Sliced Wasserstein Kernel for Persistence Diagrams

Mathieu Carriere, Marco Cuturi, Steve Y. Oudot

Sliced Wasserstein Kernel for Persistence Diagrams

Mathieu Carrière [1] **Marco Cuturi** [2] **Steve Oudot** [1]

Abstract

Persistence diagrams (PDs) play a key role in topological data analysis (TDA), in which they are routinely used to describe topological properties of complicated shapes. PDs enjoy strong stability properties and have proven their utility in various learning contexts. They do not, however, live in a space naturally endowed with a Hilbert structure and are usually compared with non-Hilbertian distances, such as the bottleneck distance. To incorporate PDs in a convex learning pipeline, several kernels have been proposed with a strong emphasis on the stability of the resulting RKHS distance w.r.t. perturbations of the PDs. In this article, we use the Sliced Wasserstein approximation of the Wasserstein distance to define a new kernel for PDs, which is not only provably stable but also discriminative (with a bound depending on the number of points in the PDs) w.r.t. the first diagram distance between PDs. We also demonstrate its practicality, by developing an approximation technique to reduce kernel computation time, and show that our proposal compares favorably to existing kernels for PDs on several benchmarks.

1. Introduction

Topological Data Analysis (TDA) is an emerging trend in data science, grounded on topological methods to design descriptors for complex data—see e.g. (Carlsson, 2009) for an introduction to the subject. The descriptors of TDA can be used in various contexts, in particular statistical learning and geometric inference, where they provide useful insight into the structure of data. Applications of TDA can be found in a number of scientific areas, including computer vision (Li et al., 2014), materials science (Hiraoka et al., 2016), and brain science (Singh et al., 2008), to name a few. The tools developed in TDA are built upon persistent homology theory (Edelsbrunner & Harer, 2010; Oudot, 2015), and their main output is a descriptor called *persistence diagram* (PD), which encodes the topology of a space at all scales in the form of a point cloud with multiplicities in the plane $\mathbb{R}^2$—see Section 2.1 for more details.

PDs as features. The main strength of PDs is their stability with respect to perturbations of the data (Chazal et al., 2009b; 2013). On the downside, their use in learning tasks is not straightforward. Indeed, a large class of learning methods, such as SVM or PCA, requires a Hilbert structure on the descriptors space, which is not the case for the space of PDs. Actually, many simple operators of $\mathbb{R}^n$, such as addition, average or scalar product, have no analogues in that space. Mapping PDs to vectors in $\mathbb{R}^n$ or in some infinite-dimensional Hilbert space is one possible approach to facilitate their use in discriminative settings.

Related work. A series of recent contributions have proposed kernels for PDs, falling into two classes. The first class of methods builds explicit feature maps: One can, for instance, compute and sample functions extracted from PDs (Bubenik, 2015; Adams et al., 2017; Robins & Turner, 2016); sort the entries of the distance matrices of the PDs (Carrière et al., 2015); treat the PD points as roots of a complex polynomial, whose coefficients are concatenated (Fabio & Ferri, 2015). The second class of methods, which is more relevant to our work, defines implicitly feature maps by focusing instead on building kernels for PDs. For instance, Reininghaus et al. (2015) use solutions of the heat differential equation in the plane and compare them with the usual $L^2(\mathbb{R}^2)$ dot product. Kusano et al. (2016) handle a PD as a discrete measure on the plane, and follow by using kernel mean embeddings with Gaussian kernels—see Section 4 for precise definitions. Both kernels are provably *stable*, in the sense that the metric they induce in their respective reproducing kernel Hilbert space (RKHS) is bounded above by the distance between PDs. Although these kernels are injective, there is no evidence that their induced RKHS distances are discriminative and therefore follow the geometry of the bottleneck distances, which are more widely accepted distances to compare PDs.

Contributions. In this article, we use the sliced Wasserstein (SW) distance (Rabin et al., 2011) to define a new ker-

[1]INRIA Saclay [2]CREST, ENSAE, Université Paris Saclay. Correspondence to: Mathieu Carrière <mathieu.carriere@inria.fr>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

nel for PDs, which we prove to be both stable and discriminative. Specifically, we provide distortion bounds on the SW distance that quantify its ability to mimic bottleneck distances between PDs. This is in contrast to other kernels for PDs, which only focus on stability. We also propose a simple approximation algorithm to speed up the computation of that kernel, confirm experimentally its discriminative power and show that it outperforms experimentally both proposals of (Kusano et al., 2016) and (Reininghaus et al., 2015) in several supervised classification problems.

2. Background on TDA and Kernels

We briefly review in this section relevant material on TDA, notably persistence diagrams, and technical properties of positive and negative definite kernel functions.

2.1. Persistent Homology

Persistent homology (Zomorodian & Carlsson, 2005; Edelsbrunner & Harer, 2008; Oudot, 2015) is a technique inherited from algebraic topology for computing stable signatures on real-valued functions. Given $f : X \to \mathbb{R}$ as input, persistent homology outputs a planar point set with multiplicities, called the *persistence diagram* of f and denoted by $\mathrm{Dg}\, f$. See Figure 1 for an example. To understand the meaning of each point in this diagram, it suffices to know that, to compute $\mathrm{Dg}\, f$, persistent homology considers the family of *sublevel sets* of f, i.e. the sets of the form $f^{-1}((-\infty, t])$ for $t \in \mathbb{R}$, and it records the *topological events* (e.g. creation or merge of a connected component, creation or filling of a loop, void, etc.) that occur in $f^{-1}((-\infty, t])$ as t ranges from $-\infty$ to $+\infty$. Then, each point $p \in \mathrm{Dg}\, f$ represents the lifespan of a particular *topological feature* (connected component, loop, void, etc.), with its creation and destruction times as coordinates. See again Figure 1 for an illustration.

For the interested reader, we point out that the mathematical tool used by persistent homology to track the topological events in the family of sublevel sets is *homological algebra*, which turns the parametrized family of sublevel sets into a parametrized family of vector spaces and linear maps. Computing persistent homology then boils down to computing a family of bases for the vector spaces, which are compatible with the linear maps.

Distance between PDs. We now define the *pth diagram distance* between PDs. Let $p \in \mathbb{N}$ and $\mathrm{Dg}_1, \mathrm{Dg}_2$ be two PDs. Let $\Gamma : \mathrm{Dg}_1 \supseteq A \to B \subseteq \mathrm{Dg}_2$ be a *partial bijection* between Dg_1 and Dg_2. Then, for any point $x \in A$, the *cost* of x is defined as $c(x) := \|x - \Gamma(x)\|_\infty^p$, and for any point $y \in (\mathrm{Dg}_1 \sqcup \mathrm{Dg}_2) \setminus (A \sqcup B)$, the *cost* of y is defined as $c'(y) := \|y - \pi_\Delta(y)\|_\infty^p$, where π_Δ is the projection onto the diagonal $\Delta = \{(x, x) \mid x \in \mathbb{R}\}$. The cost $\mathrm{c}(\Gamma)$ is defined as: $\mathrm{c}(\Gamma) := (\sum_x c(x) + \sum_y c'(y))^{1/p}$. We then define the *pth diagram distance* d_p as the cost of the best partial bijection between the PDs:

$$d_p(\mathrm{Dg}_1, \mathrm{Dg}_2) = \inf_\Gamma \mathrm{c}(\Gamma).$$

In the particular case $p = +\infty$, the cost of Γ is defined as $\mathrm{c}(\Gamma) := \max\{\max_x \delta(x) + \max_y \delta'(y)\}$. The corresponding distance d_∞ is often called the *bottleneck distance*. One can show that $d_p \to d_\infty$ when $p \to +\infty$. A fundamental property of PDs is their stability with respect to (small) perturbations of their originating functions. Indeed, the *stability theorem* (Bauer & Lesnick, 2015; Chazal et al., 2009a; 2016; Cohen-Steiner et al., 2007) asserts that for any $f, g : X \to \mathbb{R}$, we have

$$d_\infty(\mathrm{Dg}\, f, \mathrm{Dg}\, g) \leq \|f - g\|_\infty, \tag{1}$$

See again Figure 1 for an illustration.

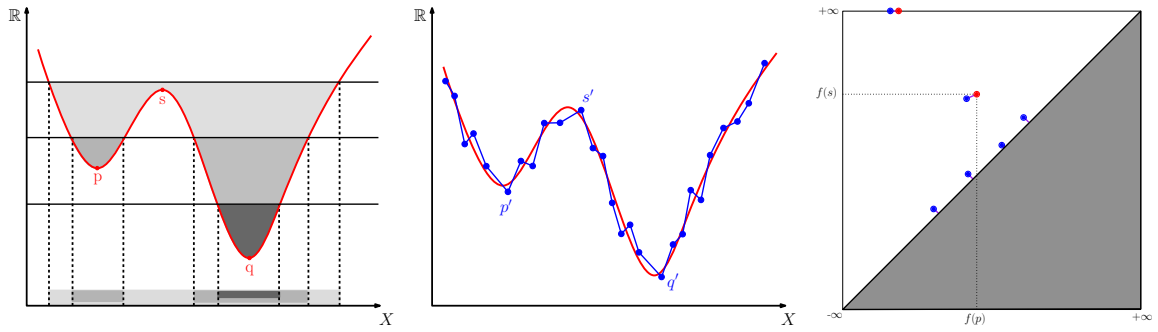
In practice, PDs can be used as descriptors for data via the choice of appropriate filtering functions f, e.g. distance to the data in the ambient space, eccentricity, curvature, etc. The main strengths of the obtained descriptors are: (a) to be provably stable as mentioned previously; (b) to be invariant under reparametrization of the data; and (c) to encode information about the topology of the data, which is complementary and of an essentially different nature compared to geometric or statistical quantities. These properties have made persistence diagrams useful in a variety of contexts, including the ones mentioned in the introduction of the paper. For further details on persistent homology and on applications of PDs, the interested reader can refer e.g. to (Oudot, 2015) and the references therein.

2.2. Kernel Methods

Positive Definite Kernels. Given a set X, a function $k : X \times X \to \mathbb{R}$ is called a *positive definite kernel* if for all integers n, for all families $x_1, ..., x_n$ of points in X, the matrix $[k(x_i, x_j)]_{i,j}$ is itself positive semi-definite. For brevity we will refer to positive definite kernels as kernels in the rest of the paper. It is known that kernels generalize scalar products, in the sense that, given a kernel k, there exists a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k$ and a *feature map* $\phi : X \to \mathcal{H}_k$ such that $k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle_{\mathcal{H}_k}$. A kernel k also induces a distance d_k on X that can be computed as the Hilbert norm of the difference between two embeddings:

$$d_k^2(x_1, x_2) \overset{\text{def.}}{=} k(x_1, x_1) + k(x_2, x_2) - 2\, k(x_1, x_2).$$

We will be particularly interested in this distance, since one of the goals we will aim for will be that of designing a kernel k for persistence diagrams such that d_k has low distortion with respect to d_1.



(a) (b) (c)

Figure 1. Sketch of persistent homology: (a) the horizontal lines are the boundaries of sublevel sets $f((-\infty, t])$, which are colored in decreasing shades of grey. The vertical dotted lines are the boundaries of their different connected components. For instance, a new connected component is created in the sublevel set $f^{-1}((-\infty, t])$ when $t = f(p)$, and it is merged (destroyed) when $t = f(s)$; its lifespan is represented by a copy of the point with coordinates $(f(p), f(s))$ in the persistence diagram of f (Figure (c)); (b) a piecewise-linear approximation g (blue) of the function f (red) from sampled values; (c) superposition of $\mathrm{Dg}\, f$ (red) and $\mathrm{Dg}\, g$ (blue), showing the partial matching of minimum cost (magenta) between the two persistence diagrams.

Negative Definite and RBF Kernels. A standard way to construct a kernel is to exponentiate the negative of a Euclidean distance. Indeed, the Gaussian kernel for vectors with parameter $\sigma > 0$ does follow that template approach: $k_\sigma(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$. An important theorem of Berg et al. (1984) (Theorem 3.2.2, p.74) states that such an approach to build kernels, namely setting

$$k_\sigma(x, y) \overset{\text{def.}}{=} \exp\left(-\frac{f(x, y)}{2\sigma^2}\right),$$

for an arbitrary function f can only yield a valid positive definite kernel for all $\sigma > 0$ if and only if f is a *negative semi-definite* function, namely that, for all integers n, $\forall x_1, ..., x_n \in X$, $\forall a_1, ..., a_n \in \mathbb{R}^n$ such that $\sum_i a_i = 0$, $\sum_{i,j} a_i a_j f(x_i, x_j) \leq 0$.

Unfortunately, as observed in Appendix A of Reininghaus et al. (2014), d_1 is not negative semi-definite (it only suffices to sample a family of point clouds to observe experimentally that more often than not the inequality above will be violated for a particular weight vector a). In this article, we use an approximation of d_1 with the *Sliced Wasserstein distance*, which is provably negative semi-definite, and we use it to define a RBF kernel that can be easily tuned thanks to its bandwidth parameter σ.

2.3. Wasserstein distance for unnormalized measures on $\mathbb{R}$

The Wasserstein distance (Villani, 2009, §6) is a distance between probability measures. For reasons that will become clear in the next section, we will focus on a variant of that distance: the 1-Wasserstein distance for *nonnegative*, not necessarily normalized, measures on the real line (Santambrogio, 2015, §2). Let μ and ν be two nonnegative measures on the real line such that $|\mu| = \mu(\mathbb{R})$ and $|\nu| = \nu(\mathbb{R})$ are equal to the same number r. We define the three following objects:

$$\mathcal{W}(\mu, \nu) = \inf_{P \in \Pi(\mu,\nu)} \iint_{\mathbb{R}\times\mathbb{R}} |x - y| P(\mathrm{d}x, \mathrm{d}y) \tag{2}$$

$$\mathcal{Q}_r(\mu, \nu) = r \int_{\mathbb{R}} |M^{-1}(x) - N^{-1}(x)| \mathrm{d}x \tag{3}$$

$$\mathcal{L}(\mu, \nu) = \inf_{f \in 1-\text{Lipschitz}} \int_{\mathbb{R}} f(x)[\mu(\mathrm{d}x) - \nu(\mathrm{d}x)] \tag{4}$$

where $\Pi(\mu, \nu)$ is the set of measures on $\mathbb{R}^2$ with marginals μ and ν, and M^{-1} and N^{-1} the generalized quantile functions of the probability measures μ/r and ν/r respectively.

Proposition 2.1. *We have $\mathcal{W} = \mathcal{Q}_r = \mathcal{L}$. Additionally* (i) *$\mathcal{Q}_r$ is negative definite on the space of measures of mass r;* (ii) *for any three positive measures μ, ν, γ such that $|\mu| = |\nu|$, we have $\mathcal{L}(\mu + \gamma, \nu + \gamma) = \mathcal{L}(\mu, \nu)$.*

Equation (2) is the generic Kantorovich formulation of optimal transport, which is easily generalized to other cost functions and spaces, the variant being that we consider an unnormalized mass by reflecting it directly in the set Π. The equality between (2) and (3) is only valid for probability measures on the real line. Because the cost function $|\cdot|$ is homogeneous, we see that the scaling factor r can be removed when considering the quantile function and multiplied back. The equality between (2) and (4) is due to the well known Kantorovich duality for a distance cost (Villani, 2009, Particular case 5.4) which can also be trivially generalized to unnormalized measures, proving therefore the main statement of the proposition. The definition of Q_r shows that the Wasserstein distance is the l_1 norm of

$rM^{-1} - rN^{-1}$, and is therefore a negative definite kernel (as the l_1 distance between two direct representations of μ and ν as functions rM^{-1} and rN^{-1}), proving point (i). The second statement is immediate.

We conclude with an important practical remark: for two unnormalized uniform empirical measures $\mu = \sum_{i=1}^n \delta_{x_i}$ and $\nu = \sum_{i=1}^n \delta_{y_i}$ of the same size, with ordered $x_1 \leq \cdots \leq x_n$ and $y_1 \leq \cdots \leq y_n$, one has: $\mathcal{L}(\mu,\nu) = \sum_{i=1}^n |x_i - y_i| = \|X - Y\|_1$, where $X = (x_1, ..., x_n) \in \mathbb{R}^n$ and $Y = (y_1, ..., y_n) \in \mathbb{R}^n$.

3. The Sliced Wasserstein Kernel

In this section we define a new kernel between PDs, called the *Sliced Wasserstein* (SW) kernel, based on the Sliced Wasserstein metric of Rabin et al. (2011). The idea underlying this metric is to slice the plane with lines passing through the origin, to project the measures onto these lines where $\mathcal{W}$ is computed, and to integrate those distances over all possible lines. Formally:

Definition 3.1. *Given $\theta \in \mathbb{R}^2$ with $\|\theta\|_2 = 1$, let $L(\theta)$ denote the line $\{\lambda\,\theta \mid \lambda \in \mathbb{R}\}$, and let $\pi_\theta : \mathbb{R}^2 \to L(\theta)$ be the orthogonal projection onto $L(\theta)$. Let $\mathrm{Dg}_1, \mathrm{Dg}_2$ be two PDs, and let $\mu_1^\theta := \sum_{p \in \mathrm{Dg}_1} \delta_{\pi_\theta(p)}$ and $\mu_{1\Delta}^\theta := \sum_{p \in \mathrm{Dg}_1} \delta_{\pi_\theta \circ \pi_\Delta(p)}$, and similarly for μ_2^θ, where π_Δ is the orthogonal projection onto the diagonal. Then, the* Sliced Wasserstein distance *is defined as:*

$$\mathrm{SW}(\mathrm{Dg}_1, \mathrm{Dg}_2) \overset{\text{def.}}{=} \frac{1}{2\pi} \int_{\mathbb{S}_1} \mathcal{W}(\mu_1^\theta + \mu_{2\Delta}^\theta, \mu_2^\theta + \mu_{1\Delta}^\theta) \mathrm{d}\theta.$$

Note that, by symmetry, one can restrict on the half-circle $[-\frac{\pi}{2}, \frac{\pi}{2}]$ and normalize by π instead of 2π. Since $\mathcal{Q}_r$ is negative semi-definite, we can deduce that SW itself is negative semi-definite:

Lemma 3.2. *Let X be the set of bounded and finite PDs. Then,* SW *is negative semi-definite on X.*

Proof. Let $n \in \mathbb{N}^*$, $a_1, ..., a_n \in \mathbb{R}$ such that $\sum_i a_i = 0$ and $\mathrm{Dg}_1, ..., \mathrm{Dg}_n \in X$. Given $1 \leq i \leq n$, we let $\tilde{\mu}_i^\theta := \mu_i^\theta + \sum_{q \in \mathrm{Dg}_k, k \neq i} \delta_{\pi_\theta \circ \pi_\Delta(q)}$, $\tilde{\mu}_{ij\Delta}^\theta := \sum_{p \in \mathrm{Dg}_k, k \neq i,j} \delta_{\pi_\theta \circ \pi_\Delta(p)}$ and $d = \sum_i |\mathrm{Dg}_i|$. Then:

$$\begin{aligned}
&\sum_{i,j} a_i a_j \mathcal{W}(\mu_i^\theta + \mu_{j\Delta}^\theta, \mu_j^\theta + \mu_{i\Delta}^\theta) \\
&= \sum_{i,j} a_i a_j \mathcal{L}(\mu_i^\theta + \mu_{j\Delta}^\theta, \mu_j^\theta + \mu_{i\Delta}^\theta) \\
&= \sum_{i,j} a_i a_j \mathcal{L}(\mu_i^\theta + \mu_{j\Delta}^\theta + \mu_{ij\Delta}^\theta, \mu_j^\theta + \mu_{i\Delta}^\theta + \mu_{ij\Delta}^\theta) \\
&= \sum_{i,j} a_i a_j \mathcal{L}(\tilde{\mu}_i^\theta, \tilde{\mu}_j^\theta) = \sum_{i,j} a_i a_j \mathcal{Q}_d(\tilde{\mu}_i^\theta, \tilde{\mu}_j^\theta) \leq 0
\end{aligned}$$

The result follows by linearity of integration. □

Hence, the theorem of Berg et al. (1984) allows us to define a valid kernel with:

$$k_{\mathrm{SW}}(\mathrm{Dg}_1, \mathrm{Dg}_2) \overset{\text{def.}}{=} \exp\left(-\frac{\mathrm{SW}(\mathrm{Dg}_1, \mathrm{Dg}_2)}{2\sigma^2}\right). \quad (5)$$

Metric equivalence. We now give the main theoretical result of this article, which states that SW is *equivalent* to d_1. This has to be compared with (Reininghaus et al., 2015) and (Kusano et al., 2016), which only prove stability and injectivity. Our equivalence result states that the k_{SW}, in addition to be stable and injective, also preserves the metric between PDs, which should intuitively lead to an improvement of the classification power. This intuition is illustrated in Section 4 and Figure 4, where we show an improvement of classification accuracies on several benchmark applications.

Theorem 3.3. *Let X be the set of bounded PDs with cardinalities bounded by $N \in \mathbb{N}^*$. Let $\mathrm{Dg}_1, \mathrm{Dg}_2 \in X$. Then, one has:*

$$\frac{d_1(\mathrm{Dg}_1, \mathrm{Dg}_2)}{2M} \leq \mathrm{SW}(\mathrm{Dg}_1, \mathrm{Dg}_2) \leq 2\sqrt{2} d_1(\mathrm{Dg}_1, \mathrm{Dg}_2),$$

where $M = 1 + 2N(2N-1)$.

Proof. Let $s^\theta : \mathrm{Dg}_1 \cup \pi_\Delta(\mathrm{Dg}_2) \to \mathrm{Dg}_2 \cup \pi_\Delta(\mathrm{Dg}_1)$ be the one-to-one bijection between $\mathrm{Dg}_1 \cup \pi_\Delta(\mathrm{Dg}_2)$ and $\mathrm{Dg}_2 \cup \pi_\Delta(\mathrm{Dg}_1)$ induced by $\mathcal{W}(\mu_1^\theta + \mu_{2\Delta}^\theta, \mu_2^\theta + \mu_{1\Delta}^\theta)$, and let s be the one-to-one bijection between $\mathrm{Dg}_1 \cup \pi_\Delta(\mathrm{Dg}_2)$ and $\mathrm{Dg}_2 \cup \pi_\Delta(\mathrm{Dg}_1)$ induced by the partial bijection achieving $d_1(\mathrm{Dg}_1, \mathrm{Dg}_2)$.

Upper bound. Recall that $\|\theta\|_2 = 1$. We have:

$$\begin{aligned}
&\mathcal{W}(\mu_1^\theta + \mu_{2\Delta}^\theta, \mu_2^\theta + \mu_{1\Delta}^\theta) = \sum |\langle p - s^\theta(p), \theta\rangle| \\
&\leq \sum |\langle p - s(p), \theta\rangle| \leq \sqrt{2} \sum \|p - s(p)\|_\infty \\
&\leq 2\sqrt{2} d_1(\mathrm{Dg}_1, \mathrm{Dg}_2),
\end{aligned}$$

where the sum is taken over all $p \in \mathrm{Dg}_1 \cup \pi_\Delta(\mathrm{Dg}_2)$. The upper bound follows by linearity.

Lower bound. The idea is to use the fact that s^θ is a piecewise-constant function of θ, and that it has at most $2 + 2N(2N-1)$ critical values $\Theta_0, ..., \Theta_M$ in $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Indeed, it suffices to look at all θ such that $\langle p_1 - p_2, \theta\rangle = 0$ for some p_1, p_2 in $\mathrm{Dg}_1 \cup \pi_\Delta(\mathrm{Dg}_2)$ or $\mathrm{Dg}_2 \cup \pi_\Delta(\mathrm{Dg}_1)$. Then:

$$\begin{aligned}
&\int_{\Theta_i}^{\Theta_{i+1}} \sum |\langle p - s^\theta(p), \theta\rangle| \mathrm{d}\theta \\
&= \sum \|p - s^{\Theta_i}(p)\|_2 \int_{\Theta_i}^{\Theta_{i+1}} |\cos(\angle(p - s^{\Theta_i}(p), \theta))| \mathrm{d}\theta \\
&\geq \sum \|p - s^{\Theta_i}(p)\|_2 (\Theta_{i+1} - \Theta_i)^2 / 2\pi \\
&\geq (\Theta_{i+1} - \Theta_i)^2 d_1(\mathrm{Dg}_1, \mathrm{Dg}_2) / 2\pi,
\end{aligned}$$

where the sum is again taken over all $p \in \mathrm{Dg}_1 \cup \pi_\Delta(\mathrm{Dg}_2)$, and where the inequality used to lower bound the integral of the cosine is obtained by concavity. The lower bound follows then from the Cauchy-Schwarz inequality. □

Note that the lower bound depends on the cardinalities of the PDs, and it becomes close to 0 if the PDs have a large number of points. On the other hand, the upper bound is oblivious to the cardinality. A corollary of Theorem 3.3 is that $d_{k_{\mathrm{SW}}}$, the distance induced by k_{SW} in its RKHS, is also equivalent to d_1 in a broader sense: there exist continuous, positive and monotone functions g, h such that $g(0) = h(0) = 0$ and $g \circ d_1 \leq d_{k_{\mathrm{SW}}} \leq h \circ d_1$.

When the condition on the cardinalities of PDs is relaxed, e.g. when we only assume the PDs to be finite and bounded, with no uniform bound, the feature map ϕ_{SW} associated to k_{SW} remains continuous and injective w.r.t. d_1. This means that k_{SW} can be turned into a universal kernel by considering $\exp(k_{\mathrm{SW}})$ (cf Theorem 1 in (Kwitt et al., 2015)). This can be useful in a variety of tasks, including tests on distributions of PDs.

Computation. In practice, we propose to approximate k_{SW} in $O(N\log(N))$ time using Algorithm 1. This algorithm first samples M directions in the half-circle $\mathbb{S}_1^+$; it then computes, for each sample θ_i and for each PD Dg, the scalar products between the points of Dg and θ_i, to sort them next in a vector $V_{\theta_i}(\mathrm{Dg})$. Finally, the ℓ_1-norm between the vectors is averaged over the sampled directions: $\mathrm{SW}_M(\mathrm{Dg}_1, \mathrm{Dg}_2) = \frac{1}{M}\sum_{i=1}^M \|V_{\theta_i}(\mathrm{Dg}_1) - V_{\theta_i}(\mathrm{Dg}_2)\|_1$. Note that one can easily adapt the proof of Lemma 3.2 to show that SW_M is negative semi-definite by using the linearity of the sum. Hence, this approximation remains a kernel. If the two PDs have cardinalities bounded by N, then the running time of this procedure is $O(MN\log(N))$. This approximation of k_{SW} is useful since, as shown in Section 4, we have observed empirically that just a few directions are sufficient to get good classification accuracies. Note that the exact computation of k_{SW} is also possible in $O(N^2\log(N))$ time using the algorithm described in (Carrière et al., 2017).

Algorithm 1 Computation of SW_M

Input: $\mathrm{Dg}_1 = \{p_1^1 \dots p_{N_1}^1\}, \mathrm{Dg}_2 = \{p_1^2 \dots p_{N_2}^2\}, M$.
Add $\pi_\Delta(\mathrm{Dg}_1)$ to Dg_2 and vice-versa.
Let $\mathrm{SW}_M = 0$; $\theta = -\pi/2$; $s = \pi/M$;
for $i = 1 \dots M$ **do**
  Store the products $\langle p_k^1, \theta\rangle$ in an array V_1;
  Store the products $\langle p_k^2, \theta\rangle$ in an array V_2;
  Sort V_1 and V_2 in ascending order;
  $\mathrm{SW}_M = \mathrm{SW}_M + s\|V_1 - V_2\|_1$;
  $\theta = \theta + s$;
end for
Output: $(1/\pi)\mathrm{SW}_M$;

4. Experiments

In this section, we compare k_{SW} to k_{PSS} and k_{PWG} on several benchmark applications for which PDs have been proven useful. We compare these kernels in terms of classification accuracies and computational cost. We review first our experimental setting, and then all our tasks.

Experimental setting All kernels are handled with the LIBSVM (Chang & Lin, 2011) implementation of C-SVM, and results are averaged over 10 runs on a 2.4GHz Intel Xeon E5530 Quad Core. The cost factor C is cross-validated in the following grid: $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. Table 1 summarizes the number of labels, and the number of training and test instances for each task. Figure 2 illustrate how we use PDs to represent complex data. We first describe the two baselines we considered, along with their parameterization, followed by our proposal.

TASK	TRAINING	TEST	LABELS
ORBIT	175	75	5
TEXTURE	240	240	24
HUMAN	415	1618	8
AIRPLANE	300	980	4
ANT	364	1141	5
BIRD	257	832	4
FOURLEG	438	1097	6
OCTOPUS	334	1447	2
FISH	304	905	3

Table 1. Number of instances in the training set, the test set and number of labels of the different applications.

PSS. The *Persistence Scale Space* kernel k_{PSS} (Reininghaus et al., 2015) is defined as the scalar product of the two solutions of the heat diffusion equation with initial Dirac sources located at the PD points. It has the following closed form expression: $k_{\mathrm{PSS}}(\mathrm{Dg}_1, \mathrm{Dg}_2) = \frac{1}{8\pi t}\sum_{p\in\mathrm{Dg}_1}\sum_{q\in\mathrm{Dg}_2}\exp\left(-\frac{\|p-q\|^2}{8t}\right) - \exp\left(-\frac{\|p-\bar{q}\|^2}{8t}\right)$, where $\bar{q} = (y, x)$ is the symmetric of $q = (x, y)$ along the diagonal. Since there is no clear heuristic on how to tune t, this parameter is chosen in the applications by ten-fold cross-validation with random 50%-50% training-test splits and with the following set of $N_{\mathrm{PSS}} = 13$ values: 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500 and 1000.

PWG. Let K, p, ρ be positive parameters. Let k_ρ be the Gaussian kernel with parameter ρ and associated RKHS $\mathcal{H}_\rho$. Let $\mathrm{Dg}_1, \mathrm{Dg}_2$ be two PDs, and let $\mu_1 := \sum_{x\in\mathrm{Dg}_1}\arctan(Kd_\infty(x,\Delta)^p)k_\rho(\cdot, x) \in \mathcal{H}_\rho$ be the kernel mean embedding of Dg_1 weigthed by the diagonal distances. Let μ_2 be defined similarly.

TASK	k_{PSS} (10^{-3})	k_{PWG} (10^{3})	k_{SW} (6)	k_{PSS} (10^{-3})	k_{PWG} (10^{3})	k_{SW} (6) $-NC$
ORBIT	63.6 ± 1.2	77.7 ± 1.2	$\mathbf{83.7} \pm 0.5$	$N(124 \pm 8.4)$	$N(144 \pm 14)$	415 ± 7.9
TEXTURE	$\mathbf{98.8} \pm 0.0$	95.8 ± 0.0	96.1 ± 0.4	$N(165 \pm 27)$	$N(101 \pm 9.6)$	482 ± 68

TASK	k_{PSS}	k_{PWG}	k_{SW}	k_{PSS}	k_{PWG}	$k_{\text{SW}} - NC$	k_{SW} (10) $-NC$
HUMAN	68.5 ± 2.0	64.2 ± 1.2	$\mathbf{74.0} \pm 0.2$	$N(29 \pm 0.3)$	$N(318 \pm 22)$	2270 ± 336	107 ± 14
AIRPLANE	65.4 ± 2.4	61.3 ± 2.9	$\mathbf{72.6} \pm 0.2$	$N(0.8 \pm 0.03)$	$N(5.6 \pm 0.02)$	44 ± 5.4	10 ± 1.6
ANT	86.3 ± 1.0	87.4 ± 0.5	$\mathbf{92.3} \pm 0.2$	$N(1.7 \pm 0.01)$	$N(12 \pm 0.5)$	92 ± 2.8	16 ± 0.4
BIRD	67.7 ± 1.8	$\mathbf{72.0} \pm 1.2$	67.0 ± 0.5	$N(0.5 \pm 0.01)$	$N(3.6 \pm 0.02)$	27 ± 1.6	6.6 ± 0.8
FOURLEG	67.0 ± 2.5	64.0 ± 0.6	$\mathbf{73.0} \pm 0.4$	$N(10 \pm 0.07)$	$N(113 \pm 13)$	604 ± 25	52 ± 3.2
OCTOPUS	77.6 ± 1.0	78.6 ± 1.3	$\mathbf{85.2} \pm 0.5$	$N(1.4 \pm 0.01)$	$N(11 \pm 0.8)$	75 ± 1.4	14 ± 2.1
FISH	76.1 ± 1.6	$\mathbf{79.8} \pm 0.5$	75.0 ± 0.4	$N(1.2 \pm 0.004)$	$N(9.6 \pm 0.03)$	72 ± 4.8	12 ± 1.1

Table 2. Classification accuracies (%) and Gram matrices computation time (s) for the benchmark applications. As explained in the text, N represents the size of the set of possible parameters, and we have $N = 13$ for k_{PSS}, $N = 5 \times 5 \times 5 = 125$ for k_{PWG} and $N = 3 \times 5 = 15$ for k_{SW}. C is a constant that depends only on the training size. In all our applications, it is less than 0.1s.

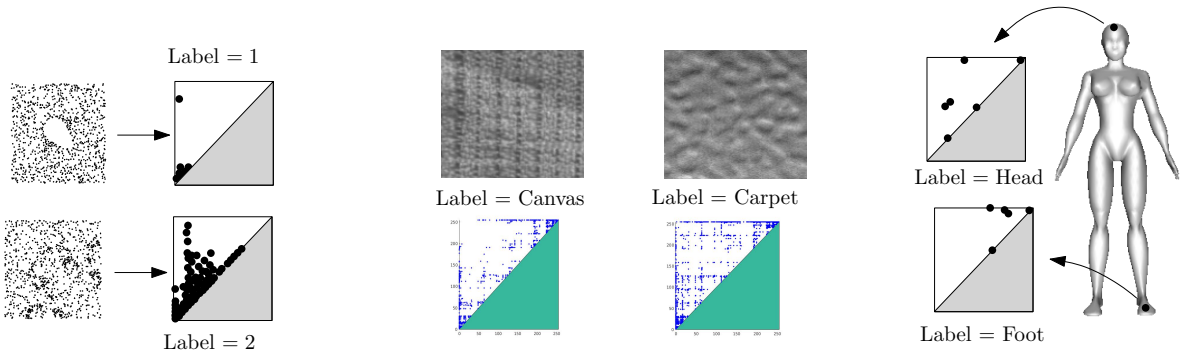


Figure 2. Examples of PDs computed on orbits, texture images and 3D shapes.

Let $\tau > 0$. The *Persistence Weighted Gaussian* kernel k_{PWG} (Kusano et al., 2016; 2017) is defined as $k_{\text{PWG}}(\text{Dg}_1, \text{Dg}_2) = \exp\left(-\frac{\|\mu_1 - \mu_2\|_{\mathcal{H}_\rho}}{2\tau^2}\right)$, i.e. the Gaussian kernel with parameter τ on $\mathcal{H}_\rho$. The authors in (Kusano et al., 2016) provide heuristics to compute K, ρ and τ and give a rule of thumb to tune p. Hence, in the applications we select p according to the rule of thumb, and we use ten-fold cross-validation with random 50%-50% training-test splits to chose K, ρ and τ. The ranges of possible values is obtained by multiplying the values computed with the heuristics with the following range of 5 factors: 0.01, 0.1, 1, 10 and 100, leading to $N_{\text{PWG}} = 5 \times 5 \times 5 = 125$ different sets of parameters.

Parameters for k_{SW}. The kernel we propose has only one parameter, the bandwidth σ in Eq. (5), which we choose using ten-fold cross-validation with random 50%-50% training-test splits. The range of possible values is obtained by computing the squareroot of the median, the first and the last deciles of all $\text{SW}(\text{Dg}_i, \text{Dg}_j)$ in the training set, then by multiplying these values by the following range of 5 factors: 0.01, 0.1, 1, 10 and 100, leading to $N_{\text{SW}} = 5 \times 3 = 15$ possible values.

Parameter Tuning. The bandwidth of k_{SW} is, in practice, easier to tune than the parameters of its two competitors when using grid search. Indeed, as is the case for all infinitely divisible kernels, the Gram matrix does not need to be recomputed for each choice of σ, since it only suffices to compute all the Sliced Wasserstein distances between PDs in the training set once. On the contrary, neither k_{PSS} nor k_{PWG} share this property, and require recomputations for each hyperparameter choice. Note however that this improvement may no longer hold if one uses other methods to tune parameters. For instance, using k_{PWG} without cross-validation is possible with the heuristics given by the authors in (Kusano et al., 2016), and leads to smaller training times, but also to worse accuracies.

4.1. 3D shape segmentation

Our first task, whose goal is to produce point classifiers for 3D shapes, follows that presented in (Carrière et al., 2015).

Data. We use some categories of the mesh segmentation benchmark of Chen et al. (Chen et al., 2009), which contains 3D shapes classified in several categories (“airplane”, “human”, “ant”...). For each category, our goal is to design a classifier that can assign, to each point in the shape, a

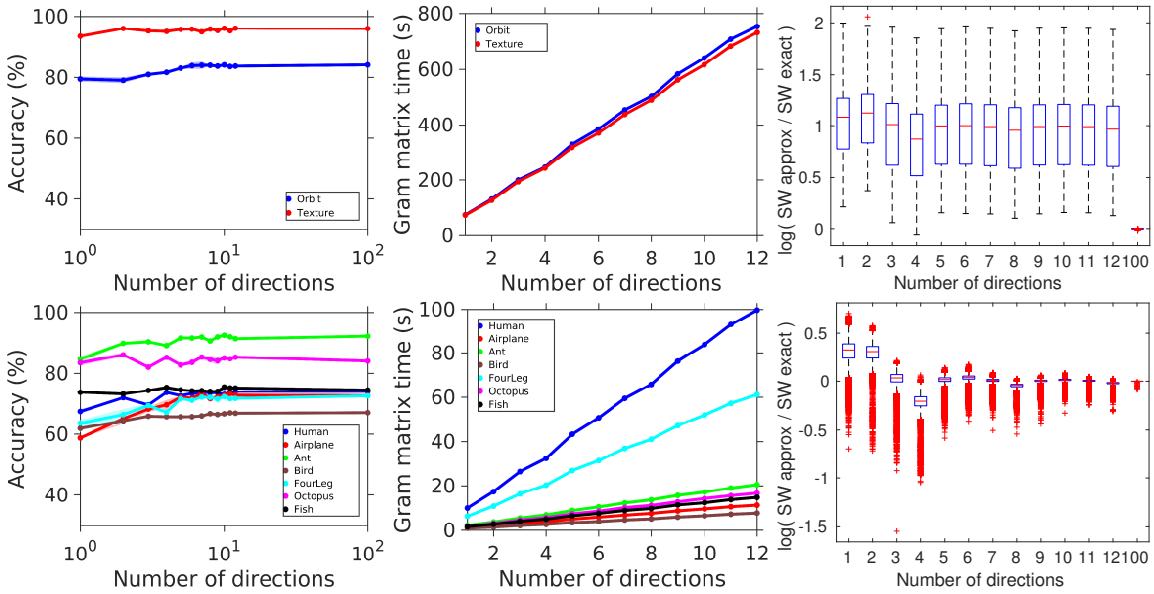


Figure 3. The first row corresponds to the orbit recognition and the texture classification while the second row corresponds to 3D shape segmentation. On each row, the left plot shows the dependence of the accuracy on the number of directions, the middle plot shows the dependence of a single Gram matrix computation time, and the right plot shows the dependence of the ratio of the approximation of SW and the exact SW. Since the box plot of the ratio for orbit recognition is very similar to that of 3D shape segmentation, we only give the box plot of texture classification in the first row.

label that describes the relative location of that point in the shape. For instance, possible labels are, for the human category, "head", "torso", "arm"... To train classifiers, we compute a PD per point using the geodesic distance function to this point—see (Carrière et al., 2015) for details. We use 1-dimensional persistent homology (0-dimensional would not be informative since the shapes are connected, leading to solely one point with coordinates $(0, +\infty)$ per PD). For each category, the training set contains one hundredth of the points of the first five 3D shapes, and the test set contains one hundredth of the points of the remaining shapes in that category. Points in training and test sets are evenly sampled. See Figure 2. Here, we focus on comparison between PDs, and not on achieving state-of-the-art results. It has been proven that PDs bring complementary information to classical descriptors in this task—see (Carrière et al., 2015), hence reinforcing their discriminative power with appropriate kernels is of great interest. Finally, since data points are in $\mathbb{R}^3$, we set the p parameter of k_{PWG} to 5.

Results. Classification accuracies are given in Table 2. For most categories, k_{SW} outperforms competing kernels by a significant margin. The variance of the results over the run is also less than that of its competitors. However, training times are not better in general. Hence, we also provide the results for an approximation of k_{SW} with 10 directions. As one can see from Table 2 and from Figure 3, this approximation leaves the accuracies almost unchanged, while the training times become comparable with the ones of the other competitors. Moreover, according to Figure 3, using even less directions would slightly decrease the accuracies, but still outperform the competitors performances, while decreasing even more the training times.

4.2. Orbit recognition

In our second experiment, we use synthetized data. The goal is to retrieve parameters of dynamical system orbits, following an experiment proposed in (Adams et al., 2017).

Data. We study the *linked twist map*, a discrete dynamical system modeling fluid flow. It was used in (Hertzsch et al., 2007) to model flows in DNA microarrays. Its orbits can be computed given a parameter $r > 0$ and initial positions $(x_0, y_0) \in [0, 1] \times [0, 1]$ as follows:

$$\begin{cases} x_{n+1} = x_n + r y_n (1 - y_n) & \mod 1 \\ y_{n+1} = y_n + r x_{n+1} (1 - x_{n+1}) & \mod 1 \end{cases}$$

Depending on the values of r, the orbits may exhibit very different behaviors. For instance, as one can see in Figure 2, when r is 2, there seems to be no interesting topological features in the orbit, while voids form when r is 1. Following (Adams et al., 2017), we use 5 different parameters $r = 2.5, 3.5, 4, 4.1, 4.3$, that act as labels. For each parameter, we generate 100 orbits with 1000 points and random initial positions. We then compute the PDs of the distance functions to the point clouds with the GUDHI

library (The GUDHI Project, 2015) and we use them (in all homological dimensions) to produce an orbit classifier that predicts the parameter values, by training over a 70%-30% training-test split of the data. Since data points are in $\mathbb{R}^2$, we set the p parameter of k_{PWG} to 4.

Results. Since the PDs contain thousands of points, we use kernel approximations to speed up the computation of the Gram matrices. In order for the approximation error to be bounded by 10^{-3}, we use an approximation of k_{SW} with 6 directions (as one can see from Figure 3, this has a small impact on the accuracy), we approximate k_{PWG} with 1000 random Fourier features (Rahimi & Recht, 2008), and we approximate k_{PSS} using Fast Gauss Transform (Morariu et al., 2009) with a normalized error of 10^{-10}. One can see from Table 2 that the accuracy is increased a lot with k_{SW}. Concerning training times, there is also a large improvement since we tune the parameters with grid search. Indeed, each Gram matrix needs not be recomputed for each parameter when using k_{SW}.

4.3. Texture classification

Our last experiment is inspired from (Reininghaus et al., 2015) and (Li et al., 2014). We use the *OUTEX00000* data base (Ojala et al., 2002) for texture classification.

Data. PDs are obtained for each texture image by computing first the sign component of CLBP descriptors (Guo et al., 2010) with radius $R = 1$ and $P = 8$ neighbors for each image, and then compute the persistent homology of this descriptor using the GUDHI library (The GUDHI Project, 2015). See Figure 2. Note that, contrary to the experiment of (Reininghaus et al., 2015), we do not downsample the images to 32×32 images, but keep the original 128×128 images. Following (Reininghaus et al., 2015), we restrict the focus to 0-dimensional persistent homology. We also use the first 50%-50% training-test split given in the database to produce classifiers. Since data points are in $\mathbb{R}^2$, we set the p parameter of k_{PWG} to 4.

Results We use the same approximation procedure as in Section 4.2. According to Figure 3, even though the approximation of SW is rough, this has again a small impact on the accuracy, while reducing the training time by a significant margin. As one can see from Table 2, using k_{PSS} leads to almost state-of-the-art results (Ojala et al., 2002; Guo et al., 2010), closely followed by the accuracies of k_{SW} and k_{PWG}. The best timing is given by k_{SW}, again because we use grid search. Hence, k_{SW} almost achieves the best result, and its training time is better than the ones of its competitors, due to the grid search parameter tuning.

Metric Distortion. To illustrate the equivalence theorem, we also show in Figure 4 a scatter plot where each point represents the comparison of two PDs taken from the Airplane segmentation data set. Similar plots can be obtained with the other datasets considered here. For all points, the x-axis quantifies the first diagram distance d_1 for that pair, while the y-axis is the logarithm of the RKHS distance induced by either k_{SW}, k_{PSS}, k_{PWG} or a Gaussian kernel directly applied to d_1, to obtain comparable quantities. We use the parameters given by the cross-validation procedure described above. One can see that the distances induced by k_{SW} are less spread than the others, suggesting that the metric induced by k_{SW} is more discriminative. Moreover the distances given by k_{SW} and the Gaussian kernel on d_1 exhibit the same behavior, suggesting that k_{SW} is the best natural equivalent of a Gaussian kernel for PDs.

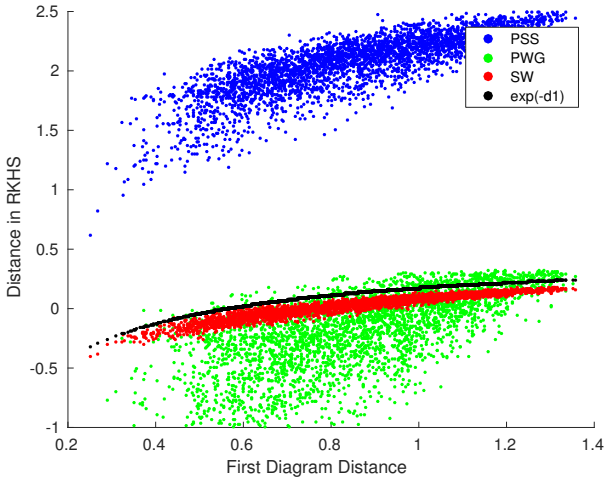


Figure 4. We show how the metric d_1 is distorted. Each point represents a pair of PDs and its abscissae is the first diagram distance between them. Depending on the point color, its ordinate is the logarithm of the distance between PDs in the RKHS induced by either k_{PSS} (blue points), k_{PWG} (green points), k_{SW} (red points) and a Gaussian kernel on d_1 (black points).

5. Conclusion

In this article, we introduce the *Sliced Wasserstein kernel*, a new kernel for PDs that is provably *equivalent* to the first diagram distance between PDs. We provide fast algorithms to approximate it, and show on several datasets substantial improvements in accuracy and training times (when tuning parameters is done with grid search) over competing kernels. A particularly appealing property of that kernel is that it is infinitely divisible, substantially facilitating the tuning of parameters through cross validation.

Acknowledgements. We thank the anonymous referees for their insightful comments. SO was supported by ERC grant Gudhi and by ANR project TopData. MC was supported by a *chaire de l'IDEX Paris Saclay*.

References

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

Bauer, U. and Lesnick, M. Induced matchings and the algebraic stability of persistence barcodes. *Journal of Computational Geometry*, 6(2):162–191, 2015.

Berg, C., Christensen, J., and Ressel, P. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*. Springer, 1984.

Bubenik, P. Statistical Topological Data Analysis using Persistence Landscapes. *Journal Machine Learning Research*, 16:77–102, 2015.

Carlsson, G. Topology and data. *Bulletin American Mathematical Society*, 46:255–308, 2009.

Carrière, M., Oudot, S., and Ovsjanikov, M. Stable Topological Signatures for Points on 3D Shapes. In *Proceedings 13th Symposium Geometry Processing*, 2015.

Carrière, M., Cuturi, M., and Oudot, S. Sliced Wasserstein Kernel for Persistence Diagrams. *CoRR*, abs/1706.03358, 2017.

Chang, C. and Lin, C. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chazal, F., Cohen-Steiner, D., Glisse, M., Guibas, L., and Oudot, S. Proximity of persistence modules and their diagrams. In *Proceedings 25th Symposium Computational Geometry*, pp. 237–246, 2009a.

Chazal, F., Cohen-Steiner, D., Guibas, L., Mémoli, F., and Oudot, S. Gromov-Hausdorff Stable Signatures for Shapes using Persistence. *Computer Graphics Forum*, pp. 1393–1403, 2009b.

Chazal, F., de Silva, V., and Oudot, S. Persistence stability for geometric complexes. *Geometriae Dedicata*, pp. 1–22, 2013.

Chazal, F., de Silva, V., Glisse, M., and Oudot, S. *The structure and stability of persistence modules*. Springer, 2016.

Chen, X., Golovinskiy, A., and Funkhouser, T. A Benchmark for 3D Mesh Segmentation. *ACM Trans. Graph.*, 28(3):73:1–73:12, 2009.

Cohen-Steiner, D., Edelsbrunner, H., and Harer, J. Stability of persistence diagrams. *Discrete Computational Geometry*, 37(1):103–120, 2007.

Edelsbrunner, H. and Harer, J. *Computational Topology: an introduction*. AMS Bookstore, 2010.

Edelsbrunner, Herbert and Harer, John. Persistent homology-a survey. *Contemporary mathematics*, 453: 257–282, 2008.

Fabio, B. Di and Ferri, M. Comparing persistence diagrams through complex vectors. *CoRR*, abs/1505.01335, 2015.

Guo, Z., Zhang, L., and Zhang, D. A completed modeling of local binary pattern operator for texture classification. *IEEE Trans. Image Processing*, pp. 1657–1663, 2010.

Hertzsch, J.-M., Sturman, R., and Wiggins, S. DNA microarrays: design principles for maximizing ergodic, chaotic mixing. In *Small*, volume 3, pp. 202–218, 2007.

Hiraoka, Y., Nakamura, T., Hirata, A., Escolar, E., Matsue, K., and Nishiura, Y. Hierarchical structures of amorphous solids characterized by persistent homology. In *Proceedings National Academy of Science*, volume 26, 2016.

Kusano, G., Fukumizu, K., and Hiraoka, Y. Persistence Weighted Gaussian Kernel for Topological Data Analysis. In *Proceedings 33rd International Conference on Machine Learning*, pp. 2004–2013, 2016.

Kusano, G., Fukumizu, K., and Hiraoka, Y. Kernel method for persistence diagrams via kernel embedding and weight factor. *CoRR*, abs/1706.03472, 2017.

Kwitt, Roland, Huber, Stefan, Niethammer, Marc, Lin, Weili, and Bauer, Ulrich. Statistical Topological Data Analysis - A Kernel Perspective. In *Advances in Neural Information Processing Systems 28*, pp. 3070–3078, 2015.

Li, C., Ovsjanikov, M., and Chazal, F. Persistence-Based Structural Recognition. In *Proceedings Conference Computer Vision Pattern Recognition*, pp. 2003–2010, 2014.

Morariu, V., Srinivasan, B., Raykar, V., Duraiswami, R., and Davis, L. Automatic online tuning for fast Gaussian summation. In *Advances Neural Information Processing Systems 21*, pp. 1113–1120, 2009.

Ojala, T., Mäenpää, T., Pietikäinen, M., Viertola, J., Kyllönen, J., and Huovinen, S. Outex - new framework for empirical evaluation of texture analysis algorithms. In *Proceedings 16th International Conference Pattern Recognition*, pp. 701–706, 2002.

Oudot, S. *Persistence Theory: From Quiver Representations to Data Analysis*. American Mathematical Society, 2015.

Rabin, J., Peyré, G., Delon, J., and Bernot, M. Wasserstein barycenter and its application to texture mixing. In *International Conference Scale Space Variational Methods Computer Vision*, pp. 435–446, 2011.

Rahimi, A. and Recht, B. Random Features for Large-Scale Kernel Machines. In *Advances Neural Information Processing Systems 20*, pp. 1177–1184, 2008.

Reininghaus, J., Huber, S., Bauer, U., and Kwitt, R. A Stable Multi-Scale Kernel for Topological Machine Learning. *CoRR*, abs/1412.6821, 2014.

Reininghaus, J., Huber, S., Bauer, U., and Kwitt, R. A Stable Multi-Scale Kernel for Topological Machine Learning. In *Proceedings Conference Computer Vision Pattern Recognition*, 2015.

Robins, V. and Turner, K. Principal Component Analysis of Persistent Homology Rank Functions with case studies of Spatial Point Patterns, Sphere Packing and Colloids. *Physica D: Nonlinear Phenomena*, 334:1–186, 2016.

Santambrogio, Filippo. Optimal transport for applied mathematicians. *Birkäuser*, 2015.

Singh, G., Memoli, F., Ishkhanov, T., Sapiro, G., Carlsson, G., and Ringach, D. Topological analysis of population activity in visual cortex. *Journal of Vision*, 8, 2008.

The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015. URL http://gudhi.gforge.inria.fr/doc/latest/.

Villani, C. *Optimal transport : old and new*. Springer, 2009.

Zomorodian, Afra and Carlsson, Gunnar. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.