Reeb graph metrics: a survey

immediate

February 15, 2021

Abstract

We survey the available options for reeb graph metrics

Contents

1 Introduction

In numerous application fields, there is an increasing need to analyze topological and geometric information about shapes. Given a real-valued function on a topological space, only commonly used object for such

analysis is the Reeb graph, which encodes the changing component structure of the level sets of the object. The resulting graph also inherits a real-valued function from this setup. Reeb graphs are utilized in a variety of computational topology and topological data analysis applications in order to get a lower dimension representation of a structure which maintains topological properties of the original data, such as shape analysis [32, 38], data skeletonization [16, 36], and surface denoising [57].

The Reeb graph is constructed on a data set known as an $\mathbb{R}$-space, which is an assignment of scalar data to each point of a topological space. More formally, we say an $\mathbb{R}$-space is a pair $(\mathbb{X}, f)$, where $\mathbb{X}$ is a topological space and $f : \mathbb{X} \to \mathbb{R}$ is a continuous, scalar valued function. In physics and other applied settings, $\mathbb{R}$-spaces are more commonly referred to as *scalar fields*. While the definitions of these two objects are identical, we often think of scalar fields having some additional structure on the space, such as being a simply connected domain. In particular, many common physical phenomena, such as temperature of a surface or distribution of pressure in a liquid, can be described using scalar fields.

As Reeb graphs have become more popular for visualization, analysis, and comparison of such data sets, there has been an increase in interest in defining distances between Reeb graphs or scalar fields; these ideas can also be expanded and modified to many other cases where we have a graph with some form of function defined on it representing the data, such as mapper graphs [17, 24, 25, 27, 39, 47, 52], merge trees/dendrograms [34, 53], contour trees [14], and Morse-Smale complexes [29, 30]. Not only is this interest sparked by mathematical intrigue, but measuring similarity in these topological structures has shown useful when comparing multiple scalar fields together [48, 49, 54].

Researchers have pulled from various areas of mathematical research such as Banach and metrics spaces, category theory, sequence and string matching, and graph theory **[CITE THESE?]** in order to provide inspiration for distances that we can define on Reeb graphs. There are (at least) three different distances which have been defined on Reeb graphs which the literature has shown to be promising: the interleaving distance [51], the functional distortion distance [3], and the Reeb graph edit distance [26]. **[Erin: should we add more citations, or just the first paper of each distance here?]** Each distance has been shown to be both stable and be more discriminative than the well-studied bottleneck distance [**CITE BOTTLENECK DISTANCE**], leading us to believe that these metrics can be useful in different application settings.

Unfortunately, with the varying fields that these metrics have been pulled from, it is not always immediately clear how these metrics relate to one another or which may work better on certain types of data. Several papers have proven bounds comparing the distances [3, 4], yet there is a lack of a cohesive story for the landscape of these distances. Furthermore, while each metric has been heavily researched in recent years, the computational difficulties and overall complexity of these metrics have introduced challenges for both the applied researcher intending to compute these distances, as well as the newcomer who is attempting to develop an intuition for how these metrics operate.

Our contribution This work is a constructive survey focusing on the three aforementioned distances, as well as an analysis of the properties of each metric so that we can better understand the relationship and use cases for each. Specifically, this paper

- provides concrete examples for these distances to help develop the intuition of new researchers;
- provides returning researchers a reference for fundamental properties of eacht metric;
- compares and contrasts the various metrics and introduce a common nomenclature for their properties in general;
- provides guidelines for which applied scenarios each metric would be well-suited;

- discusses the computational hurdles and the literature of possible approximations and/or simpler cases for each metric (merge tree, contour tree, etc.). **[Josh: "etc." means what here?]**

1.1 Notation

Throughout the several papers we survey, Reeb graphs not only have different notation, but the interpretation of them varies. Here, we list the various notations we will use throughout this document:

- $\mathbb{X}$ and $\mathbb{Y}$ generally refer to a compact 2-manifold (surface) without boundary, unless otherwise noted;
- $(\mathbb{X}, f)$ is an $\mathbb{R}$-space/scalar field;
- $\mathcal{R}_f$ is the *topological* Reeb graph of $(\mathbb{X}, f)$ – the Reeb graph viewed as an $\mathbb{R}$-space;
- F is the *abstract* Reeb graph of $(\mathbb{X}, f)$ – the Reeb graph viewed as a cosheaf;
- Γ_f is the *combinatorial* Reeb graph of $(\mathbb{X}, f)$ – the Reeb graph viewed as a labeled multigraph;

Each distance treats the Reeb graph differently. However, we will always finalize our distance measures as simply being a distance between the topological Reeb graphs, or the scalar fields themselves, since both the abstract and combinatorial Reeb graph are more "restricted" versions of the topological Reeb graph. Furthermore, unless otherwise noted, we assume that two Reeb graphs $\mathcal{R}_f, \mathcal{R}_g$ are defined on possibly different spaces $\mathbb{X}, \mathbb{Y}$.

2 Basic Definitions

2.1 Scalar Fields and Reeb Graphs

Definition 2.1. *A **scalar field** (equivalently an $\mathbb{R}$-**space**) is a pair $(\mathbb{X}, f)$ where $\mathbb{X}$ is topological space and $f : \mathbb{X} \to \mathbb{R}$ is a continuous real-valued function.*

Definition 2.2. *We define an equivalence relation $\sim_f$ on $\mathbb{X}$ by stating that $x \sim_f y$ if $f(x) = f(y) = a$ and x and y both lie in the same connected component of the levelset $f^{-1}(a)$. We define $\mathbb{X}_f$ to be the quotient space $\mathbb{X}/\sim_f$ and define $\tilde{f} : \mathbb{X}_f \to \mathbb{R}$ to be the restriction of f to the domain $\mathbb{X}_f$. The pair $\mathcal{R}_f := (\mathbb{X}_f, \tilde{f})$ is called the **Reeb Graph** of $(\mathbb{X}, f)$.*

Without sufficient restrictions on $(\mathbb{X}, f)$, it is possible to have Reeb graphs which are not well-behaved[1]. To avoid these situations, we introduce the notion of **constructibilty**. We say that a scalar field is **constructible** if there are a finite number of critical points of $\mathbb{X}$ and there is a cylindrical structure between the critical points. This guarantees that the Reeb graph $\mathcal{R}_f$ is indeed a graph. See [51] for a more in-depth treatment of constructibility.

Examples of constructible scalar fields/$\mathbb{R}$-spaces include piecewise linear functions defined on compact polyhedra and **Morse functions** defined on compact manifolds.

Definition 2.3. *A Morse function is a smooth function $f : \mathbb{X} \to \mathbb{R}$ defined on a manifold $\mathbb{X}$ such that all critical points are non-degenerate and all critical points have distinct function values.*

[1] For example, if $\mathbb{X}$ is the unit disk minus the origin and $f(x, y) = y$, the resulting Reeb graph will be non-Hausdorff.

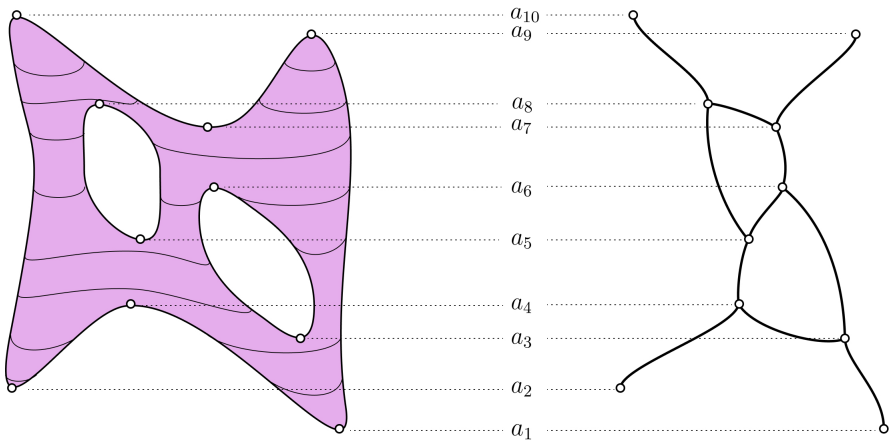


Figure 1: A scalar field $(\mathbb{X}, f)$, where $\mathbb{X}$ is a 2-manifold and f is a Morse function, along with its corresponding Reeb graph $\mathcal{R}_f$.

In an applied setting, it is common for our topological space to be some compact 2-manifold. While many of the definitions can be defined on constructible scalar fields or scalar fields with even less restrictions, in this document **we will consider these scalar fields to be Morse functions defined on compact 2-manifolds**, unless otherwise noted.

Given this restriction of $(\mathbb{X}, f)$, the nodes of the Reeb graph have a well-defined structure to them[2]: a degree 1 node with one edge exiting the node corresponds to an index 0 or minima of the scalar field; a degree 1 node with one edge entering the node corresponds to an index 2 or maxima of the scalar field; a degree 3 node corresponds to a index 1 or saddle point of the scalar field. **[Josh: Ok if we're gonna talk about index, talking about Morse functions makes sense too. I'm not sure we can avoid defining "index"]** Saddle points which have two edges entering the node are called **down-forks** and saddle points which have two edges exiting the node are called **up-forks**.

2.2 Stability

Definition 2.4. *Let $f, g : \mathbb{X} \to \mathbb{R}$ be two real-valued continuous functions defined on the same topological space $\mathbb{X}$. The L^∞ **distance** between f and g is defined as*

$$||f - g||_\infty := \max_{x \in \mathbb{X}} |f(x) - g(x)|.$$

*If d is a metric between the two Reeb graphs $\mathcal{R}_f, \mathcal{R}_g$, we say that d is **stable** if*

$$d(\mathcal{R}_f, \mathcal{R}_g) \leq ||f - g||_\infty$$

Stability helps us ensure that these metrics do not compute arbitrarily large values for two Reeb graphs defined on the same domain. However, each of the metrics that we will define later can be defined on two Reeb graphs that differ in both function *and* the topological space on which they are defined. A metric that is stable does not state any conclusion about these cases.

[2]The Reeb graph for constructible scalar fields will also have a well-defined structure. However, the varying dimension of the space $\mathbb{X}$ will change this labeling of these indices.

2.3 Extended Persistence

Persistence intuitively captures the length of time [**Josh: "time"? intuition isn't here unless you also say "sweep"**] 7 that features of a scalar field (or other data sets [31]) take to disappear once they have been introduced. The notion of persistence has been widely used as a tool in topological data analysis [**CITE**]. The data provided by persistence can be compactly encoded into a **barcode** –where each feature is provided is associated with a horizontal bar corresponding to the length of time that the feature exists in the data – or in a **persistence diagram** – where each feature is a coordinate pair (a, b) with a representing the birth time and b representing the death time.

While a Reeb graph provides a finer grained detail of the topology of a scalar field, the persistence diagram has been proven useful for its computability and the slew of metrics that can be defined on these persistence diagrams which are also computationally tractable [**CITE**]. We refer the reader to [28] for a more detailed overview of persistence diagrams and metrics defined on these diagrams.

2.3.1 Definition

Let $(\mathbb{X}, f)$ be a scalar field with critical points $\{v_1, \ldots, v_k\}$. We define $\mathbb{X}_a := f^{-1}(-\infty, a]$ to be the **sublevel set** of $\mathbb{X}$ at a. Now, let $\{b_0, \ldots, b_k\}$ be a set of real numbers such that

$$b_0 < f(v_1) < b_1 < f(v_2) < \ldots < b_{k-1} < f(v_k) < b_k.$$

This induces a sequence of nested subspaces

$$\emptyset = \mathbb{X}_{b_0} \subset \mathbb{X}_{b_1} \subset \ldots \subset \mathbb{X}_{b_{k-1}} \subset \mathbb{X}_{b_k} = \mathbb{X},$$

called a **filtration** of the scalar field $(\mathbb{X}, f)$. We can then associate each $\mathbb{X}_{b_i}$ with a corresponding homology group $H_n(\mathbb{X}_{b_i})$ for a fixed dimension n to obtain the sequence

$$\emptyset = H_d(\mathbb{X}_{b_0}) \to H_d(\mathbb{X}_{b_1}) \to \ldots \to H_d(\mathbb{X}_{b_{k-1}}) \to H_d(\mathbb{X}_{b_k}),$$

where each arrow between homology group represents the homomorphism $h_d^{i,j} : H_d(\mathbb{X}_{b_i}) \to H_d(\mathbb{X}_{b_j})$ induced by the inclusion $\mathbb{X}_{b_i} \subset \mathbb{X}_{b_j}$. Note that we have chosen these $\{b_0, \ldots, b_k\}$ to be specifically interleaved between the critical values of f so that the homology of the sequence changes at every iteration [3].

Definition 2.5. *The* ***d^{th}-persistent homology groups*** *are the images of the homology group homomorphisms, $H_d^{i,j} := \mathsf{Im}(h_d^{i,j})$ and the* ***d^{th}-persistent Betti numbers*** *are their corresponding ranks, $\beta_d^{i,j} = \mathsf{Rank}(H_d^{i,j})$.*

The classes of homology groups intuitively represent various n-dimensional holes in the surface. Classes of H_0 represent connected components and classes of H_1 represent closed loops. The Betti numbers corresponding to a particular topological space then simply count the number of holes. Thus, the 0^{th} Betti numbers tell us the number of connected components, while the 1^{st} Betti numbers tell us the number of loops, and so on.

Definition 2.6. *We say that a class α is* ***born*** *at i if $\alpha \in H_d(\mathbb{X}^i) - \mathsf{Im}(h_d^{i-1,i})$. We say that α* ***dies*** *at j if $h_d^{i,j-1}(\alpha) \notin \mathsf{Im}(h_d^{i-1,j-1}(\alpha))$ but $h_d^{i,j}(\alpha) \in \mathsf{Im}(h_d^{i-1,j}(\alpha))$. The* ***persistence*** *of α is $f(v_j) - f(v_i)$. If α never dies, then we say that the persistence of α is $+\infty$.*

[3] The homology groups do not necessarily change, but the only possible places that these sublevelsets have different topologies is when we pass over critical points. Choosing values that are not surrounding the critical points would cause our sequence to have multiple homology groups that are guaranteed to be repeated.

Low persistence features are often attributed to *noise* or *insignificant* features of the data being studied, while high persistence features are often associated with *significant* features. **[Josh: "significant" perhaps? essentially persistence relativizes the size of a feature, but i'm not convinced we do our readers any favors by calling this "important"][Brian: Kind of fixed? Just changed to significant with no other changes 02-04-21]**

Definition 2.7. *The* ***d^{th}-persistence diagram*** *of f, denoted as $Dgm_d(f)$, is a scatterplot of its persistent features that records each feature α of the d^{th}-persistent homology groups as a coordinate pair $(f(v_i), f(v_j))$, where α is born in $\mathbb{X}_i$ and dies entering $\mathbb{X}_j$, in the extended plane $\bar{\mathbb{R}}^2 := \mathbb{R} \cup \{+\infty\} \times \mathbb{R} \cup \{+\infty\}$.*

If the class α is born at $\mathbb{X}_i$ and never dies, it is represented as the coordinate $(f(v_i), +\infty)$ in the persistence diagram. Such classes are known as **essential** homology classes. The ordered pairs of a persistence diagram correspond directly to pairs of critical points in a scalar field, which in turn correspond to nodes in the Reeb graph. Index 0 critical points create connected components (0-dimensional homology classes) which are then destroyed by down-forks while up forks create closed loops (1-dimensional homology classes) which are destroyed by index 2 critical points. However, critical points which create essential homology classes are not paired with other critical points in the scalar field like inessential homology classes are. To alleviate this, we can leverage Poincaré and Lefschetz duality to create a new sequence of homology groups where we begin and end with the trivial group [18]. This guarantees that each homology class that is born will also die, meaning each critical point in the Reeb graph will be matched with another critical point at least once.

Let $X^a := f^{-1}[a, \infty)$ be the **superlevel set** of $\mathbb{X}$ at a and let $H_k(\mathbb{X}, \mathbb{X}^a)$ denote the relative homology group. From this, we can create a new sequence of homology groups

$$\begin{aligned} 0 =& H_d(\mathbb{X}_{b_0}) \to H_d(\mathbb{X}_{b_1}) \to \ldots \to H_d(\mathbb{X}_{b_{i-1}}) \to H_d(\mathbb{X}_{b_i}) \\ =& H_d(\mathbb{X}, \mathbb{X}^{b_i}) \to H_d(\mathbb{X}, \mathbb{X}^{b_{i-1}}) \to \ldots \to H_d(\mathbb{X}, \mathbb{X}^{b_1}) \to H_d(\mathbb{X}, \mathbb{X}^{b_0}) = 0. \end{aligned}$$

We can now construct an **extended persistence** diagram, denoted $\text{ExDgm}_d(f)$, from the new pairs in the same fashion as before. To differ between the two types persistence, we will often use the term **ordinary persistence** for the former process. However, we should note that extended persistence is strictly more powerful than ordinary persistence because it captures all inessential homology classes as well as essential homology classes, where as ordinary persistence will only capture inessential classes.

Perliminary results in [1] showed a pairing between all critical points of a 2-manifold while [18] extended this to general manifolds later. We provide a standard example of extended persistence and the process behind it to give the reader some intuition.

Example 2.8. *Figure 1 shows a genus-2 surface embedded in $\mathbb{R}^3$ along with its Reeb graph. To find the persistence pairs, we begin by sweeping upwards and tracking the features which are born and destroyed. Critical points a_1 and a_2 create classes in H_0; a_3, a_5, a_6, a_7, and a_8 create classes in H_1; a_{10} creates a class in H_2. We pair a_2 with a_4 since a_4 merges two connected components together, and we pair a_7 with a_9 since a_9 closes a hole created by a_7. Thus, the ordinary persistence diagram contains only two points. Going downwards, a_{10} destroys the class in H_0 created by a_1; a_6, a_8, a_3 and a_5 destroy the classes in H_1 created by a_3, a_5, a_6 and a_8, respectively; a_{10} and a_4 create classes in H_2 which are destroyed by a_1 and a_2, respectively. Thus, we have the pairs (a_1, a_{10}) and (a_4, a_2) for dimension 0; (a_3, a_6), (a_5, a_8), (a_6, a_3) and (a_8, a_5) for dimension 1; (a_4, a_2) and (a_{10}, a_1) for dimension 2. Figure 2 shows the ordinary and extended persistence diagrams for this example.*

Remark 2.9. ***[Brian: Remark about not using zigzag persistence]***

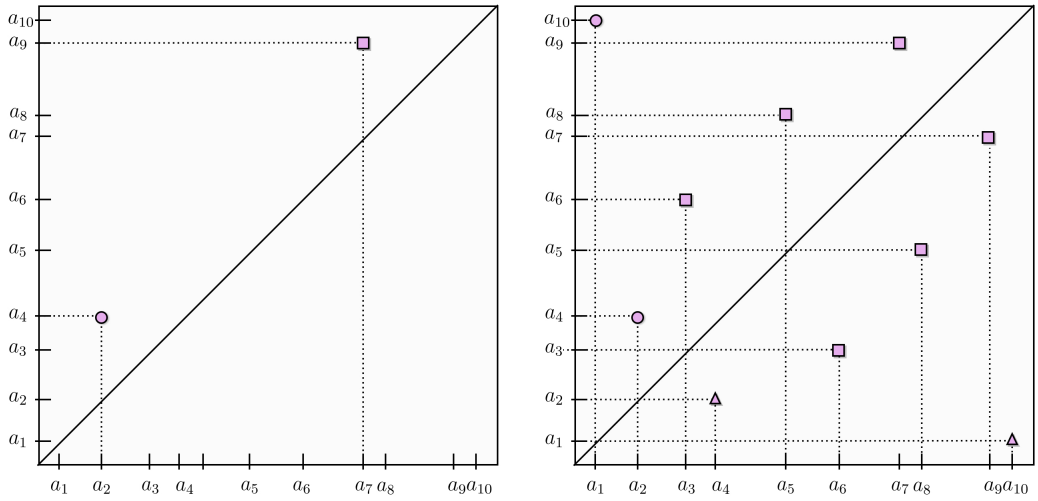


Figure 2: The ordinary persistence and extended persistence diagrams for the surface in Figure 1. Circles denote H_0 classes, squares denote H_1 classes, and triangles denote H_2 classes.

2.3.2 Bottleneck Distance

[Brian: Need history of bottleneck distance?] [Josh: Maybe not a "history", but you probably want to at least say something like "given two topological summaries, it would be natural to ask how to compare. Bottleneck distances offers a solution..." (rewrite this of course). The exact history though i think is closer to "we used a metric to define stability and then hijacked it for comparison" See Cohen-Steiner, Edelsbrunner, and Harer 2005]

Definition 2.10. *Let $(\mathbb{X}, f), (\mathbb{Y}, g)$ be two scalar fields and $X := \mathrm{ExDgm}_d(f), Y := \mathrm{ExDgm}_d(g)$ be the their corresponding extended persistence diagrams. We define the* ***bottleneck distance*** *d_B between these diagrams as*

$$d_B(X, Y) = \inf_{\eta: X \to Y} \sup_{x \in X} ||x - \eta(x)||_\infty,$$

where η is a bijection between X, Y.

Remark 2.11. *For purposes of finding the bottleneck distance, we add infinitely many points to the diagonal of the persistence diagram. This allows the bijection to pair an off-diagonal point to the diagonal.*

Theorem 2.12. *Let $(\mathbb{X}, f), (\mathbb{X}, g)$ be two constructible scalar fields[4] defined on $\mathbb{X}$. Then*

$$d_B(\mathrm{ExDgm}_d(f), \mathrm{ExDgm}_d(g)) \leq ||f - g||_\infty.$$

That is, the bottleneck distance is a stable metric.

[Brian: Need Example]

[Josh: Idea: use this to introduce our first "comparison" example that we'll use in later sections too?]

[4]We can show this result for two tame functions f, g defined on the same simplicial complex. Constructibility is a more restrictive property than tame functions.

Given two persistence diagrams X and Y, computing the bottleneck distance will only reveal the worst-case distance in the best-case matching η. The **degree-q Wasserstein Distance** was introduced to be more sensitive to multiple distances in the best-matching η.

Definition 2.13. *Let $(\mathbb{X}, f), (\mathbb{Y}, g)$ be two scalar fields and $X := \text{ExDgm}_d(f), Y := \text{ExDgm}_d(g)$ be the their corresponding extended persistence diagrams. The **degree-q Wasserstein distance** between X and Y, for any postivie real number q, is defined as*

$$W_q(X,Y) = \left[\inf_{\eta: X \to Y} \sum_{x \in X} ||x - \eta(x)||_\infty^q \right]^{1/q}$$

[Brian: The stability for Wasserstein distance is only proved for Lipschitz continuous function, which I'm not sure how that relates to our own restrictions of constructibility]

3 Interleaving Distance

3.1 History

[Liz: This is direct self-plagarism and requires editing] **[Erin: Took a stab at this on 1/29 - please re-check for coherence]** The interleaving distance on Reeb graphs takes root in earlier work that defines the interleaving distance for persistence modules [15], **[Josh: perhaps we can chop this up by discussing interleaving distance on persistence modules briefly in the last section (and thus fix the self-plagiarism?). Right now, we probably need a better handoff between the two sections]** and is heavily inspired by the subsequent category theoretic treatment [11, 12]. This viewpoint comes from encoding the data of a Reeb graph in a constructible set-valued cosheaf [20–22]. In fact, it is known that this metric is a special case of a more general theory of interleaving distances given on a *category with a flow* [19, 23, 55]; this more general theory also encompases other metrics including ℓ_∞ distance on points or functions, regular Hausdorff distance, and the Gromov-Hausdorff distance [13, 55].

Interleaving metrics have been studied in the context of $\mathbb{R}$-spaces [8], multiparameter persistence modules [42], merge trees [44], and formigrams [40, 41], and on more general category theoretic constructions [9, 50], as well as developed for Reeb graphs [51] **[Erin: Add link to our newer paper for truncated smoothing once we have it]**. There are also interesting restrictions to labeled merge trees, where one can pass to a matrix representation and show that the interleaving distance is equivalent to the point-wise ℓ_∞ distance [35, 45, 56, 58].

On the negative side, it has been shown that Reeb graph interleaving is graph isomorphism complete [6, 51], and that many other variants are also NP-hard [6, 7]. All of this means that these metrics, while mathematically interesting, may not lead to feasible algorithms for comparison and analysis. However, a glimmer of hope arises with work investigating fixed parameter tractable algorithms [33, 56], and comparisons are possible in polynomial time if the Reeb graph has simple enough structure, such as a contour tree or merge tree. **[Josh: Wonders if we should move this paragraph to section 8]**

In addition, notions of similarity for graphs in general, and Reeb graphs in particular, are of pressing interest due to their extensive use in data analysis; in many such settings, we are concerned with questions of quality in the face of noise, and computing approximations to the exact object with may converge nicely in some abstract limit. For example, the interleaving distance has been used in evaluating the quality of the mapper graph [52], which can be proven to be a approximation of the Reeb graph using this metric [10, 46].

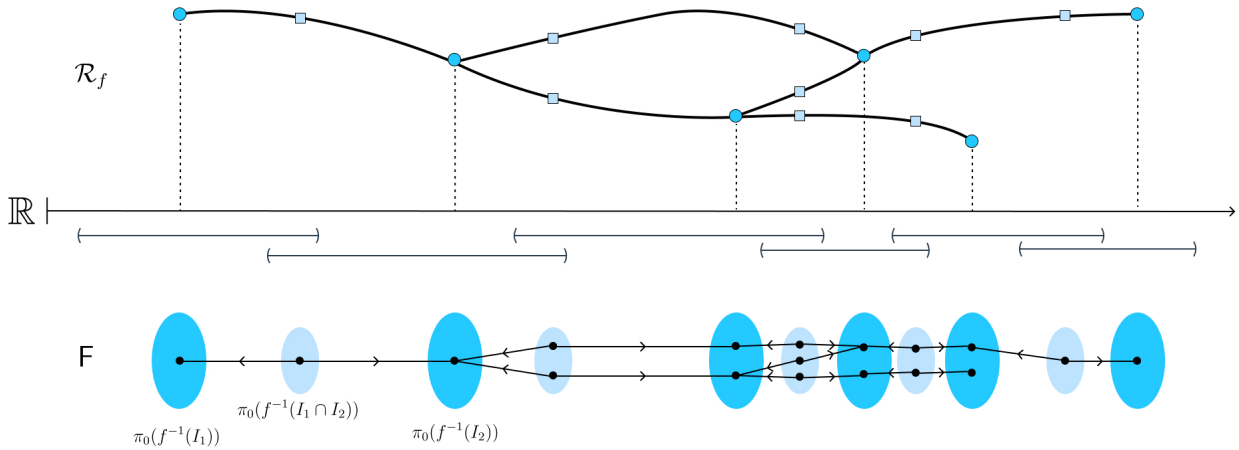


Figure 3: A Reeb graph (Top) with its corresponding cosheaf (bottom). The larger, darker blue sets were chosen specifically to surround the critical points of the Reeb graph. The smaller, light blue sets are the pairwise intersection of the sets surrounding it. The arrows represent the morphisms from each small set to the larger sets they are included in. While a cosheaf is defined for *all* intervals on the real line, the intervals above are enough to capture the topology of the Reeb graph fully. This figure was adapted from [51].

Furthermore, there is considerable interest in unifying the interleaving distance with the emerging collection of other Reeb graph metrics.

3.2 Definition

A **pre-cosheaf** is a functor F from the category of intervals on the real line **Int** to the category of sets **Set** [5]. Intuitively, it is a way to assign data to the open intervals of the real line in a way that respects inclusion of the intervals. Given a Reeb graph $\mathcal{R}_f$ we can construct its pre-cosheaf F by the formulas

$$\mathsf{F}(I) = \pi_0(f^{-1}(I)), \qquad \mathsf{F}[I \subseteq J] = \pi_0[f^{-1}(I) \subseteq f^{-1}(J)],$$

where $\pi_0(U)$ is the set of path connected components of the set U. Figure 3 provides a depiction of converting a Reeb graph $\mathcal{R}_f$ to its pre-cosheaf F.

Stating that two Reeb graphs $R(f), R(g)$ are isomorphic is equivalent to stating that their associated pre-cosheafs F, G are isomorphic. Recall that an isomoprhism between two functors is a pair of natural transformations $\varphi : \mathsf{F} \Rightarrow \mathsf{G}$, $\psi : \mathsf{G} \Rightarrow \mathsf{F}$ such that $\psi_I \circ \varphi_I = \mathbf{Id}_{\mathsf{F}(I)}$ and $\varphi_I \circ \psi_I = \mathbf{Id}_{\mathsf{G}(I)}$, for all $I \in \mathbf{Int}$. If we do not have a true isomorphism between these pre-cosheafs, we can approximate the isomorphisms to form an ε-**interleaving**.

Remark 3.1. *It turns out that the Reeb graph satisfies additional "gluing" constraints which guarantee that its pre-cosheaf is actually a well-defined* ***cosheaf****; see [51] for this equivalence and [20] for a rigorous treatment of cosheafs and their applications to topological data analysis. For the remainder of this document, we will refer to* F *as a cosheaf rather than a pre-cosheaf.*

Definition 3.2. *Let* $I = (a, b) \subseteq \mathbb{R}$ *and* $I^\varepsilon = (a - \varepsilon, b + \varepsilon)$*. The* ε-***smoothing functor***, $\mathcal{S}_\varepsilon : \mathbf{Pre} \to \mathbf{Pre}$*, where* $\varepsilon > 0$*, is defined by* $\mathcal{S}_\varepsilon(\mathsf{F}(I)) = \mathsf{F}(I^\varepsilon)$ *for each* I*.*

[5] In general, we can define pre-cosheafs where the domain category is the open sets of any topological space and the range category is unrestricted.

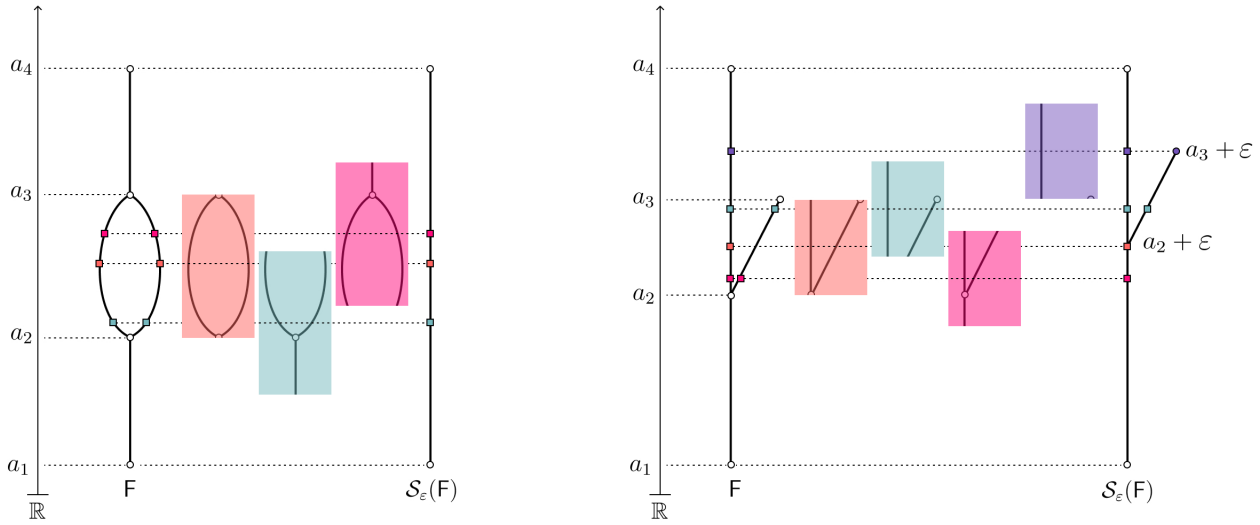
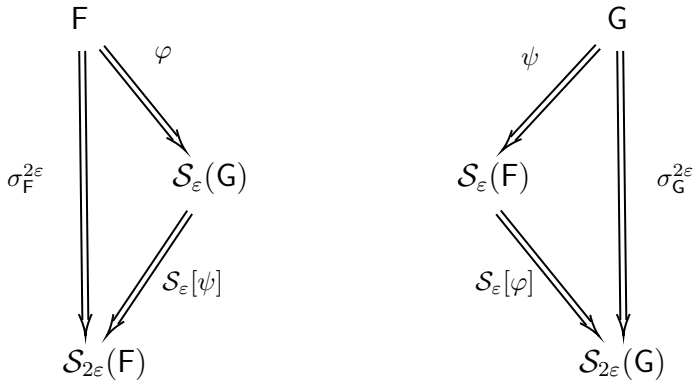


Figure 4: *(Left)* A Reeb graph of a torus along with its ε-smoothed version. To cover the hole completely, ε has to be large enough so that every interval I, the expanded interval I^ε will only have one path connected component. Setting $\varepsilon \geq \frac{a_3-a_2}{2}$ will guarantee this. *(Right)* Leaves of a Reeb graph will be shifted by ε. As the center of the interval passes $\frac{a_3+a_2}{2}$, the number of components changes from one to two, essentially creating a leaf in the smoothed version that is shifted upwards. Note that the last component (purple) maps to only *one* component in the smoothed Reeb graph. Similarly, a leaf pointing downwards will be shifted downwards by ε.

In essence, the ε-smoooothing functor expands each interval I by ε in both directions before assigning data. This implies that the minimum width of an interval is now 2ε rather than a single point. In several cases, the increase in the intervals causes the data associated to these intervals to be fundamentally changed, sometimes removing features entirely. Figure 4 shows two examples of smoothing for various simple features. We discuss the affects of smoothing on larger features composed of multiple smaller features in Section 7. Note that while the smoothing operation is done on the cosheafs directly, we can represent this using the Reeb graph since the fundamental structure of the cosheaf is captured completely in the Reeb graph, as seen in Figure 3.

Definition 3.3. *Let $\sigma^\varepsilon_{\mathsf{F}}$ be the natural transformation $\mathsf{F} \Rightarrow \mathcal{S}_\varepsilon(\mathsf{F})$ created by noting that $I \subseteq I^\varepsilon$ implies $\mathsf{F}(I) \to \mathsf{F}(I^\varepsilon) = \mathcal{S}_\varepsilon(\mathsf{F})$. We say that two cosheafs F, G are ε-**interleaved** if there exists a pair of natural transformations $\varphi : \mathsf{F} \Rightarrow \mathcal{S}_\varepsilon(\mathsf{G})$ and $\psi : \mathsf{G} \Rightarrow \mathcal{S}_\varepsilon(\mathsf{F})$ such that the following diagrams commute:*



If $\varepsilon = 0$, then this is exactly the definition of an isomorphism between F and G. When two cosheafs are ε-interleaved, we say that there exists an ε-**interleaving** between them.

Definition 3.4. *The interleaving distance between two Reeb graphs $\mathcal{R}_f, \mathcal{R}_g$ is the minimum ε such that their respective cosheafs are ε-interleaved. Formally,*

$$d_I(\mathcal{R}_f, \mathcal{R}_g) = \inf_{\varepsilon \in \mathbb{R}^+} \{\varepsilon | \text{ there exists an } \varepsilon\text{-interleaving between } \mathsf{F} \text{ and } \mathsf{G}\},$$

where F, G are their respective cosheafs.

3.3 Properties and Examples

[Josh: Are we going to start this section with an explanation of Figure ???]

Several nice properties are immediate when using the interleaving distance, as well as some drawbacks or cases where it is not useful.

Proposition 3.5 (Proposition 4.3 in [51])**.** *The interleaving distance d_I is an extended pseudometric: it takes values in $[0, \infty]$, is symmetric, satisfies the triangle inequality, and $d_I(\mathsf{F}, \mathsf{F}) = 0$.*

[Erin: Need some transition text here to translate and contextualize the next few theorems, I think [Josh: me too]]

Theorem 3.6 (Theorem 4.4 in [51])**.** *Let $\mathcal{R}_f$ and $\mathcal{R}_g$ be two Reeb graphs defined on the same space $\mathbb{X}$, and let F, G be their respective cosheafs. Then*

$$d_I(\mathcal{R}_f, \mathcal{R}_g) \leq ||f - g||_\infty$$

Proposition 3.7 (Proposition 4.5 in [51])**.** *The interleaving distance between two Reeb graphs is finite if and only if they have the same number of path components.*

Proposition 3.8 (Proposition 4.6 in [51])**.** *The interleaving distance between two Reeb graphs is zero if and only if they are isomorphic.*

- Interleaving distance is NOT invariant to shift in function value, i.e. if $g(x) = f(x) + \delta$, for some $\delta > 0$, then the interleaving distance between the Reeb graphs of $(\mathbb{X}, f)$ and $(\mathbb{X}, g)$ is at least δ (might be exactly delta)
- Let U be an open subset of $\mathbb{X}$ such that $\mathbb{X}$ has a maximum in U. Suppose $f_1 = f_2$ except on U, where $f_1 > f_2$ and is interpolated to the boundary of U so that both functions are still continuous. We can picture this as two identical functions except where one maximum is larger than the maximum located at the same subset of $\mathbb{X}$. The interleaving distance is equal to the distance in the maximum's function value.
- The interleaving distance between a single loop Reeb graph and a single line Reeb graph (assuming that the minimum function value and maximum function value is equal) is $\varepsilon = \delta/2$, where δ is half the distance between the bottom and top of the loop.
- The interleaving distance between a single leaf Reeb graph and a single line Reeb graph (assuming that the minimum function value and maximum function value is equal) is $\varepsilon = \delta/2$, where δ is half the distance between the bottom and top of the leaf.

Use Cases

- Possibly good for time-dependent scalar fields, such as feature tracking over time. Small changes in the features of a scalar field should lead to relatively low interleaving distance
- NOT good for multimodel simulations where we compare various types of data. The difference in scale will certainly cause large interleaving distances

3.4 Extensions

[Brian: Discuss truncated interleaving distance]

4 Functional Distortion Distance

4.1 History

[Brian: I might be wrong on some of these facts, so should double check.]

The functional distortion distance was first defined as a metric in [3], inspired from the well-known Gromov-Hausdorff distance which was first introduced in [37]. The Gromov-Hausdorff distance is a way to measure the distance between two Banach or metric spaces. More formally, suppose A and B are two metric spaces and let $i_A : A \to Z$,$i_B : B \to Z$ be isometric embeddings into a common metric space Z. We can then find the Hausdorff distance between the embeddings: $d_H(i_A(A), i_B(B))$. **[Erin: Just adding a note as I'm reading - have we formally defined Hausdorff distance somewhere?]** **[Josh: Nope. It might be helpful to earlier on as a "warmup" because we could then we could distinguish between d_H and L^∞ in terms of their "pointwise" treatment vs "matching" treatment, as baselines for comparing objects?]**

The goal of the functional distortion distance is then to find the minimum Hausdorff distance achieved by ranging over all possible embeddings and the common space to which they are embedded. Intuitively, we are trying to determine a common area where we can embedd both A and B, while preserving the integrity of the spaces (hence the embeddings being isometries), such that the A and B fit nicely together. We can picture A and B as being two crumpled up pieces of paper (of varying sizes) and our common metric space to be a flat surface. One way to measure the difference in sizes between A and B is to stretch both out flat onto the surface and then compare them that way. Trying to determine the sizes while the paper is still crumpled would be a much more difficult task.

4.2 Definition

The Gromov-Hausdorff (GH) distance has multiple different equivalent definitions; see [43]. Functional distortion distance (FDD) borrows from a very specific variation of the GH distance. Here, we break down the definition into several parts which we will stitch together to form the final definition of FDD.

Definition 4.1. *Let $u, v \in \mathcal{R}_f$ (not necessarily nodes) and let π be a continuous path between u and v. The **range** of this path is the interval* $\text{range}(\pi) = [\min_{x\in\pi} f(x), \max_{x\in\pi} f(x)]$. *The **height** is the length of the*

range, denoted $\text{height}(\pi) = \max_{x\in\pi} f(x) - \min_{x\in\pi} f(x)$. *We define the distance between* u *and* v *to be*

$$d_f(u,v) = \min_{\pi: u\rightsquigarrow v} \text{height}(\pi),$$

where π *ranges over all continuous paths from* u *to* v, *denoted* $u \rightsquigarrow v$, *and* f *in* d_f *refers to the function* f.

Definition 4.2. *Let* $\varphi : \mathcal{R}_f \to \mathcal{R}_g$, $\psi : \mathcal{R}_f \to \mathcal{R}_g$ *be two continuous maps.*[6] *We define the* ***supergraph*** *of* φ *and* ψ *as*

$$G(\varphi,\psi) = \{(x,\varphi(x)) : x \in \mathcal{R}_f\} \cup \{(\psi(y),y) : y \in \mathcal{R}_g\}.$$

$G(\varphi,\psi)$ *is simply the union of the two graphs of* φ *and* ψ.

Definition 4.3. *The* ***point distortion*** λ *between* $(x,y),(x',y') \in G(\varphi,\psi)$ *is defined as*

$$\lambda((x,y),(x',y')) = \frac{1}{2}|d_f(x,x') - d_g(y,y')|.$$

The we define the ***map distortion*** $D(\varphi,\psi)$ *between* $\mathcal{R}_f$ *and* $\mathcal{R}_g$ *to be the supremum of point distortions ranging over all possible pairs in the supergraph* $G(\varphi,\psi)$. *That is,*

$$D(\varphi,\psi) = \sup_{(x,y),(x',y')\in G(\varphi,\psi)} \lambda((x,y),(x',y')).$$

In essence, this distance will measure how much each Reeb graph is being altered to map into the other Reeb graph. For example, if x and x' are relatively close (in terms of d_f) but their outputs under φ are far apart (in terms of d_g), then this distance will be larger. This distance finds the worst case scenario where two points are close on one Reeb graph and their correspondences are far on the other. Keep in mind that there are two maps φ and ψ. Thus, even if both maps are non-surjective, every point of each Reeb graph still has at least one correspondence in the supergraph $G(\varphi,\psi)$.

Definition 4.4. *The* ***functional distortion distance*** *is defined as*

$$d_{FD}(\mathcal{R}_f,\mathcal{R}_g) = \inf_{\varphi,\psi} \max\{D(\varphi,\psi), ||f - g\circ\varphi||_\infty, ||f\circ\psi - g||_\infty\},$$

where φ *and* ψ *range over all continuous maps between* $\mathcal{R}_f$ *and* $\mathcal{R}_g$.

Remark 4.5. *If* φ *and* ψ *are simply translations or negations, the distances* d_f *and* d_g *are not affected since these will preserve the relative closeness of the pairs* (x,x') *and* (y,y'). *In this case,* $D(\varphi,\psi)$ *would be 0. The two terms* $||f - g\circ\varphi||_\infty$ *and* $||f\circ\psi - g||$ *are introduced to address this fact. They simply measure the length of the translation or similar isometries. See Figure 6 for an example.*

Example 4.6. *Figure 5 displays two different pairs of continuous maps between* $\mathcal{R}_f$ *and* $\mathcal{R}_g$. *Note that both* f *and* g *are height functions which are mapping horizontally to the real line, so we have* $f(a_1) = g(a_1)$, *etc. In* (**a**), *the map* φ *maps like an isometry up to* a_2, *where it then maps the rest of* $\mathcal{R}_f$ *into the leaf of* $\mathcal{R}_g$. *The point distortion between the points* (a_3,b_3) *and* (a_2,b_2) *would then be* $|d_f(a_3,a_2) - d_g(b_3,b_2)| = f(a_3) - g(b_3)$. *For* ψ, *the map also acts like an isometry, except the leaf is collapsed, mapping horizontally to* $\mathcal{R}_f$. *In this case, the supergraph contains the points* (a_2',b_3') *and* (a_2',b_3). *Thus, the point distortion between these would be* $|d_f(a_2',a_2') - d_g(b_3',b_3)| = g(b_3) - g(b_2)$. *We can check that other pairs of points from the supergraph will not lead to a higher map distortion. Thererfore,* $D(\varphi,\psi) = \max\{f(a_3) - g(b_3), g(b_3) - g(b_2)\}$. *In* (**b**), *the map* φ *is an isometry and therefore no points in the supergraph which come from* φ *will contribute to the distortion value. The map* ψ *is close to an isometry besides contracting the leaf to a single point. The point distortion between the pairs* (a_2,b_2) *and* (a_2,b_3) *is simply* $g(b_3) - g(b_2)$. *Note that this is the same distortion value which was achieved in part* (a) *when we mapped the leaf straight across to* $\mathcal{R}_f$. *This comes from the definition of* d_g *which only looks at the height of the path that is traversed from one point to the next and does not take into account the total distance traversed.*

[6] These continuous maps need not be function preserving like in the category of Reeb graphs. In other words, these maps are not well-defined morphisms between $\mathcal{R}_f$ and $\mathcal{R}_g$ as objects in **Reeb**.

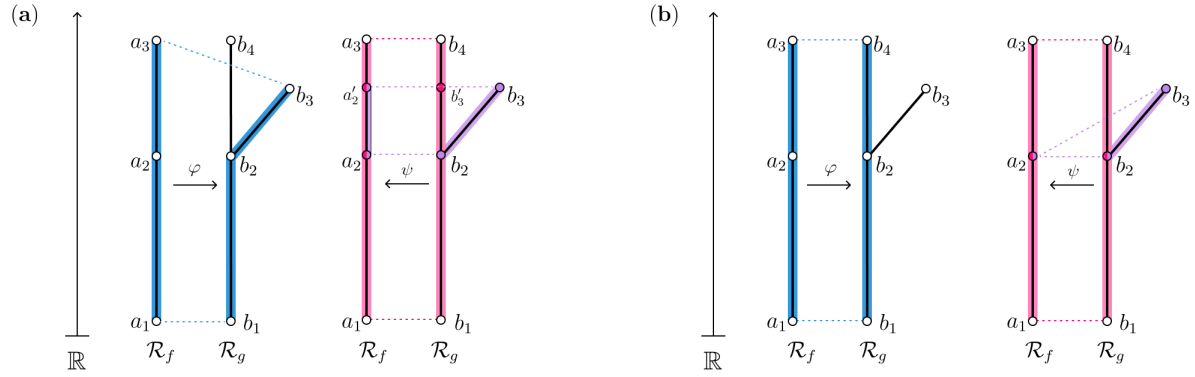


Figure 5: Two examples of continuous maps between simple Reeb graphs $\mathcal{R}_f$ and $\mathcal{R}_g$ as described in example 4.6. Dotted lines indicate the correspondences that these maps are creating, while the color indicates which section we are referring to. (**a**) The map φ distorts $\mathcal{R}_f$ to fit into the leaf on the right side. The map ψ maps each value straight across. (**b**) The map φ is an isometry, which will not alter the distortion alone. The distortion will be solely be based on ψ, which is almost an isometry except for collapsing the leaf to a single point. It turns out that the map contracting the leaf to a single base point at a_2 will result in an equivalent distortion value to mapping it horizontally as in in (a).

Remark 4.7. *When finding continuous maps between two topological Reeb graphs, it is important to remember that the topology defined on the Reeb graph is quotient topology coming from the original compact, 2-manifold $\mathbb{X}$. Let R_g be as in Figure 5 and let ψ be the same map constructed for 5(b). Now, supposed R_g is the Reeb graph of a compact 2-manifold $\mathbb{X}$ and let p be the quotient map carrying $\mathbb{X}$ to $\mathcal{R}_g$. let U be any open set surrounding a_2. Suppose U is an open set containing a_2. Then, $\psi^{-1}(U)$ contains the entire leaf in $\mathcal{R}_g$ from b_2 to b_3. Whether this set is open depends on the structure of the manifold $\mathbb{X}$. If $\mathbb{X}$ is a manifold with no boundary, then $p^{-1}(\psi^{-1}(U))$ will be an open set since any open set V surrounding a point x on the leaf will also be contained within the leaf. Thus, by definition of the quotient map, the set $\psi^{-1}(U)$ is also open. From this, one can draw the conclusion that if $\mathbb{X}$ has no boundary, then the space $\mathcal{R}_g$ also has no boundary.*

4.3 Properties

5 Edit Distance

5.1 History

[Brian: Need History]

5.2 Combinatorial Reeb Graph Edit Distance

Let $(\mathbb{X}, f)$ be a scalar field and $\mathcal{R}_f$ be it's Reeb graph. As stated before, $(\mathbb{X}, f)$ being constructible means that our Reeb graph will be a well-defined graph with a finite number of nodes and edges. While functional distortion distance considered $\mathcal{R}_f$ as a topological space and the interleaving distance considered $\mathcal{R}_f$ as a cosheaf F, here we only need to consider it as a graph, known as the **combinatorial Reeb graph**.

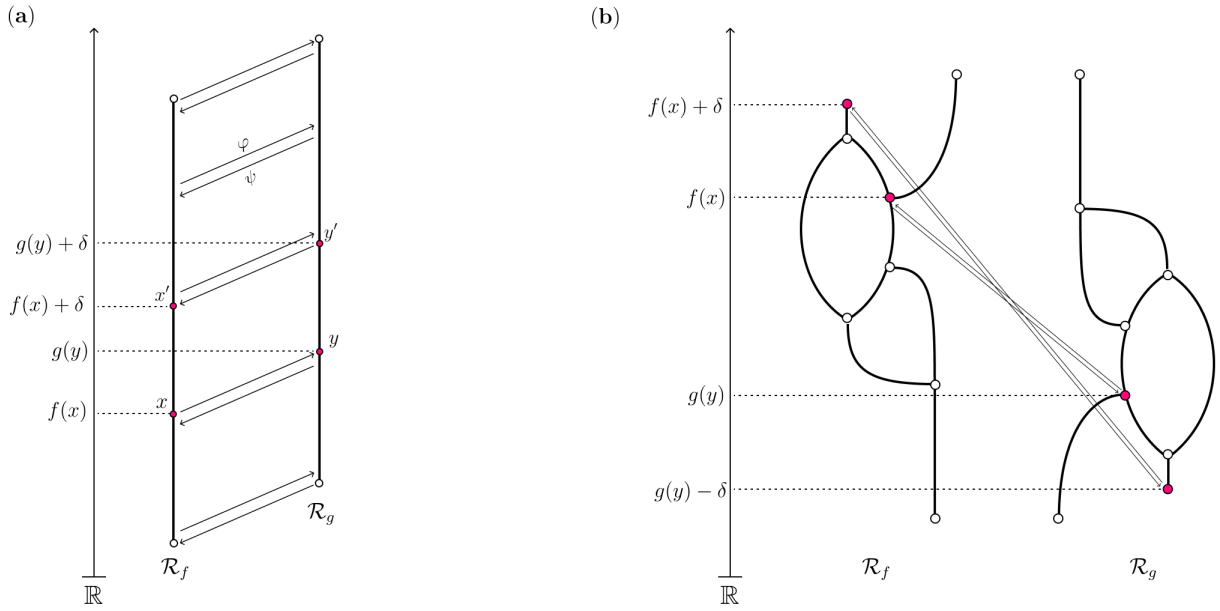


Figure 6: **(a)** An example of maps where φ is a translation and $\varphi = \psi^{-1}$. In this, $d_f(x, x') = \delta = d_g(y, y')$, which implies that $D(\varphi, \psi) = 0$. Without the $||f - g \circ \varphi||_\infty, ||f \circ \psi - g||_\infty\}$ terms, the functional distortion distance between these two Reeb graphs would be 0. With these terms included, we can see that the functional distortion distance will be δ, which is equal to the magnitude of the translation. **(b)** An example of maps where φ and ψ are negations of each other. Because of this, the, the point distortion between any two pairs of points will always be 0.

The construction of Reeb graph edit distance above is the generalized version of the edit distance defined previously in [2]. This distance takes a form closer to that of the original graph edit distance as in it is defined by a set of legal operations performed on the combinatorial Reeb graph Γ_f such as inerstion/deletion of nodes, insertion/deletion of edges, insertaion/deletion of loops, and so on.

Definition 5.1. *A **multigraph** is a graph $\Gamma = (V, E)$, with vertex set V and edge set E, such that the edges in E need not be unique. A **labeled multigraph** is a pair (Γ, l), where Γ is a multigraph (V, E) and l is a scalar function $l : V(\Gamma) \to \mathbb{R}$ defined on the vertices of the multigraph.*

Definition 5.2. *Let $(\mathbb{X}, f)$ be a constructible scalar field and $\mathcal{R}_f$ be its Reeb graph. The **combinatorial Reeb graph**, (Γ_f, l_f) is the labeled multigraph $(V(\Gamma_f), E(\Gamma_f))$, where $V(\Gamma_f)$ and $E(\Gamma_f)$ are the vertices (critical points) and edges of $\mathcal{R}_f$, respectively, and $l_f = f|_{V(\Gamma_f)}$. That is, the labeling is simply the function f restricted to the vertices of the graph.*

5.2.1 Elementary Deformations

The operations that are permitted to be performed on the graph Γ are known as **elementary deformation**, which we define below.

1. **Insert Operations:**

   Definition 5.3. *We define a **vertex insertion** I_v to be any transformation T of (Γ, l) such that for a fixed edge $e(v_1, v_2) \in E(\Gamma)$, with $l(v_1) > l(v_2)$, $T(\Gamma, l)$ is a labeled graph (Γ', l') defined as follows:*

   - $V(\Gamma') = V(\Gamma) \cup \{u\}$;
   - $E(\Gamma') = (E(\Gamma) - \{e(v_1, v_2)\}) \cup \{e(v_1, u), e(u, v_2)\}$;
   - $l'|_{V(\Gamma)} = l$ *and* $l(v_1) \geq l'(u) \geq l(v_2)$.

   Definition 5.4. *We define an **edge insertion** I_e to be any transformation T of (Γ, l) such that for a fixed vertex $v \in V(\Gamma)$, $T(\Gamma, l)$ is the labeled graph (Γ', l') defined as follows:*

   - $V(\Gamma') = V(\Gamma) \cup \{u\}$;
   - $E(\Gamma') = E(\Gamma) \cup \{e(v, u)\}$;
   - $l'|_{V(\Gamma)} = l$ *and* $l'(u) = l(v)$.

   Definition 5.5. *We define a **loop insertion** I_l to be any transformation T of (Γ, l) such that for a fixed vertex $v \in V(\Gamma)$, $T(\Gamma, l)$ is the labeled graph (Γ', l') defined as follows:*

   - $V(\Gamma') = V(\Gamma)$;
   - $E(\Gamma') = E(\Gamma) \cup \{e(v, v)\}$;
   - $l'|_{V(\Gamma)} = l$.

2. **Delete Operations:**

   Definition 5.6. *We define a **vertex deletion** D_v to be any transformation T of (Γ, l) such that for fixed edges $e(v_1, u), e(u, v_2) \in E(\Gamma)$, with u a vertex of degree 2, and $l(v_1) \geq l(u) \geq l(v_2)$, $T(\Gamma, l)$ is the labeled graph (Γ', l') defined as follows:*

   - $V(\Gamma') = V(\Gamma) - \{u\}$;
   - $E(\Gamma') = (E(\Gamma) - \{e(v_1, u), e(u, v_2)\}) \cup \{e(v_1, v_2)\}$;
   - $l' = l|_{V(\Gamma) - \{u\}}$.

Definition 5.7. *We define an* ***edge deletion*** D_e *to be any transformation* T *of* (Γ, l) *such that for a fixed edge* $e(v,u) \in E(\Gamma)$*, with* u *a vertex of degree 1, and* $l(v) \geq l(u)$*,* $T(\Gamma, l)$ *is the labeled graph* (Γ', l') *defined as follows:*

- $V(\Gamma') = V(\Gamma) - \{u\}$;
- $E(\Gamma') = E(\Gamma) - \{e(v,u)\}$;
- $l' = l|_{V(\Gamma)-\{u\}}$.

Definition 5.8. *We define an* ***loop deletion*** D_l *to be any transformation* T *of* (Γ, l) *such that for a fixed edge* $e(v,v) \in E(\Gamma)$*,* $T(\Gamma, l)$ *is the labeled graph* (Γ', l') *defined as follows:*

- $V(\Gamma') = V(\Gamma)$;
- $E(\Gamma') = E(\Gamma) - \{e(v,v)\}$;
- $l' = l$.

3. **Slide Operation:**

Definition 5.9. *We define an* ***edge sliding*** S_e *to be any transformation* T *of* (Γ, l) *such that, for fixed edges* $e(v_1,v_2), e(v_2,v_3) \in E(\Gamma)$*, with eitehr* $l(v_1) > l(v_2) = l(v_3)$*, or* $l(v_1) < l(v_2) = l(v_3)$*,* $T(\Gamma, l)$ *is the labeled graph* (Γ', l') *defined as follows:*

- $V(\Gamma') = V(\Gamma)$;
- $E(\Gamma') = (E(\Gamma) - \{e(v_1,v_2)\}) \cup \{e(v_1,v_3)\}$;
- $l' = l$.

4. **Relabel Operation:**

Definition 5.10. *We define a* ***relabeling*** R_v *to be any transformation* T *of* (Γ, l) *such that* $T(\Gamma, l)$ *is the labeled graph* (Γ', l') *defined as follows:*

- $V(\Gamma') = V(\Gamma)$;
- $E(\Gamma') = E(\Gamma)$;
- *For every* $u, v \in V(\Gamma)$*, if* $l(u) \leq l(v)$*, then* $l'(u) \leq l'(v)$.

5.2.2 Edit Distance

Since $T(\Gamma, l)$ is another labeled multigraph, we can chain together these elementary deformations to form a sequence of edit operations.

Definition 5.11. *An* ***edit sequence*** *of the labeled graph* (Γ, l) *is any finite ordered sequence* $S = (T_1, T_2, \ldots, T_n)$ *of edit operations such that* T_1 *is an edit operation acting on* (Γ, l)*,* T_2 *is an operation acting on* $T(\Gamma, l)$*, and so on. We denote the result of applying this entire sequence to* (Γ, l) *as* $S(\Gamma, l)$*. Finally, we denote* $\mathcal{S}((\Gamma, l), (\Gamma', l'))$ *to be the set of edit sequences* S *that carry* (Γ, l) *to* (Γ', l').

Definition 5.12. *Let* $S = (T_1, \ldots, T_n) \in \mathcal{S}((\Gamma, l), (\Gamma', l'))$*. Setting* $(\Gamma_1, l_1) := (\Gamma, l)$ *and* $(\Gamma_{n+1}, l_{n+1}) := (\Gamma', l')$*, let* $(\Gamma_{i+1}, l_{i+1}) = T_i(\Gamma_i, l_i)$*, for all* $i \in \{1, \ldots, n\}$*. Then, let* $J_s(v)$ *be the indices* i *such that* $v \in V(\Gamma_i)$*. The* ***cost*** *of* S *is then*

$$c(S) = \max_{v \in \bigcup_i^{n+1} V(\Gamma_i)} \left(\max_{i \in J_s(v)} l_i(v) - \min_{j \in J_s(v)} l_j(v) \right)$$

Definition 5.13. *The* ***edit distance*** δ_e *between any two labeled graphs* (Γ, l) *and* (Γ', l') *is defined to be*

$$\delta_e((\Gamma, l), (\Gamma', l')) = \inf_{\mathcal{S}((\Gamma,l),(\Gamma',l'))} c(S)$$

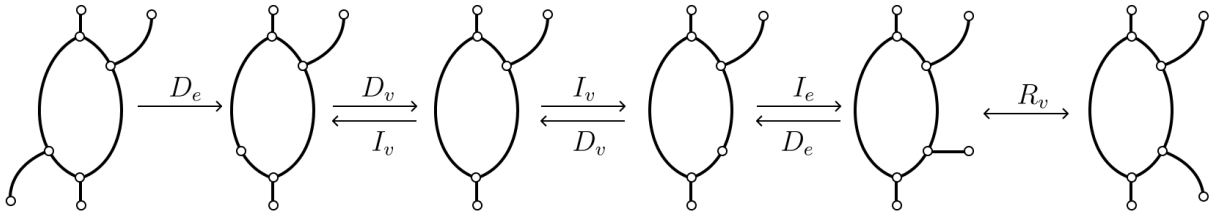
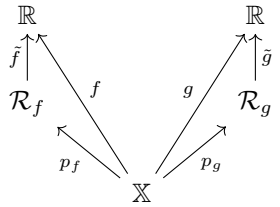


Figure 7: Caption

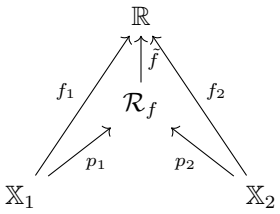
5.3 Generalized Reeb Graph Edit Distance

Similar to the standard graph edit distance, the Reeb graph edit distance captures the amount of "work" to transform one Reeb graph to another by associating a cost to a sequence of transformations. Suppose we have two scalar fields $(\mathbb{X}, f)$ and $(\mathbb{X}, g)$ defined on the same domain $\mathbb{X}$. Then, we can construct two different Reeb graphs, $\mathcal{R}_f, \mathcal{R}_g$, equipped with scalar functions $\tilde{f}$ and $\tilde{g}$. Naturally, there exists quotient maps p_f and p_g such that $f = \tilde{f} \circ p_f$ and $g = \tilde{g} \circ p_g$. In other words, the following diagram commutes:



(5.14)

Now, similarly, we can construct two slightly different topological spaces $\mathbb{X}_1$ and $\mathbb{X}_2$ which, when endowed with different functions f_1 and f_2, can create the same Reeb graph $\mathcal{R}_f$. This creates the following commutative diagram:



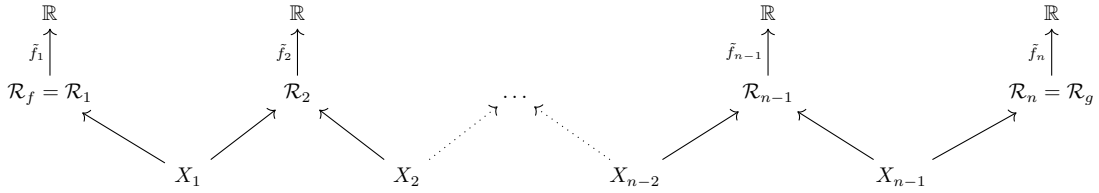
(5.15)

Diagram 5.14 corresponds to a **relabel** operation; changing the values of the vertices of the Reeb graph without changing the underlying topology. Diagram 5.15 corresponds to an **edit** operation; changing the topological space without disturbing the Reeb graph. Examples of relabel operations include increasing/decreasing the function value of specific critical points, while edit operators can come in the form of adding flat plateaus to the topological space.

In what we have just described, we are restricting our Reeb graphs $\mathcal{R}_f$ and $\mathcal{R}_g$ to be Reeb graphs of $\mathbb{X}_1$ and $\mathbb{X}_2$ in the traditional sense. However, if we loosen the restriction on the types of maps from $\mathbb{X}_1, \mathbb{X}_2$ to $\mathcal{R}_f, \mathcal{R}_g$, we allow for more freedom on the choices of $\mathbb{X}_1, \mathbb{X}_2$.

Definition 5.16. *Let $\mathcal{R}_f$ be a Reeb graph, and let $(X, \hat{f})$ be some scalar field. We say that the quotient map $p : X \to \mathcal{R}_f$ is a **Reeb quotient map** if p is has locally connected fibers and if $\hat{f} = \tilde{f} \circ p$. In this case, we say that $\mathcal{R}_f$ is a **Reeb graph of** $\hat{f} : X \to \mathbb{R}$.*

We loosen the restriction on the edit and relabel operations defined in Diagram 5.15 and 5.14 by requiring that $\mathcal{R}_1, \mathcal{R}_2$ be Reeb graphs of X_1, X_2 in the sense of definition 5.16. Then, given a sequence of Reeb graphs $\mathcal{R}_f = \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{n-1}, \mathcal{R}_n = \mathcal{R}_n$, we can choose a sequence of topological spaces $X_1, X_2, \ldots, X_{n-1}$ such that the following zigzag diagram commutes:



(5.17)

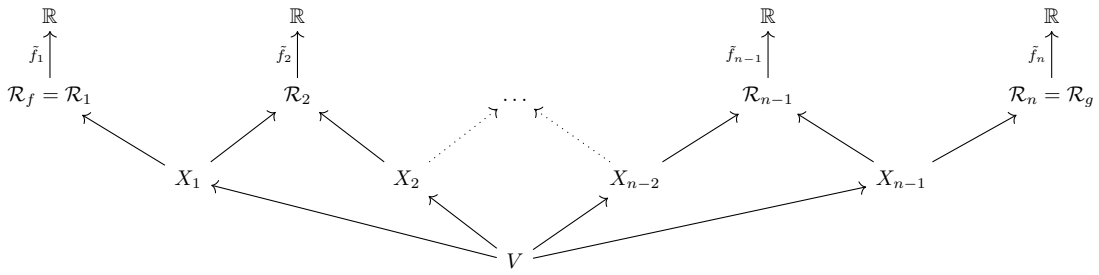
This zigzag diagram can now be considered a sequence of relabel and edit operations which carry the Reeb graph R_f to R_g. We call Diagram 5.17 a **Reeb zigzag diagram**.

Now, rather than think of $X_1, \ldots, X_{n-1}$ as being arbitrary topological spaces, we can think of them as being 1-dimensional topological graphs. That way, the edits to these topological spaces are more in-line with the standard graph edit operations; insertion/deletion of nodes, insertion/deletion of edges, etc. Figure 8 shows an example of a sequence carrying the Reeb graphs originally defined in Figure **??** to one another. We let $p_{i,j}$ denote the Reeb quotient maps from X_i to $\mathcal{R}_j$ and $\hat{f}_{i,j}$ to denote the function induced by the quotient map $p_{i,j}$. That is, $\hat{f}_{i,j} = \tilde{f}_j \circ p_{i,j}$.

We now want to assign a cost to this zigzag diagram. To do this, we introduce the notion of a **Reeb cone**, which will provide us a common space for us to assign a cost function rather than splitting the function across various X_i's.

Definition 5.18. *Let Z be a zigzag diagram such as the one Diagram 5.17 in V be a space such that there exists Reeb quotient maps $V \to X_i$ for all $i \in \{1, \ldots, n-1\}$. By commutativity of Reeb quotient maps, this creates Reeb quotient maps form $V \to \mathcal{R}_i$ for all $i \in \{1, \ldots, n\}$. We call V a **Reeb cone**[7] of the diagram Z.*

Adding the Reeb cone to Diagram 5.17 creates the following commutative diagram:



(5.19)

[7] This definition of Reeb cone makes V a well-defined cone in the category theory sense as well.

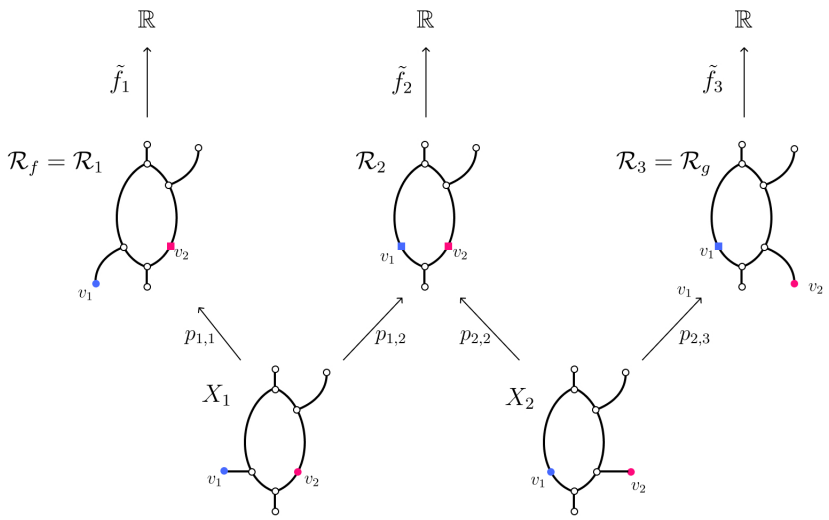


Figure 8: A depiction of a Reeb zigzag diagram which shows how we can transform one Reeb graph into another. The first topological space X_1 has a map to $\mathcal{R}_f = \mathcal{R}_1$ through $p_{1,1}$ by making $\hat{f}_{1,1}$ a height function except on the edge containing v_1, where the function is adjusted to be a downward leaf. X_2 maps into $\mathcal{R}_g = \mathcal{R}_3$ in a similar way. Both X_1 and X_2 map into $\mathcal{R}_2$ via the height function besides the horizontal stems. In fact, $p_{1,2}$ and $p_{2,2}$ are simply the standard Reeb quotient maps in the traditional sense from Definition 2.2

For every $x \in V$, we now have n different scalar values corresponding to the n different functions $\tilde{f}_1, \ldots, \tilde{f}_n$. By commutativity of this diagram, each $\tilde{f}_i$ induces a unique function $f_i : V \to \mathbb{R}$. We define a cost on this zigzag diagram as follows:

Definition 5.20. *Let Z be a Reeb zigzag diagram and V be a Reeb cone of Z. We define the **spread** of V to be the function*

$$s^V : V \to \mathbb{R}, \quad x \mapsto \max_{i=1,\ldots,n} f_i(x) - \min_{i=1,\ldots,n} f_i(x).$$

*Moreover, we define the **cost** of Z to be the supremum of the spread of L, where L is the limit of the diagram Z. That is,*

$$c_Z = ||s^L||_\infty = \sup_{x \in L} \left(\max_{i=1,\ldots,n} f_i(x) - \min_{i=1,\ldots,n} f_i(x) \right)$$

[Brian: How to explain limit without getting into category theory too much?]

While we leave the precise definition of the limit L to the interested reader, essentially it is a cone of Z such that every other cone V of Z can map into L such that the commutativity of the Diagram 5.19 is preserved.

Finally, we can define the edit distance between Reeb graphs $\mathcal{R}_f$ and $\mathcal{R}_g$:

Definition 5.21. *We define the **edit distance** δ_e between $\mathcal{R}_f$ and $\mathcal{R}_g$ to be the infimum cost over all zigzag diagrams carrying $\mathcal{R}_f$ to $\mathcal{R}_g$. That is,*

$$\delta_e(\mathcal{R}_f, \mathcal{R}_g) = \inf_Z c_Z$$

5.4 Properties

6 Examples

[Brian: Designate this section for providing the examples which involve every distance. Think it will be easier to have an entire example with the distances back to back.]

6.1 Example One

Let (X, f), (Y, g) be scalar fields as shown in Figure **??**.

6.1.1 Bottleneck Distance

The extended persistence diagrams for both figures are identical, making the Bottleneck distance equal to 0.

6.1.2 Interleaving Distance

Before determining the interleaving distance between these two scalar fields, we should recognize that these two graphs are graph isomorphic, but not isomorphic as Reeb graphs. By property **[Brian: cite property from interleaving section]**, this implies that the two Reeb graphs do not have a 0-interleaving between them. To see this, consider Figure **??**.

6.1.3 Functional Distortion Distance

6.1.4 Edit Distance

7 Comparison of Distances

Here, we only consider regular and extended persistence diagrams. I assume that we will try and make it so that all the relationships are in terms of extended persistence (which might not be difficult).

Facts:

1. In [26], we are shown that the Reeb graph edit distance is greater than the functional distortion distance.
2. In [26], we are shown that the Reeb graph Edit distance is greater than the bottleneck distance.
3. In [5], we are shown that the Interleaving Distance and Functional Distortion Distance are strongly equivalent:
$$d_I(f, g) \leq d_{FD}(f, g) \leq 3d_I(f, g)$$

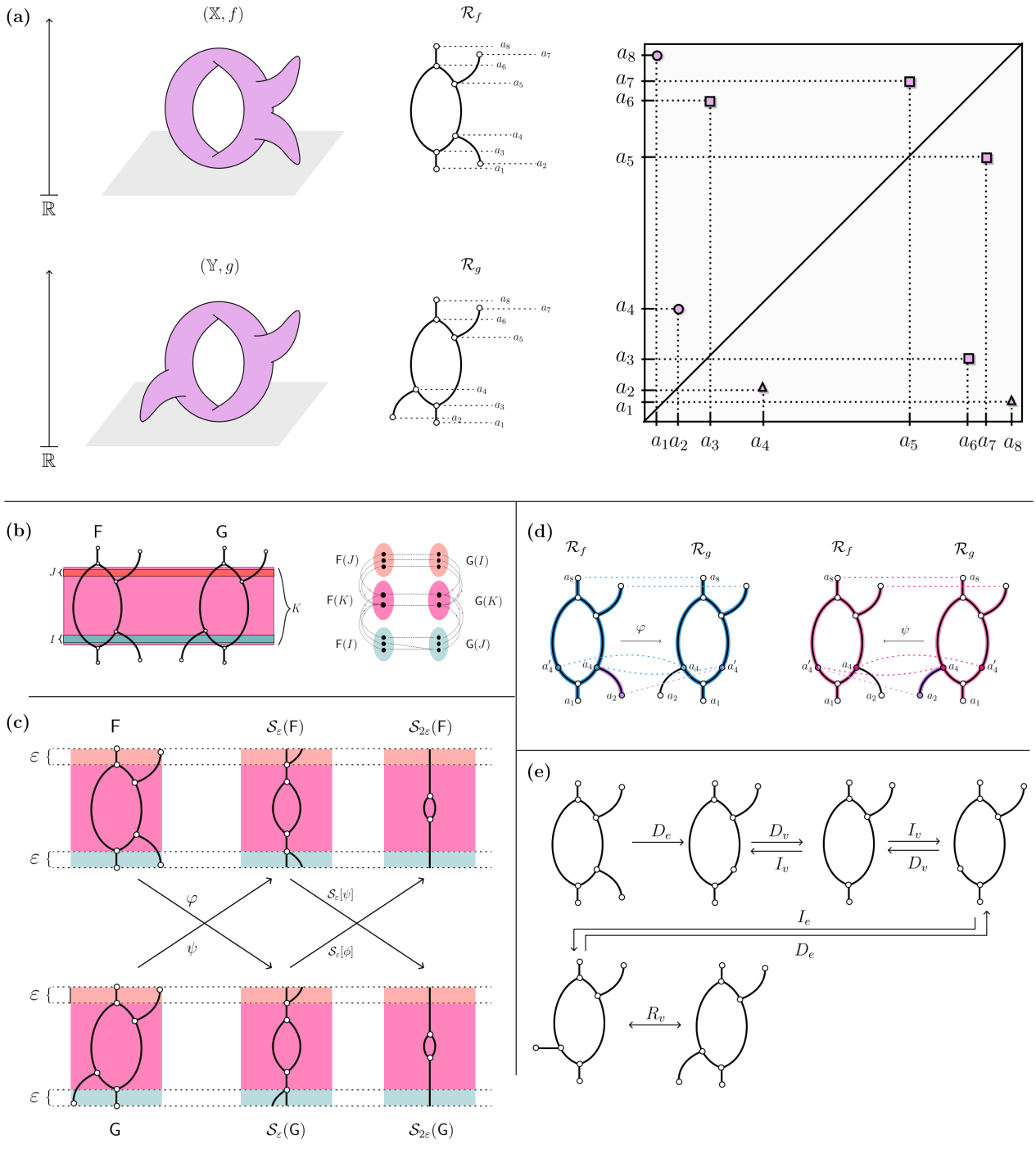


Figure 9: Caption

4. In [5] we are shown the relationship between interleaving distance on Reeb graphs compared to the bottleneck distance on persistence diagrams:

$$d_B(\mathrm{Dg}_0(f), \mathrm{Dg}_0(g)) \leq 3d_I(f,g)$$

$$d_B(\mathrm{ExDg}_1(f), \mathrm{ExDg}_1(g)) \leq 9d_I(f,g)$$

5. In [3] we are shown the relationship between functional distortion distance on Reeb graphs and bottleneck distance on persistence Diagrams:

$$d_B(\mathrm{Dg}_0(f), \mathrm{Dg}_0(g)) \leq d_{FD}(f,g)$$

$$d_B(\mathrm{ExDg}_1(f), \mathrm{ExDg}_1(g)) \leq 3d_{FD}(f,g)$$

8 Computation

8.1 Complexity Results

[Brian: Statement of complexity results for each metric, which short discussion. I believe there is a clear difference in the complexity for interleaving distance and functional distortion distance since FDD has to iterate through all continuous maps with less restrictions.]

8.2 Approximations/Implementations

[Brian: Since to our knowledge there are no current approximations/implementations, this section can be more focused on a discussion for pointing research in the right direction. Some points might be the reducing interleaving distance to simpler cases (covering hole example and sliding leaf example)]

9 Alternatives to Reeb Graphs

9.1 Contour Trees

[Brian: Frechet Distance?]

9.2 Merge Trees

[Brian: Current applications are using merge trees or branch decomposition trees of merge tress]

10 Applications

[Brian: Question – If these metrics can be computed/closely approximated, what applications would they be used for?]

[Brian: Question – What other metrics exist for scalar fields that closely resemble these use cases?]

References

[1] Pankaj Agarwal, Herbert Edelsbrunner, John Harer, and Yusu Wang. "Extreme elevation on a 2-manifold". In: Jan. 2004, pp. 357–365. DOI: 10.1145/997817.997871.

[2] Ulrich Bauer, Barbara Di Fabio, and Claudia Landi. "An Edit Distance for Reeb Graphs". In: *Eurographics Workshop on 3D Object Retrieval*. Ed. by A. Ferreira, A. Giachetti, and D. Giorgi. The Eurographics Association, 2016. ISBN: 978-3-03868-004-8. DOI: 10.2312/3dor.20161084.

[3] Ulrich Bauer, Xiaoyin Ge, and Yusu Wang. "Measuring Distance between Reeb Graphs". In: *Annual Symposium on Computational Geometry - SOCG'14*. ACM Press, 2014. DOI: 10.1145/2582112.2582169.

[4] Ulrich Bauer, Claudia Landi, and Facundo Mémoli. "The Reeb Graph Edit Distance is Universal". In: *Foundations of Computational Mathematics* (Dec. 2020). DOI: 10.1007/s10208-020-09488-3.

[5] Ulrich Bauer, Elizabeth Munch, and Yusu Wang. "Strong Equivalence of the Interleaving and Functional Distortion Metrics for Reeb Graphs". In: *31st International Symposium on Computational Geometry (SoCG 2015)*. Ed. by Lars Arge and János Pach. Vol. 34. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 461–475. ISBN: 978-3-939897-83-5. DOI: http://dx.doi.org/10.4230/LIPIcs.SOCG.2015.461.

[6] Håvard Bakke Bjerkevik and Magnus Bakke Botnan. "Computational Complexity of the Interleaving Distance". In: (Dec. 2017). arXiv: 1712.04281v1 [cs.CG].

[7] Håvard Bakke Bjerkevik, Magnus Bakke Botnan, and Michael Kerber. "Computing the Interleaving Distance is NP-Hard". In: *Foundations of Computational Mathematics* (Nov. 2019). DOI: 10.1007/s10208-019-09442-y.

[8] Andrew J. Blumberg and Michael Lesnick. "Universality of the Homotopy Interleaving Distance". In: (May 2017). arXiv: 1705.01690v1 [math.AT].

[9] Magnus Bakke Botnan, Justin Curry, and Elizabeth Munch. "A Relative Theory of Interleavings". In: (Apr. 2020). arXiv: 2004.14286v1 [math.CT].

[10] Adam Brown, Omer Bobrowski, Elizabeth Munch, and Bei Wang. "Probabilistic Convergence and Stability of Random Mapper Graphs". In: *Accepted to Journal of Applied and Computational Topology* (Sept. 2019). arXiv: 1909.03488v1 [math.AT].

[11] Peter Bubenik and Jonathan A. Scott. "Categorification of Persistent Homology". English. In: *Discrete & Computational Geometry* 51.3 (2014), pp. 600–627. ISSN: 0179-5376. DOI: 10.1007/s00454-014-9573-x.

[12] Peter Bubenik, Vin de Silva, and Jonathan Scott. "Metrics for Generalized Persistence Modules". In: *Foundations of Computational Mathematics* 15.6 (Oct. 2014), pp. 1501–1531. DOI: 10.1007/s10208-014-9229-5.

[13] Peter Bubenik, Vin de Silva, and Jonathan Scott. "Interleaving and Gromov-Hausdorff distance". In: (July 2017). arXiv: 1707.06288v3 [math.CT].

[14] Hamish Carr, Jack Snoeyink, and Ulrike Axen. "Computing contour trees in all dimensions". In: *Computational Geometry* 24.2 (2003). Special Issue on the Fourth {CGC} Workshop on Computational Geometry, pp. 75–94. ISSN: 0925-7721. DOI: http://dx.doi.org/10.1016/S0925-7721(02)00093-7.

[15] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. "Proximity of persistence modules and their diagrams". In: *Proceedings of the 25th annual symposium on Computational geometry*. SCG '09. Aarhus, Denmark: ACM, 2009, pp. 237–246. ISBN: 978-1-60558-501-7. DOI: 10.1145/1542362.1542407.

[16] Frédéric Chazal and Jian Sun. "Gromov–Hausdorff Approximation of Filamentary Structures Using Reeb-Type Graphs". In: *Discrete & Computational Geometry* 53 (June 2014). DOI: 10.1145/2582112.2582129.

[17] Daniel H. Chitwood, Mitchell Eithun, Elizabeth Munch, and Tim Ophelders. "Topological Mapper for 3D Volumetric Images". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2019, pp. 84–95. DOI: 10.1007/978-3-030-20867-7_7.

[18] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. "Extending Persistence Using Poincaré and Lefschetz Duality". In: *Foundations of Computational Mathematics* 9 (Feb. 2009), pp. 79–103. DOI: 10.1007/s10208-008-9027-z.

[19] Joshua Cruz. "Metric Limits in Categories with a Flow". In: (Jan. 2019). arXiv: 1901.04828v1 [math.CT].

[20] Justin Curry. "Sheaves, Cosheaves and Applications". PhD thesis. University of Pennsylvania, 2014.

[21] Justin Curry and Amit Patel. "Classification of Constructible Cosheaves". In: *ArXiv:1603:01587* (2016).

[22] JustinMichael Curry. "Topological data analysis and cosheaves". English. In: *Japan Journal of Industrial and Applied Mathematics* (2015), pp. 1–39. ISSN: 0916-7005. DOI: 10.1007/s13160-015-0173-9.

[23] Vin de Silva, Elizabeth Munch, and Anastasios Stefanou. "Theory of interleavings on categories with a flow". In: *Theory and Applications of Categories* 33.21 (2018), pp. 583–607.

[24] Tamal K Dey, Facundo Mémoli, and Yusu Wang. "Multiscale mapper: topological summarization via codomain covers". In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics. 2016, pp. 997–1013.

[25] Tamal K. Dey, Facundo Memoli, and Yusu Wang. "Topological Analysis of Nerves, Reeb Spaces, Mappers, and Multiscale Mappers". In: *arXiv 1703.07387* (Mar. 2017). arXiv: 1703.07387v1 [cs.CG].

[26] Barbara Di Fabio and Claudia Landi. "The Edit Distance for Reeb Graphs of Surfaces". In: *Discrete & Computational Geometry* 55.2 (Jan. 2016), pp. 423–461. DOI: 10.1007/s00454-016-9758-6.

[27] Paweł Dłotko. "Ball mapper: a shape summary for topological data analysis". In: (Jan. 2019). arXiv: 1901.07410v1 [math.AT].

[28] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. Jan. 2010. ISBN: 978-0-8218-4925-5. DOI: 10.1007/978-3-540-33259-6_7.

[29] Herbert Edelsbrunner, John Harer, Vijay Natarajan, and Valerio Pascucci. "Morse-Smale complexes for piecewise linear 3-manifolds". In: *Proceedings of the nineteenth annual symposium on Computational geometry*. SCG '03. San Diego, California, USA: ACM, 2003, pp. 361–370. ISBN: 1-58113-663-3. DOI: http://doi.acm.org/10.1145/777792.777846.

[30] Herbert Edelsbrunner, John Harer, and Afra Zomorodian. "Hierarchical Morse-Smale Complexes for Piecewise Linear 2-Manifolds". In: *Discrete & Computational Geometry* 30 (1 2003). 10.1007/s00454-003-2926-5, pp. 87–107. ISSN: 0179-5376.

[31] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. "Topological Persistence and Simplification". In: (Jan. 2003).

[32] F. Escolano, Edwin Hancock, and Silvia Biasotti. "Complexity Fusion for Indexing Reeb Digraphs". In: Aug. 2013, pp. 120–127. DOI: 10.1007/978-3-642-40261-6_14.

[33] Elena Farahbakhsh Touli and Yusu Wang. "FPT-Algorithms for Computing Gromov-Hausdorff and Interleaving Distances Between Trees". In: *27th Annual European Symposium on Algorithms (ESA 2019)*. Ed. by Michael A. Bender, Ola Svensson, and Grzegorz Herman. Vol. 144. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 83:1–83:14. ISBN: 978-3-95977-124-5. DOI: 10.4230/LIPIcs.ESA.2019.83.

[34] Christoph Flamm, Ivo L. Hofacker, Peter F. Stadler, and Michael T. Wolfinger. "Barrier Trees of Degenerate Landscapes". In: *Zeitschrift für Physikalische Chemie* 216.2 (Jan. 2002). DOI: 10.1524/zpch.2002.216.2.155.

[35] Ellen Gasparovic, Elizabeth Munch, Steve Oudot, Katharine Turner, Bei Wang, and Yusu Wang. "Intrinsic Interleaving Distance for Merge Trees". In: (July 2019). arXiv: 1908.00063 [cs.CG].

[36] Xiaoyin Ge, Issam Safa, Mikhail Belkin, and Yusu Wang. "Data Skeletonization via Reeb Graphs". In: *Adv. Neural Inform. Process. Syst.* 24 (May 2012).

[37] Mikhael Gromov. "Groups of polynomial growth and expanding maps (with an appendix by Jacques Tits)". en. In: *Publications Mathématiques de l'IHÉS* 53 (1981), pp. 53–78.

[38] Masaki Hilaga, Yoshihisa Shinagawa, Taku Komura, and Tosiyasu Kunii. "Topology matching for fully automatic similarity estimation of 3D shapes". In: Jan. 2001, pp. 203–212. DOI: 10.1145/383259.383282.

[39] Rachel Jeitziner, Mathieu Carrière, Jacques Rougemont, Steve Oudot, Kathryn Hess, and Cathrin Brisken. "Two-Tier Mapper, an unbiased topology-based clustering method for enhanced global gene expression analysis". In: *Bioinformatics* (Feb. 2019). Ed. by Bonnie Berger. DOI: 10.1093/bioinformatics/btz052.

[40] Woojin Kim and Facundo Memoli. "Stable Signatures for Dynamic Metric Spaces via Zigzag Persistent Homology". In: (Dec. 2017). arXiv: 1712.04064v1 [math.AT].

[41] Woojin Kim, Facundo Mémoli, and Anastasios Stefanou. "The metric structure of the formigram interleaving distance". In: (Dec. 2019). arXiv: 1912.04366v1 [math.AT].

[42] Michael Lesnick. "The Theory of the Interleaving Distance on Multidimensional Persistence Modules". English. In: *Foundations of Computational Mathematics* 15.3 (2015), pp. 613–650. ISSN: 1615-3375. DOI: 10.1007/s10208-015-9255-y.

[43] Facundo Mémoli. "Gromov-Hausdorff distances in Euclidean spaces". In: July 2008, pp. 1–8. ISBN: 978-1-4244-2339-2. DOI: 10.1109/CVPRW.2008.4563074.

[44] Dmitriy Morozov, Kenes Beketayev, and Gunther Weber. "Interleaving Distance between Merge Trees". In: *Proceedings of TopoInVis*. 2013.

[45] Elizabeth Munch and Anastasios Stefanou. "The $\ell\infty$-Cophenetic Metric for Phylogenetic Trees As an Interleaving Distance". In: *Association for Women in Mathematics Series*. Springer International Publishing, 2019, pp. 109–127. DOI: 10.1007/978-3-030-11566-1_5.

[46] Elizabeth Munch and Bei Wang. "Convergence between Categorical Representations of Reeb Space and Mapper". In: *32nd International Symposium on Computational Geometry (SoCG 2016)*. Ed. by Sándor Fekete and Anna Lubiw. Vol. 51. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 53:1–53:16. ISBN: 978-3-95977-009-5. DOI: http://dx.doi.org/10.4230/LIPIcs.SoCG.2016.53.

[47] Alejandro Robles, Mustafa Hajij, and Paul Rosen. "The Shape of an Image: A Study of Mapper on Images". In: (Oct. 2017). arXiv: 1710.09008v1 [cs.CV].

[48] Himangshu Saikia, Hans-Peter Seidel, and Tino Weinkauf. "Fast Similarity Search in Scalar Fields using Merging Histograms". In: June 2017, pp. 121–134. ISBN: 978-3-319-44682-0. DOI: 10.1007/978-3-319-44684-4_7.

[49] Himangshu Saikia and T. Weinkauf. "Global Feature Tracking and Similarity Estimation in Time-Dependent Scalar Fields". In: *Computer Graphics Forum* 36 (June 2017), pp. 1–11. DOI: 10.1111/cgf.13163.

[50] Luis Scoccola. "Locally persistent categories andmetric properties of interleaving distances". PhD thesis. Western University, 2020.

[51] Vin de Silva, Elizabeth Munch, and Amit Patel. "Categorified Reeb Graphs". In: *Discrete & Computational Geometry* (2016), pp. 1–53. ISSN: 1432-0444. DOI: 10.1007/s00454-016-9763-9.

[52] Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson. "Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition". In: *Eurographics Symposium on Point-Based Graphics*. 2007.

[53] Robert R. Sokal and F. James Rohlf. "The Comparison of Dendrograms by Objective Methods". In: *Taxon* 11.2 (Feb. 1962), p. 33. DOI: 10.2307/1217208.

[54] Raghavendra Sridharamurthy, Talha Masood, Adhitya Kamakshidasan, and Vijay Natarajan. "Edit Distance between Merge Trees". In: *IEEE Transactions on Visualization and Computer Graphics* PP (Oct. 2018), pp. 1–1. DOI: 10.1109/TVCG.2018.2873612.

[55] Anastasios Stefanou. "Dynamics on Categories and Applications". PhD thesis. University at Albany, State University of New York, 2018.

[56] Anastasios Stefanou. "Tree decomposition of Reeb graphs, parametrized complexity, and applications to phylogenetics". In: *Journal of Applied and Computational Topology* (Feb. 2020). DOI: 10.1007/s41468-020-00051-1.

[57] Zoë Wood, Hugues Hoppe, Mathieu Desbrun, and Peter Schröder. "Removing Excess Topology from Isosurfaces". In: *ACM Trans. Graph.* 23 (Apr. 2004), pp. 190–208. DOI: 10.1145/990002.990007.

[58] Lin Yan, Yusu Wang, Elizabeth Munch, Ellen Gasparovic, and Bei Wang. "A Structural Average of Labeled Merge Trees for Uncertainty Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* (2019), pp. 1–1. DOI: 10.1109/tvcg.2019.2934242. arXiv: 1908.00113.