \in [1, \infty]$ the **Sliced p-Wasserstein Distance (SWD)** between them is defined as*

$$SW_p(\mathsf{S}_1, \mathsf{S}_2) := \left(\frac{1}{\pi} \int_0^\pi W_p\big(\pi_\theta(\mathsf{S}_1 \cup \widetilde{S}_2), \pi_\theta(\mathsf{S}_2 \cup \widetilde{S}_1)\big)^p \, d\theta \right)^{1/p}.$$

We see that for large n, the set of angles A_n from (5) define discretizations of $[0, \pi)$ and thus we have

$$\lim_{n \to \infty} \frac{1}{n^{1/p}} D_j^{A_n}(\mathsf{P}_1, \mathsf{P}_2) = SW_p(\mathsf{P}_1^j, \mathsf{P}_2^j). \quad (6)$$

By (Bonnotte 2013, Thm. 5.1.5), for each $p \in [1, \infty)$ there exist $c_p, C_p > 0$ such that restricted to pairs 2-dimensional point clouds $\mathsf{S}_1, \mathsf{S}_2$ included in a ball of radius $\sqrt{2}T$ (which is true for $\mathsf{S}_1 = \mathsf{P}_1^j \cup \widetilde{P}_2^j$ and $\mathsf{S}_2 = \mathsf{P}_2^j \cup \widetilde{P}_1^j$ if we truncate persistence filtrations at parameter value T as in the introduction) we get the following distance comparison with Wasserstein distance:

$$c_p SW_p \leq W_p \leq C_p T^{(p-1)/3p} \left(SW_p\right)^{1/3p}. \quad (7)$$

In particular, stability properties for PDs such as those proved in (Atienza, González-Díaz, and Soriano-Trigueros 2020) for W_p distances, extend via (7) for SW_p as well, and via (6) we get stability bounds in the large-n limit for ETD_{A_n}. See the discussion in Appendix A. More precise quantification of these bounds at both steps (W_p bounds and control for finite n in (6)) are interesting theory questions outside the scope of this paper.

Theoretical Time-Complexity Comparison

In Table 1, we present the theoretical time complexity of ETD, compared to the state-of-art computation methods including Wasserstein (WD), and Sliced-Wasserstein (SWD). The WD computes Wasserstein distance using the Scikit-tda library(Saul and Tralie 2019) which uses a variant of Hungarian Algorithm (Kuhn 1955), Python Optimal Transport (Pot WD) (Flamary et al. 2021) Wasserstein is based on (Lacombe, Cuturi, and Oudot 2018), Hera WD (Kerber, Morozov, and Nigmetov 2017), and SWD (Carrière, Cuturi, and Oudot 2017). The Pot WD, Hera WD and SWD are implemented in the Gudhi Library (Maria et al. 2014), which is one of the most popular TDA frameworks. The recent paper (Dey and Wang 2022) also compares computational cost of many recent Wasserstein approximate algorithms. PS computation requires to compute the four vectorizations (3) for each $\mathsf{P}_1^j, \mathsf{P}_2^j$ (with same notation as in Def. 2), and then to

Distance	Time Complexity
WD	$O(kM^3)$
HeraWD	$O(kM^{1.5}logM)$
SWD	$O(k\ a\ M \log M)$
PS	$O(kM)$
ETD	$O(kM \log M)$
ETD_A	$O(\#A\ k\ M \log M)$

Table 1: Time complexities, where k is the number of computed homology groups, $M = \max_j(\#\mathsf{P}_1^j + \#\mathsf{P}_2^j)$ and for SWD the quantity a is the number of slices used ($a = 50$ in the original implementation). Note that even for $a = \#A$, our ETD_A implementation is faster than the one of SWD with a slices, because performing projections to the diagonal via our eq.(4) and Def.2 is more efficient than via the method of (Carrière, Cuturi, and Oudot 2017).

take average, variance and quantiles. The vectorization calculations have a bound of $O(M)$ for each value of j, and the computation of statistical quantifiers requires further $O(M)$ operations for each j, for a total of $O(kM)$. The actual distance calculation is of lower order, and can be included in this latter bound up to increasing the implicit constant.

Experiments With PD-Based Machine Learning Tasks

While shedding light on the underlying scalability guarantees, it is important to note that the theoretical comparison in the previous section is not the final word for practical purposes. This is due to two issues:

1. Unlike for ETD, several of the most performant other methods extra data structures such as kd-trees and graphs have to be produced before the distance is computed, and these computational overheads are not explicitly discussed.
2. It is important to consider practical accuracy comparisons between metrics, especially for PS and ETD type metrics which may have lower distinguishing power than WD and SWD distances on some tasks.

We thus perform a few experiments, for comparing ETD versus state-of-art metrics based on PS, WD and SWD, to find evidence for the above two points in typical ML tasks that use PD information as an input.

We summarize in Table 3 a wall-clock comparison between the same metrics as in Table 1, in two applications to PDs coming from the ML pipelines, and we compare accuracy for the tasks in Table 2, and Figure 2 below.

Note that indeed as expected from above point 1., there are substantial differences between Table 1 and Table 3. See the next sections for precise descriptions of the considered experiments in more detail.

Experiment 1: Supervised Learning

In this experiment, we perform a common TDA+ML classification using ETD, WD-based distances, and PS, via an adaptation of the experiments from (Ali et al. 2023) to our setting.Recall that in (Ali et al. 2023) they conduct supervised learning experiments on image classification datasets: the Outex texture database (Ojala et al. 2002), the SHREC14 shape retrieval dataset (Pickup et al. 2016), and the Fashion-MNIST database (Xiao, Rasul, and Vollgraf 2017). The cited paper follows the conventional TDA+ML hybrid classification approach, where the dataset is transformed by computing PD associated to each sample, which are then vectorized via PS or other vectorizations, followed by a classification by a conventional classifier such as Support Vector Machines. In an adaptation of the above experiments to our framework, we use the k-Nearest Neighbor (kNN) classifier instead of Support Vector Machines, and we use as distances the ETD, WD, SWD and HeraWD distances besides PS-based distances (with the choice of exponent $p = 2$ in all cases). The classifier choice was motivated mainly by the necessity of provide ad-hoc topological kernels. In (Ali et al. 2023) they produce a new feature set after providing different vectorization methods, then applied directly on the conventional SVM kernels (RBF, linear). When dealing with distance matrices, defining a theoretically sound kernel is a challenging task, outside the scope of the present work. To simplify our task, we use k-Nearest Neighbor classifier which only relies on the considered distances. We compute the corresponding distances from Table 1, and we build the kernels without the need of passing through the vectorization stage for distances other than PS. We then compute the same metrics as (Ali et al. 2023) and compare the accuracy of all methods. Focussing on the simpler case, i.e. on the Outex pattern dataset, we reduced the number of classes to classify to 10, and we chose 20 samples per class. We computed a Cubical Complex on each class using the Gudhi library (Maria et al. 2014). For further details and theory of cubical complexes, please consult (Kaczynski, Mischaikow, and Mrozek 2004) as well as the following paper (Wagner, Chen, and Vuçini 2011). We compute a distance matrix using each of the distances Basic $ETD, ETDA$ with $A \in \{A_2, A_4, A_8, A_{16}\}, WD, HeraWD, PotWD, SWD, PS$. We conduct a Repeated Randomized Search (Bergstra and Bengio 2012) to determine the best k and $weight$ hyperparameters for a k-Nearest Neighbors classifier on each distance matrices. The experimental results are shown in Table 2 where the weights were omitted since using the distance weight leads to optimal accuracy. As usual, the function $h(x_q)$ which k-NN uses to assign a label to a query point x_q, simply assigns the most voted label among its k nearest neighbors (James et al. 2023):

$$h(x_q) = \arg_{y \in Y} \max \sum_{i=1}^{k} w(x_q, x_i)\mathbf{1}(c(x_i), y), \quad (8)$$

where $c(x_i)$ is the true class of x_i, $w(x_q, x_i)$ is a weight and $\mathbf{1}(c(x_i), y)$ is the indicator function that equals 1 when the x_i class is equal to y and 0 otherwise. We optimize over choices of $k \leq 9$, and optimal values of k are shown in the second column of Table 2. For w we tried two possible choices: $w(x_q, x_i) = 1$ (uniform) or $w(x_q, x_i) = \frac{1}{d(x_q, x_i)}$ (distance), and as shown in the last column in Table 2, in all

Distance	Accuracy	k	weight (u: uniform d: distance)
WD	0.98	6,9	d,d
SWD	0.96	3,6	d,d
Hera WD	0.99	3,4,9	d,d,d
ETD_{A_1}	0.89	3,4	d,d,d
ETD_{A_2}	0.83	3,4	d,d
ETD_{A_4}	0.99	3,6	d,d
ETD_{A_8}	0.99	3,4	d,d
$ETD_{A_{16}}$	0.99	3,4	d,d
PS	0.99	3,4	d,d

Table 2: kNN accuracy with some of the optimal k and w choices for each such k (see description of (8) for detailed description).

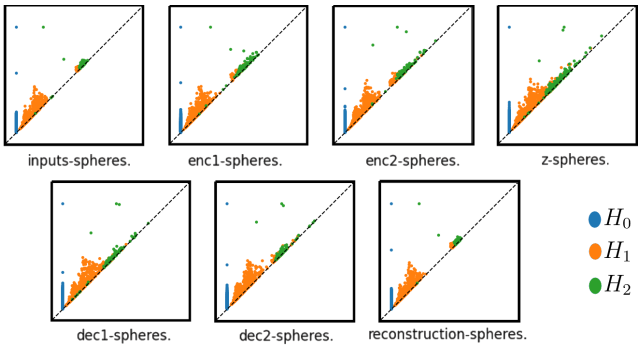

Figure 1: Example of data from Experiment 2: for each autoencoder layer, we plot the corresponding PD for H_0, H_1, H_2, in order from the input layer (left, first line) to the output/reconstruction layer (right, second line), for a total of 7 layers. We plot the distance of each persistence diagram to the first one with respect to different metrics in Fig. 2.

cases, for optimum k the optimum choice of w was the latter. We rely on Scikit-learn (Pedregosa et al. 2012) k-NN implementation. The computation of k nearest neighbors is highly sensitive to the chosen metric, a property which allows to compare metrics on this task.

We see that all methods reach high accuracy, and thus this is an example of framework in which time-effectiveness of the methods would be the relevant criterion for the choice of metric. For this experiment, the third column of Table 3 shows average time in seconds for computing the distance between two PDs in this experiment, showing that ETD_{A_1} would be the optimal choice.

Experiment 2: Autoencoder Weight Topology

According to (Naitzat, Zhitnikov, and Lim 2020), ReLU activations have a more significant impact on the topology compared to homeomorphic activations like Tanh or Leaky ReLU. ReLU activations *seem to collapse* the topology in earlier layers more rapidly.

On the one hand, autoencoders are neural networks that aim to minimize the distance between the original data and its reconstruction, creating both an 'encoder' and 'decoder' (Bengio, Goodfellow, and Courville 2017). On the other hand, the stability theorem of persistent homology (Cohen-Steiner, Edelsbrunner, and Harer 2007), implies that training an autoencoder to reconstruct data within a narrow margin $\epsilon > 0$ leads to the persistence diagrams, representing topology, that remain in close proximity within the same ϵ value. This implies that the *topology cannot be altered significantly*, even when using ReLU activations and a deep autoencoder.

We conduct an experiment to quantify and allow interpretation to the extent to which the quantification of these properties depend on the chosen PD metrics. As a toy example we consider data sampled from two concentric balls of radiuses 1 and 2 in $\mathbb{R}^{100}$ as a high-dimensional dataset with 2000 sampled points each, then train a simple autoencoder with 7 layers (of dimensions 100-20-10-3-10-20-100 respectively). After training the autoencoder, we compute persistence diagrams (corresponding to homology dimensions 0 and 1, i.e. for $j = 0$ and $j = 1$, in our notation) on the resulting point cloud given by activation vectors of each layer. Then we create so-called **topological curves** by comparing the PD of the input dataset (1st layer) with PDs corresponding to sets of output values of successive deeper layers. The comparison is done with the different metrics considered above: $WD_2, HeraWD_2, SWD_2$, PS-based metric and our new distances ETD_{A_i} for $i = 1, 2, 4, 8, 16$. The topology curves are meant to assess how much each layer changes the topology. The results depend on the chosen metric for PD comparison. Results are summarized in Figures 1 and 2. **Interpretation of the results.** Qualitative observation of topological curves as in Figure 2 across several experiments indicates that PS eliminates all variations at the level of 0th homology group H_0, and introduces large variations for successive homology groups, whereas the other more precise metrics indicate lower variations. This indicates high unreliability for PS metrics on qualitative tasks. We see agreement in the overall diagram shapes between W_2 curves, SW_2 curves, and ETD_{A_i} curves for varying values of i, which may be due to relative normalization factors between the metrics. Recall that as i increases, in theory, due to (6), we expect that ETD_{A_i} become more accurate because it approximates SW distance more closely. We observe that the ETD_{A_i} curves generally diminish their oscillations as i increases, with topological curve shapes similar to the one for Wasserstein distance (see Figure 2). Table 3 shows time in seconds for computing such curves.

We also tested the case of SWD metric with a number of slices of $a = 1, 2, 4, 8, 16$ versus the corresponding ETD_A distances with $\#A = a$, and obtained that the best computational time improvement of ETD_A versus SWD is for low values of $a = \#A$: we obtain respectively an improvement by a factor of $19.4, 1.29, 1.21, 1.1, 1.1$ for the values of $a = 1, 2, 4, 8, 16$ for trials of our autoencoder tasks from Experiment 2. This means that our implementation is more time-efficient than SWD by these factors even when applying an equal number of slices, provided this value is rela-

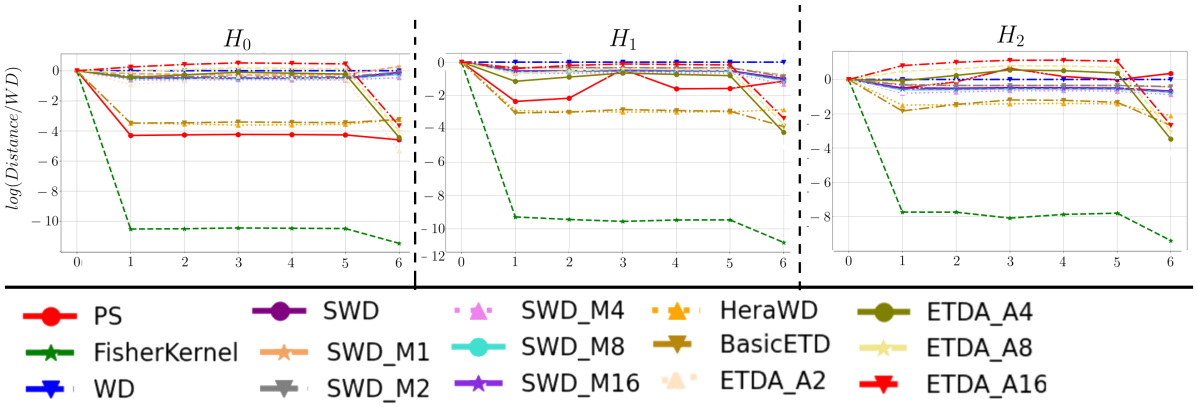

Figure 2: Example data from Experiment 2: we plot, for each homology dimension $0, 1, 2$, the values of $\log(\mathsf{dist}(P_i, P_0)/\mathsf{WD}(P_i, P_0)), 0 \leq i \leq 6$ where P_i is the PD of the i-th layer, and dist is amongst our allowed metrics. For completeness, we also include the Fisher Kernel distance comparison, which is much less discriminative than other metrics.

Distance	**Time in milliseconds**		
	Autoencoder Weight Topology		**Supervised Learning**
	RELU	**LRELU**	**Outex**
WD	12544.96	13626.25	459.24
SWD	1588.80	1551.46	404.09
Hera WD	5816.86	6574.51	864.83
ETD_{A_1}	3.69	3.77	4.19
ETD_{A_2}	88.24	72.17	8.55
ETD_{A_4}	118.87	135.30	17.11
ETD_{A_8}	236.52	271.12	34.43
$ETD_{A_{16}}$	469.74	545.9890	69.11
PS	18.61	17.53	7.51

Table 3: Average time of each distance in milliseconds spanned by activation function and by datasets on the autoencoder and supervised learning experiments respectively.

tively low.

Conclusion

We have introduced a new class of distances ETD_A on PDs for varying small parameter set A. These distances on the one hand may extend the distance between vectorizations used as the basis of Persistence Statistics, and on the other hand can in theory be enriched (at the cost of increasing A) to approximate Sliced Wasserstein Distance between PDs. In the low-$\#A$ range considered here, ETD_A pseudodistance computation has theoretical complexity bounds lower than previous distances, with no additional overhead time cost (contrary to most performant WD approximations which require to construct extra data structures with an overhead to the theoretical computational cost). The cost of ETD_A for low number of angles $\#A$ turns out to be only marginally higher than the simpler PS-based metrics, and for A_1 ("basic ETD" case) it is actually substantially lower than for PS due to optimizations specific to this case (see discussion after Def. 2). In practice, computational time for ETDs is considerably lower than state-of-art versions of WD or SWD distances. At the same time, in terms of accuracy loss, when tested on several common ML pipelines based on PDs, we see that ETD has higher performance than PS, and competitive accuracy performance compared to WD and SWD on ML tasks, since with ETD we reach similar qualitative description as with WD/SWD in Experiment 2, while PS-metrics seem to have unreliable qualitative behavior. Thus while having no strong theoretical guarantees of accuracy, the loss of accuracy of ETD_A, even for for low values of $\#A$, seems to be minimal compared to finer distances such as WD or SWD distances.

In the comparison of our new implementation to SWD with equal numbers of slices, we find that having been careful with diagonal projections of the PD's allows notable gains for very low values of these numbers of slices (especially $a = 1, 2, 4$) compared to SWD. These are values of major interest in our experiments, for which we find a qualitative gain of accuracy in our tasks.

In summary we find that ETD_A-distances allow a compromise between low computational time and low accuracy losses, allowing to interpolate between accurate WD or SWD metrics and simple PS-based metrics, by changing an in-built complexity parameter $\#A$. Furthermore, several possibly task-specific modifications presented in Appendix B can allow further adaptation of these distances to specific tasks. We expect that the theoretical control as well as experimentation with a wider variety of tasks will be a fruitful future avenue of research.

Acknowledgements

Rolando Kindelan Nuñez was supported by Beca Anid 2018/Beca doctorado Nacional-21181978, Mircea Petrache was supported by Centro Nacional de Inteligencia Artificial (CenIA) and by Fondecyt grant 1210426, Mauricio Cerda was supported by grants Fondecyt 1221696, ICN09_015, and PIA ACT192015, Nancy Hitschfeld was supported by Fondecyt grant 1211484.

References

Adams, H.; Emerson, T.; Kirby, M.; Neville, R.; Peterson, C.; Shipman, P.; Chepushtanova, S.; Hanson, E.; Motta, F.; and Ziegelmeier, L. 2017. Persistence Images: A Stable Vector Representation of Persistent Homology. *JMLR*, 18(8): 1–35.

Ali, D.; Asaad, A.; Jimenez, M.-J.; Nanda, V.; Paluzo-Hidalgo, E.; and Soriano-Trigueros, M. 2023. A Survey of Vectorization Methods in Topological Data Analysis. *IEEE Trans. Patt. Anal. Mach.e Intell.*, 1–14.

Atienza, N.; González-Díaz, R.; and Soriano-Trigueros, M. 2020. On the stability of persistent entropy and new summary functions for topological data analysis. *Pattern Recognition*, 107: 107509.

Backurs, A.; Dong, Y.; Indyk, P.; Razenshteyn, I.; and Wagner, T. 2020. Scalable nearest neighbor search for optimal transport. In *ICML*, 497–506. PMLR.

Bayraktar, E.; and Guo, G. 2021. Strong equivalence between metrics of Wasserstein type. *Electronic Communications in Probability*, 26(none): 1–13. Publisher: Institute of Mathematical Statistics and Bernoulli Society.

Bengio, Y.; Goodfellow, I.; and Courville, A. 2017. *Deep learning*, volume 1. MIT press Cambridge, MA, USA.

Bergstra, J.; and Bengio, Y. 2012. Random Search for Hyper-Parameter Optimization. *JMLR*, 13(Feb): 281–305.

Bonneel, N.; Rabin, J.; Peyré, G.; and Pfister, H. 2015. Sliced and Radon Wasserstein Barycenters of Measures. *J. Math. Imaging and Vision*, 51(1): 22–45.

Bonnotte, N. 2013. *Unidimensional and evolution methods for optimal transportation*. Ph.D. thesis, Université Paris Sud-Paris XI; Scuola normale superiore (Pise, Italie).

Bubenik, P.; and Elchesen, A. 2022. Universality of persistence diagrams and the bottleneck and Wasserstein distances. *Computational Geometry*, 105-106: 101882.

Carlsson, G. 2009. Topology and data. *Bull. Amer. Math. Soc.*, 46(2): 255–308.

Carr, H.; Garth, C.; and Weinkauf, T. 2017. *Topological Methods in Data Analysis and Visualization IV. Theory, Algorithms, and Applications*. Mathematics and Visualization. Springer International Publishing. ISBN 9783319446844.

Carrière, M.; Cuturi, M.; and Oudot, S. 2017. Sliced Wasserstein Kernel for Persistence Diagrams. In Precup, D.; and Teh, Y. W., eds., *ICML*, volume 70 of *PMLR*, 664–673. PMLR.

Chakrabarty, D.; and Khanna, S. 2021. Better and simpler error analysis of the Sinkhorn–Knopp algorithm for matrix scaling. *Mathematical Programming*, 188(1): 395–407.

Chazal, F.; Cohen-Steiner, D.; Guibas, L. J.; Mémoli, F.; and Oudot, S. Y. 2009. Gromov-Hausdorff Stable Signatures for Shapes using Persistence. *Computer Graphics Forum*, 28(5): 1393–1403.

Chazal, F.; de Silva, V.; and Oudot, S. 2014. Persistence stability for geometric complexes. *Geometriae Dedicata*, 173(1): 193–214.

Chen, S.; and Wang, Y. 2021. Approximation Algorithms for 1-Wasserstein Distance Between Persistence Diagrams. In *19th Int. Symp. Exper. Alg.*, 1.

Chintakunta, H.; Gentimis, T.; Gonzalez-Diaz, R.; Jimenez, M.-J.; and Krim, H. 2015. An entropy-based persistence barcode. *Pattern Recognition*, 48(2): 391–401.

Chung, Y.-M.; and Lawson, A. 2022. Persistence Curves: A canonical framework for summarizing persistence diagrams. *Advances in Computational Mathematics*, 48(1): 6.

Cohen-Steiner, D.; Edelsbrunner, H.; and Harer, J. 2007. Stability of Persistence Diagrams. *Discr. Comp. Geom.*, 37(1): 103–120.

Cohen-Steiner, D.; Edelsbrunner, H.; Harer, J.; and Mileyko, Y. 2010. Lipschitz Functions Have Lp-Stable Persistence. *Found. Comp. Math.*, 10(2): 127–139.

Cuturi, M. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *NIPS*, volume 26. Curran Associates, Inc.

Dey, T. K.; and Wang, Y. 2022. *Computational topology for data analysis*. Cambridge University Press.

Dey, T. K.; and Zhang, S. 2022. *Approximating 1-Wasserstein Distance between Persistence Diagrams by Graph Sparsification*, 169–183. Society for Industrial and Applied Mathematics.

Dobrushin, R. L. 1970. Prescribing a System of Random Variables by Conditional Distributions. *Theory of Probability & Its Applications*, 15(3): 458–486.

Edelsbrunner, H.; and Harer, J. 2010. *Computational Topology - an Introduction*. American Mathematical Society. ISBN 978-0-8218-4925-5.

Fasy, B.; Qin, Y.; Summa, B.; and Wenk, C. 2020. Comparing Distance Metrics on Vectorized Persistence Summaries. In *TDA & Beyond*.

Flamary, R.; Courty, N.; Gramfort, A.; Alaya, M. Z.; Boisbunon, A.; Chambon, S.; Chapel, L.; Corenflos, A.; Fatras, K.; Fournier, N.; Gautheron, L.; Gayraud, N. T.; Janati, H.; Rakotomamonjy, A.; Redko, I.; Rolet, A.; Schutz, A.; Seguy, V.; Sutherland, D. J.; Tavenard, R.; Tong, A.; and Vayer, T. 2021. POT: Python Optimal Transport. *JMLR*, 22(78): 1–8.

Garside, K.; Henderson, R.; Makarenko, I.; and Masoller, C. 2019. Topological data analysis of high resolution diabetic retinopathy images. *PLOS ONE*, 14(5): 1–10.

Ghrist, R. 2008. Barcodes: The persistent topology of data. *BULLETIN (New Series) OF THE AMERICAN MATHEMATICAL SOCIETY*, 45: 61–75.

Horak, D.; Yu, S.; and Salimi-Khorshidi, G. 2021. Topology Distance: A Topology-Based Approach for Evaluating Generative Adversarial Networks. *AAAI*, 35(9): 7721–7728.

Hossny, M.; Mohammed, S.; Nahavandi, S.; Nelson, K.; and Hossny, M. 2016. Driving behaviour analysis using topological features. In *SMC2016*. IEEE.

James, G.; Witten, D.; Hastie, T.; Tibshirani, R.; and Taylor, J. 2023. *An Introduction to Statistical Learning: with Applications in Python*. Springer Texts in Statistics. Springer. ISBN 9783031387470.

Kaczynski, T.; Mischaikow, K.; and Mrozek, M. 2004. *Computational Homology*. Applied Mathematical Sciences. Springer New York. ISBN 9780387408538.

Kerber, M.; Morozov, D.; and Nigmetov, A. 2017. Geometry Helps to Compare Persistence Diagrams. *ACM J. Exp. Algorithmics*, 22.

Khrulkov, V.; and Oseledets, I. 2018. Geometry Score: A Method For Comparing Generative Adversarial Networks. In *ICML*.

Kuhn, A.; Engelke, W.; Flatken, M.; Hege, H.; and Hotz, I. 2017. Topology-Based Analysis for Multimodal Atmospheric Data of Volcano Eruptions. In et al., C., ed., *Topological Methods in Data Analysis and Visualization IV*, 35–50. Springer. ISBN 978-3-319-44684-4.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Nav. Res. Log. Quart.*, 2(1-2): 83–97.

Lacombe, T.; Cuturi, M.; and Oudot, S. 2018. Large Scale Computation of Means and Clusters for Persistence Diagrams Using Optimal Transport. In *Proc. 32nd Int. Conference on Neural Information Processing Systems*, NIPS'18, 9792–9802. Curran Associates Inc.

Lamar, J.; Alonso, R.; Garcia, E.; and Gonzalez-Díaz, R. 2016. Persistent homology-based gait recognition robust to upper body variations. In *ICPR2016, Cancún, Mexico, December 4-8, 2016*.

Maria, C.; Boissonnat, J.; Glisse, M.; and Yvinec, M. 2014. The Gudhi Library: Simplicial Complexes and Persistent Homology. In Hong, H.; and Yap, C., eds., *Mathematical Software – ICMS 2014*. Springer Berlin Heidelberg.

Munkres, J. 1957. Algorithms for the Assignment and Transportation Problems. *Journal of the SIAM*, 5(1): 32–38. Full publication date: Mar., 1957.

Nadjahi, K. 2021. *Sliced-Wasserstein distance for large-scale machine learning : theory, methodology and extensions*. Thesis, Institut Polytechnique de Paris.

Naitzat, G.; Zhitnikov, A.; and Lim, L.-H. 2020. Topology of Deep Neural Networks. *JMLR*, 21(184): 1–40.

Ojala, T.; Maenpaa, T.; Pietikainen, M.; Viertola, J.; Kyllonen, J.; and Huovinen, S. 2002. Outex - new framework for empirical evaluation of texture analysis algorithms. In *2002 ICPR*, volume 1, 701–706 vol.1.

Panaretos, V. M.; and Zemel, Y. 2019. Statistical Aspects of Wasserstein Distances. *Annual Review of Statistics and Its Application*, 6(1): 405–431.

Paty, F.-P.; and Cuturi, M. 2019. Subspace Robust Wasserstein Distances. In *ICML*, 5072–5081. PMLR. ISSN: 2640-3498.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E.; and Louppe, G. 2012. Scikit-learn: Machine Learning in Python. *JMLR*, 12.

Peyré, G.; Cuturi, M.; et al. 2019. Computational optimal transport: With applications to data science. *FTML*, 11(5-6): 355–607.

Pickup, D.; Sun, X.; Rosin, P. L.; Martin, R. R.; Cheng, Z.; Lian, Z.; Aono, M.; Hamza, A. B.; Bronstein, A.; Bronstein, M.; Bu, S.; Castellani, U.; Cheng, S.; Garro, V.; Giachetti, A.; Godil, A.; Isaia, L.; Han, J.; Johan, H.; Lai, L.; Li, B.; Li, C.; Li, H.; Litman, R.; Liu, X.; Liu, Z.; Lu, Y.; Sun, L.; Tam, G.; Tatsuma, A.; and Ye, J. 2016. Shape Retrieval of Non-rigid 3D Human Models. *IJCV*, 120(2): 169–193.

Rabin, J.; Peyré, G.; Delon, J.; and Bernot, M. 2012. Wasserstein Barycenter and Its Application to Texture Mixing. In Bruckstein, A. M.; ter Haar Romeny, B. M.; Bronstein, A. M.; and Bronstein, M. M., eds., *Scale Space and Variational Methods in Computer Vision*, 435–446. Springer Berlin Heidelberg. ISBN 978-3-642-24785-9.

Rawson, M.; Dooley, S.; Bharadwaj, M.; and Choudhary, R. 2022. Topological Data Analysis for Word Sense Disambiguation.

Santambrogio, F. 2015. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63): 94.

Saul, N.; and Tralie, C. 2019. Scikit-TDA: Topological Data Analysis for Python.

Seversky, L. M.; Davis, S.; and Berger, M. 2016. On Time-Series Topological Data Analysis: New Data and Opportunities. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1014–1022.

Sizemore, A. E.; Phillips-Cremins, J. E.; Ghrist, R.; and Bassett, D. S. 2019. The importance of the whole: Topological data analysis for the network neuroscientist. *Network Neuroscience*, 3(3): 656–673.

Ŝkraba, P. 2018. Persistent Homology and Machine Learning. *Informatica*, 42(2): 253–258.

Umeda, Y. 2017. Time Series Classification via Topological Data Analysis. *Trans. Jap. Soc. AI*, 32: D–G72.

Venkataraman, V.; Ramamurthy, K. N.; and Turaga, P. K. 2016. Persistent homology of attractors for action recognition. In *ICIP 2016*, 4150–4154.

Wagner, H.; Chen, C.; and Vuçini, E. 2011. Efficient computation of persistent homology for cubical data. In *Topological methods in data analysis and visualization II: theory, algorithms, and applications*, 91–106. Springer.

Wagner, H.; and Dłotko, P. 2014. Towards topological analysis of high-dimensional feature spaces. *Comp. Vis. Im. Underst.*, 121: 21–26.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR*, abs/1708.07747.

Zomorodian, A. 2009. *Topology for Computing*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press. ISBN 0521136091.