Realizable Piecewise Linear Paths of Persistence Diagrams with Reeb Graphs

Rehab Alharbi
Department of Mathematics, Science College, Jazan University, Saudi Arabia

Erin W. Chambers
Department of Computer Science and Engineering, University of Notre Dame, USA

Elizabeth Munch
Department of Computational Science, Mathematics, and Engineering; and Department of Mathematics, Michigan State University, USA

Abstract

Reeb graphs are widely used in a range of fields for the purposes of analyzing and comparing complex spaces via a simpler combinatorial object. Further, they are closely related to extended persistence diagrams, which largely but not completely encode the information of the Reeb graph. In this paper, we investigate the effect on the persistence diagram of a particular continuous operation on Reeb graphs; namely the (truncated) smoothing operation. This construction arises in the context of the Reeb graph interleaving distance, but separately from that viewpoint provides a simplification of the Reeb graph which continuously shrinks small loops. We then use this characterization to initiate the study of inverse problems for Reeb graphs using smoothing by showing which paths in persistence diagram space (commonly known as vineyards) can be realized by a path in the space of Reeb graphs via these simple operations. This allows us to solve the inverse problem on a certain family of piecewise linear vineyards when fixing an initial Reeb graph. While this particular application is limited in scope, it suggests future directions to more broadly study the inverse problem on Reeb graphs.

Keywords and phrases Reeb graphs, smoothing, inverse problems, topological data analysis

Digital Object Identifier 10.57717/cgt.v3i1.38

Funding *Erin W. Chambers*: Work funded in part by the National Science Foundation through grants CCF-1907612, CCF-2106672, and DBI-1759807.
Elizabeth Munch: Work funded in part by the National Science Foundation through grants CCF-1907591, CCF-2106578, and CCF-2142713.

Acknowledgements The authors wish to thank Woojin Kim for helpful feedback on an early version of this work.

1 Introduction

Reeb graphs are an important tool in topological data analysis for the purpose of visualizing continuous functions on complex spaces, as they yield a simplified discrete structure. Originally developed in relation to Morse theory [29], these objects are used extensively for shape comparison, constructing skeletons of data sets, surface simplification, and visualization; for more details on these and more applications, we refer to recent surveys on the topic [6, 30]. While some information is lost in the construction of the Reeb graph, such simplified structures allow for more efficient methods to analyze and compare data sets. More precisely, given a topological space M and a real valued function $f : M \to \mathbb{R}$, the pair (M, f) is known as an $\mathbb{R}$-space; e.g. see Figure 1. The Reeb graph of f is then obtained by collapsing each connected component in a level set into single point, and collecting the points together using the quotient topology. The result, with appropriate assumptions on the input, is a 1-dimensional stratified space (i.e. a graph) along with an induced real-valued function; we will refer to this pair as (X, f).

Given the many algorithms available to compute these objects efficiently [20, 28, 22, 21], the Reeb graph is a practical, simplified structure which can be used for tasks ranging from simplification to visualization. Thus, there is a practical need for ways to compare and analyze Reeb graphs. Many possible options have been studied recently [19, 3, 16, 10, 2, 8, 12, 4, 2, 16, 3]; see [5, 30, 7] for recent surveys and discussions on practical implications and theoretical trade-offs. This work focuses on [16, 10] which introduces the concept of smoothing a Reeb graph as a byproduct of the interleaving distance. This smoothing operation generates a new Reeb graph $S_\varepsilon(X, f)$ for every $\varepsilon \geq 0$ which simplifies the topological structure of the graph. In particular, it continuously removes small loops, which are often viewed as noise in the input data.

Despite being defined via category theory, the resulting construction can be viewed from a completely combinatorial viewpoint. An algorithm for constructing the smoothed Reeb graph was provided in [16], and has been extended to the more recently introduced truncated smoothing functor as well [10]. Further, it has been shown that given a Reeb graph (X, f) with critical points $S = \{a_i\}$, the smoothed Reeb graph $S_\varepsilon(X, f)$ has critical set contained in $S_\varepsilon = (S - \varepsilon) \cup (S + \varepsilon) = \{a - \varepsilon, a + \varepsilon \mid a \in S\}$ [16]. However, no exact combinatorial characterization of the graph changes has been proven in the literature prior to this paper.

We note that $\mathbb{R}$-space data can also be studied via its persistent homology, an algebraic method for measuring topological features of shapes and functions [27, 18]. In fact, there is a close relationship between critical values of the Reeb graph and points in the extended persistence diagram [1, 11], which will be a key component of our work. Because of this relationship, one might expect there to be an inverse map; i.e. given an extended persistence diagram, can we reconstruct the Reeb graph uniquely? The answer, in general, is no since many different Reeb graphs can have the same persistence diagram. This, of course, is to be expected as the construction of either the Reeb graph or the persistence diagram from input data can be viewed as a lossy process, where information is consolidated, but certainly not preserved perfectly, from the original data. Nevertheless, these inverse problems are increasingly of interest in topological data analysis as a whole [26, 15], as they can provide insight into how much information is lost in the process of computing such a topological signature, be it a persistence diagram or a Reeb graph.

Note that the idea of how changes in a Reeb graph (such as those induced by a change in the original function used to construct it) affect the persistence diagrams is not new in the literature. For example, prior work compared Reeb graphs using the bottleneck distance

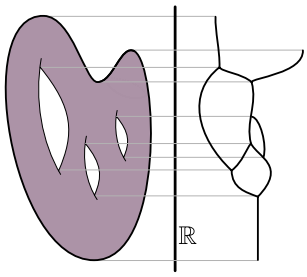


Figure 1 Reeb graph (right) for a given space with real valued function (left) defined by height.

on the corresponding persistence diagram, and was able to show that the comparison was stable under perturbations to the Reeb graph inputs [8, 9]. Further, the exact movement of the points in the extended persistence diagram related to the smoothing operation on Reeb graphs was also studied in [24]. However in that paper, there is more of a focus on the algebraic structure of the persistence module. In contrast, our work focuses on the explicit geometric changes in the Reeb graphs to draw the same conclusion and then utilizes this characterization to study the inverse problem explicitly.

Our contributions: The main contribution of this paper is to initiate the study of inverse problems on vineyards realized by Reeb graphs, using the smoothing and truncated functors to determine when such an inverse can be determined. While this is defined for an admittedly restricted class of paths in persistence diagram space, this work is a first step towards broader notions of approximate inverse paths and controlled modifications of a Reeb graph to realize a broad class of diagram changes. After introducing necessary background and notation in Section 2, in Section 3 we explicitly enumerate all changes to a Reeb graph under Reeb graph smoothing, and then generalize this framework for the more recently developed truncated smoothing operation. In Section 4, we use this characterization to solve the inverse problem in a restricted setting for Reeb graphs using smoothing and truncated smoothing, by showing that certain paths in persistence diagram space (commonly known as vineyards [13]) can be realized by a path in the space of Reeb graphs. Namely, using our characterization of how truncated smoothing affects the combinatorial structure of the Reeb graph, we are able to determine sufficient restrictions on a time varying set of persistence diagrams, so that as long as an initial Reeb graph is specified, we can determine a set of time varying Reeb graphs which realize the vineyard. We conclude in Section 5 by discussing several possible future directions motivated by our work.

2 Background and definitions

2.1 Reeb graphs

Let X be a topological space and $f : X \to \mathbb{R}$ be continuous real-valued function. The pair (X, f) is referred to as an $\mathbb{R}$-space (or a scalar field, in some of the literature). The level set of f at a is the set $f^{-1}(a) = \{x \in X \mid f(x) = a\}$. We define an equivalence relation $\sim_f$ on X by $x \sim y$ if and only if $f(x) = f(y) = a$ and x and y are in the same path connected component of the levelset $f^{-1}(a)$. The Reeb graph $\mathcal{R}(X, f)$ is the quotient space $X/_\sim$, with an induced function inherited from the $\mathbb{R}$-space given by $\overline{f}([x]) = f(x)$. (We will generally abuse notation and call both functions f; we similarly use X instead of (X, f) for brevity.)

For the purposes of our work, we will often divorce the idea of a Reeb graph from the

$\mathbb{R}$-space it came from. In particular, given appropriate assumptions on the input $\mathbb{R}$-space, the Reeb graph is indeed a finite graph, so we will assume that our graphs have this property. We will assume further that all Reeb graphs are *constructible*, defined as follows.

▶ **Definition 1.** *An $\mathbb{R}$-space is* constructible *if it is homeomorphic to one constructed in the following manner. We are given a finite set of critical points $a_1, \ldots, a_n$, and a collection of spaces $\{V_i\}_{i=1}^n$ and $\{E_i\}_{i=1}^{n-1}$. Further, we specify left attaching maps $\ell_i : E_i \to V_i$ for $i = 1, \cdots n-1$ and right attaching maps $r_i : E_i \to V_{i+1}$. We then define X to be the quotient space $\coprod_{i=1}^{n}(V_i \times \{a_i\}) \bigcup \coprod_{i=1}^{n-1}(E_i \times [a_i, a_{i+1}])$ with respect to the identifications $(\ell_i(e), a_i) \sim (e, a_i)$ and $(r_i(e), a_{i+1}) \sim (e, a_{i+1})$. We define the function f to be the projection on the second factor. A constructible $\mathbb{R}$-space is a* Reeb graph *if all the V_i and E_i are discrete, finite sets.*

A Reeb graph can be encoded by the combinatorial data of a finite graph X, and a function defined on the vertices $f : V(X) \to \mathbb{R}$. This can be extended to the edges linearly, and in this case, we require that no adjacent vertices have the same function value. The number of edges incident to v that have higher values of f is called its *up-degree*, and define the term *down-degree* symmetrically. We always assume that a vertex of both up- and down-degree 1 is replaced with the relevant edge; this way, every vertex in the combinatorial representation of the Reeb graph is a critical point. A vertex v is a local maximum (local minimum) if it has up-degree 0 (down-degree 0). Likewise, a vertex is an up-split (down-split) if it has up-degree (down-degree) strictly larger than 1. Note that a vertex can be both an up-split and a down-split; or an up-split and a local minimum, etc., although this does not happen for a generic Morse function on a constructible $\mathbb{R}$-space.

In this paper, we will primarily restrict our attention to Reeb graphs with fairly strong genericity assumptions in order to simplify proofs. Our notion of genericity will involve two distinct criteria. The first violation of genericity is when we have more than one critical point at a given function value; if no such pair exists, we say the graph is *function-generic*. The second possible violation is if the vertex is of a type not seen in the case of Morse functions on 2-manifolds. The four kinds of vertices which can appear in this case, with their (down-, up-) degrees specified, are local minima (0,1); up-forks (1,2); down-forks (2,1); and local maxima (1,0). If a Reeb graph has only vertices of these four types, it is called *Morse-generic*. If a Reeb graph is both function- and Morse-generic, we simply call it *generic*.

In the case of Morse-generic Reeb graphs, we have a strong characterizations of the critical points. We say a down-fork v is an *ordinary down fork* if the two lower branches of v are contained in different connected components of the open sublevel set $\mathcal{R}(G)_{<a} := f^{-1}(-\infty, a)$. Otherwise, we say v is an *essential down fork*. The ordinary and essential up-forks are defined in the same way, using the open super-level set $\mathcal{R}(G)_{>a} := f^{-1}(a, \infty)$.

2.2 Smoothing and truncated smoothing

We now turn our attention to the (geometric) definition of smoothing given in [16], and the truncated smoothing given in [10]. Both smoothing and truncated smoothing were studied in the context of comparing two Reeb graphs, where the focus was on defining distances between the graphs. Both allow for the definition of an interleaving distance with desirable theoretical properties [16, 10]. While these distances partially motivate our study, we do not directly use them, but rather focus on the two operations themselves.

Let (X, f) be a Reeb graph and let $\varepsilon \geq 0$. Define $(f + \mathrm{Id}) : X \times [-\varepsilon, \varepsilon] \to \mathbb{R}$ by $(x, t) \mapsto f(x) + t$. We define the *ε-smoothing* $S_\varepsilon(X, f)$ to be the Reeb graph of $(X \times [-\varepsilon, \varepsilon], f + \mathrm{Id})$; and denote the corresponding quotient map by $q : X \times [-\varepsilon, \varepsilon] \to S_\varepsilon(X, f)$.

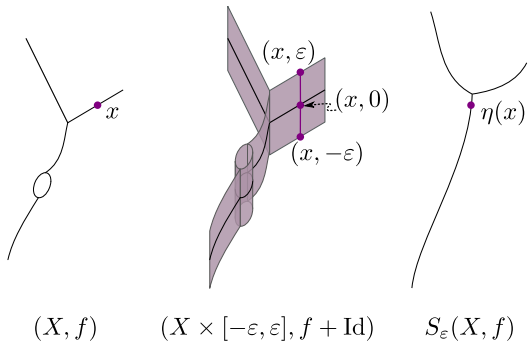
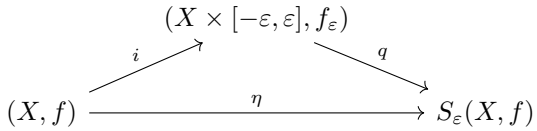


Figure 2 An example of the thickening and smoothing procedure. Given a Reeb graph (left) with function implied by height, we cross the graph with an interval $[-\varepsilon,\varepsilon]$, drawn so that the induced function f_ε is still visualized by height. Then the smoothed Reeb graph $S_\varepsilon(X,f)$ is the Reeb graph of this $\mathbb{R}$-space.

We again slightly abuse notation and refer to $f+\mathrm{Id}$ as f_ε and $S_\varepsilon(X,f)$ as $S_\epsilon(X)$ for brevity. Further, if $i : X \to X\times[-\varepsilon,\varepsilon]$ is given by $i(x) = (x,0)$, we denote the induced map $\eta := q\circ i : X \to S_\varepsilon(X,f)$. That is, η is defined so that the diagram



commutes. See Fig. 2 for an example.

It is worth noting that smoothing is a functor, with further structure that we will not utilize in this paper; see [16, 17] for details. We will focus on the combinatorial properties of smoothing and the η map in our work, and will characterize how a Reeb graph changes under η as ε varies. We begin with the following lemma that shows η does not change the connected components; this result is implicitly referenced in prior work [16] but never explicitly proven, so we include a proof for completeness.

▶ **Lemma 2.** *The induced map $\pi_0[\eta]$ gives an isomorphism $\pi_0(X) \simeq \pi_0(S_\varepsilon(X,f))$.*

Proof. Note that because $\eta = q\circ i$, functoriality says that we need only show that $\pi_0[q]$ and $\pi_0[i]$ are isomorphisms as $\pi_0[\eta] = \pi_0[q]\circ\pi_0[i]$.

From Theorem 23.6 of Munkres [25], a finite Cartesian product of connected spaces is connected. So for each connected component $A \subseteq G$, $A\times[-\varepsilon,\varepsilon]$ is connected, and thus $\pi_0[i] : \pi_0(X,f) \to \pi_0((X)\times[-\varepsilon,\varepsilon])$ is an isomorphism.

The map $\pi_0[q] : \pi_0(X\times[-\varepsilon,\varepsilon]) \to \pi_0(S_\varepsilon(X,f))$ is surjective because by definition of a quotient map, q is surjective. We next show $\pi_0[q] : \pi_0(X\times[-\varepsilon,\varepsilon]) \to \pi_0(S_\varepsilon(X,f))$ is injective. By Munkres Exercise 23.11 [25], since our quotient map q has $q^{-1}(y)$ connected for every $y \in S_\varepsilon(X)$, then for each connected component A of $S_\varepsilon(X,f)$, $q^{-1}(A)$ is connected. This implies that two connected components of $X\times[-\varepsilon,\varepsilon]$ cannot map to the same connected component of $S_\varepsilon(X,f)$ under q without a contradiction. Thus $\pi_0[q]$ is injective. ◀

Truncated smoothing is a more recently developed variation of the smoothing functor, which smooths the input graph and then "chops off" tails in the smoothed graph [10]. More formally, define a path in the Reeb graph (X,f) to be a map $\gamma : [0,1] \to X$. This path is monotone increasing (resp. decreasing) if $f(\gamma(t)) \le f(\gamma(t'))$ (resp. $f(\gamma(t)) \ge f(\gamma(t'))$) for

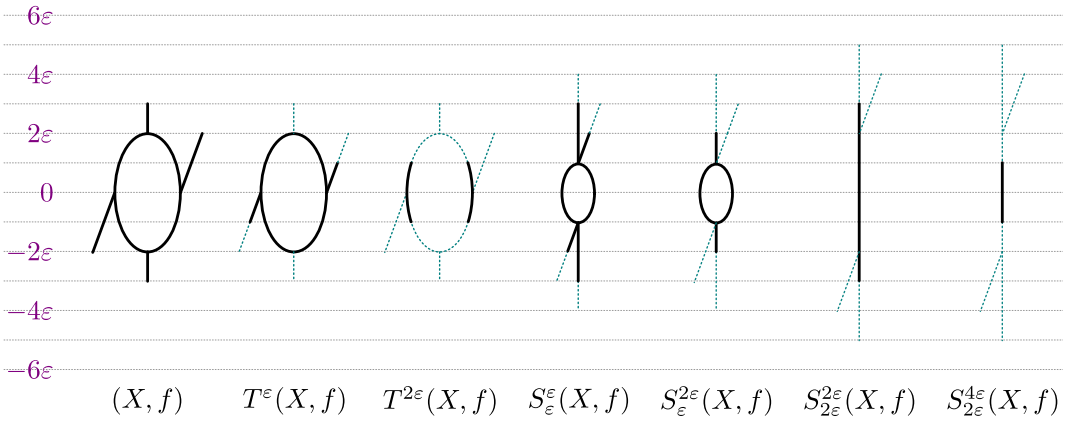


Figure 3 Examples of smoothing and truncating a given (left most) Reeb graph. The sets $U_\tau(X,f)$ and $D_\tau(X,f)$ from $S_\varepsilon(X,f)$ are indicated as dotted lines in the graphs to the right, which are removed in the truncated graph. The notation $S^\tau_\varepsilon = T^\tau S_\varepsilon$ denotes truncation after smoothing. Notice that different choices of ε and τ can result in drastically different topology in $S^\tau_\varepsilon(X)$.

all $t \leq t'$. We also call these up- and down-paths, respectively. The height of a monotone path γ is $|f(\gamma(0)) - f(\gamma(1))|$. Let $U_\tau(X,f)$ be the set of points of X that do not have a height τ up-path and let D_τ be the set of points of X that do not have a height τ down-path. The truncation of Reeb graph (X,f) is defined by $T^\tau(X,f) = (X,f) \setminus (U_\tau \cup D_\tau)$ so that we keep only the subgraph of (X,f) that consists of the points that have both up-path and down-path of height τ. For a choice of ε and τ, the truncated smoothing of Reeb graph (X,f) is defined to be $S^\tau_\varepsilon(X,f) = T^\tau S_\varepsilon(X,f)$. See Fig. 3 for examples.

The truncated smoothing operation inherits many of the useful properties of regular smoothing, at least for some values of τ relative to ε. First, S^τ_ε is a functor and we have a map $\eta : (X,f) \to S^\tau_\varepsilon(X,f)$ for any $0 \leq \tau \leq \varepsilon$. Note that we abuse notation and write η since this map is a restriction of the map $\eta : (X,f) \to S_\varepsilon(X,f)$. We note also that $\pi_0(S_\varepsilon(X,f)) \simeq \pi_0(S^\tau_\varepsilon(X,f))$, since it is shown in [10] that if $0 \leq \tau \leq 2\varepsilon$ and (X,f) is connected then $S^\tau_\varepsilon(X,f)$ is connected. If τ is outside of these ranges or if we truncate without smoothing, it is possible to change the topology of the resulting graph; see Figure 3.

2.3 Persistent Homology

In this section, we give a brief introduction to extended persistent homology. We assume that the reader is familiar with standard homology, relative homology, and persistent homology and direct the reader to [25, 23, 27] for further details.

Given a sequence of real values $a_1 \leq a_2 \leq \cdots \leq a_n$, a *persistence module* is a sequence of vector spaces and linear transformations of the form $\mathbb{V} := \{V_{a_1} \to V_{a_2} \to \cdots \to V_{a_n}\}$. Given an interval $[b,d) \subset \mathbb{R}$ with $b,d \in \{a_i\}$, an interval module $\mathbb{I}_{[b,d)} = \{I_{a_1} \to \cdots \to I_{a_n}\}$ is a persistence module where $I_a = 0$ is the trivial vector space for $a \notin [b,d)$ and $I_a = \mathsf{k}$ is the 1-dimensional vector space otherwise; and the linear maps are isomorphisms when possible and 0 otherwise. By [31] (see also [27, Thm. 1.9]), any pointwise finite dimensional persistence module can be written as a direct sum of interval modules $\mathbb{V} = \bigoplus_{(b,d)\in\mathcal{B}} \mathbb{I}_{[b,d)}$ and this representation is unique up to isomorphism.

The structure of a Reeb graph is encoded in the *extended persistence diagram* [1, 12], defined as follows. We extend our ordinary persistence module by using the relative homology of the super level sets. Write $X_a = f^{-1}(-\infty, a]$ and $X^a = f^{-1}[a, \infty)$ for the sub- and

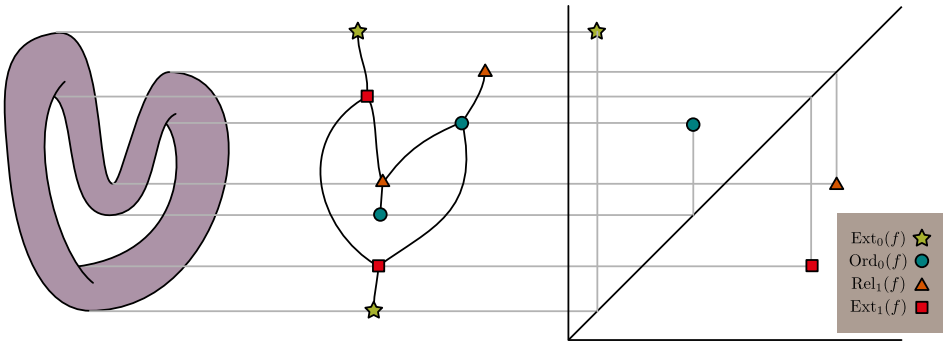


Figure 4 An example Reeb graph, its extended persistence diagram, and vertex pairing.

super-level sets respectively. Note that $X = X_{a_n}$; X^{a_n} is a discrete set consisting of the points with function value at the global maxima; and $H_d(X, X^{a_1}) = 0$. Further, $H_d(X_{a_n}) = H_d(X) = H_d(X, \emptyset)$, so we have a map $H_d(X) \to H_d(X, X^{a_n})$. Thus, we build the extended persistence module

$$0 \to H_d(X_{a_1}) \to \cdots \to H_d(X_{a_n}) \to H_d(X, X^{a_n}) \to \cdots \to H_d(X, X^{a_1}) = 0.$$

We then decompose the resulting persistence module into interval modules. Unlike standard persistence, we associate an interval to the indices a_i and a_j for where the interval starts and ends, while also encoding whether these endpoints happen in the first or second half of the persistence module by putting them in a different sub-diagram. In a Reeb graph, the resulting *extended persistence diagram* can be decomposed into four sub-diagrams [1].

Intervals with both endpoints in the first half of the module are represented by *ordinary persistence points* since they correspond to finite-lifetime points which would show up in the traditional persistence diagram. In the case of a Reeb graph, these only appear in dimension 0, and so we include a point (a_i, a_j) with $i \leq j$ in the ordinary sub-diagram, Ord_0. The second type of point, *relative persistence points*, come from bars in the persistence module which are contained in the second half of the diagram. In Reeb graphs, these only occur in dimension 1. We represent an interval lasting from $H_1(X, X^{a_j}) \to H_1(X, X^{a_i})$ at (a_j, a_i) in the relative sub-diagram, Rel_1, noting that $j \geq i$ so these points are always below the diagonal. The last kind of intervals that appear in the persistence module are those that begin at $H_d(X_{a_i})$ and end at $H_d(X, X^{a_j})$; we then include a point in the extended sub diagram Ext_d at (a_i, a_j). In this case, we can have either $i \leq j$ or $j \leq i$ so points can be both above and below the diagonal in the diagram. In the case of a Reeb graph, these kind of intervals can appear in both dimensions 0 and 1, so we have Ext_0 with $i \leq j$ and Ext_1 with $i \geq j$.

For a Morse-generic Reeb graph, we have a complete pairing of the vertices with respect to the extended persistence diagram. Each point in the Ext_0 diagram corresponds to a connected component of the Reeb graph, born at the global minimum value and dying at the global max value. Each point in the Ext_1 diagram corresponds to a loop in the Reeb graph, born at the highest function value vertex in the loop, and dying at the lowest function value vertex. Note that these are always essential up- and down-forks. Points in Ord_0 and Rel_1 correspond to pairs of vertices: local minima with ordinary down forks in the first case and local maxima with ordinary up forks in the second case. See Figure 4 for an example.

If we lose the Morse-generic assumption, this pairing is a bit more subtle. It is possible to have a vertex in the Reeb graph which contributes to more than one type of persistence point.

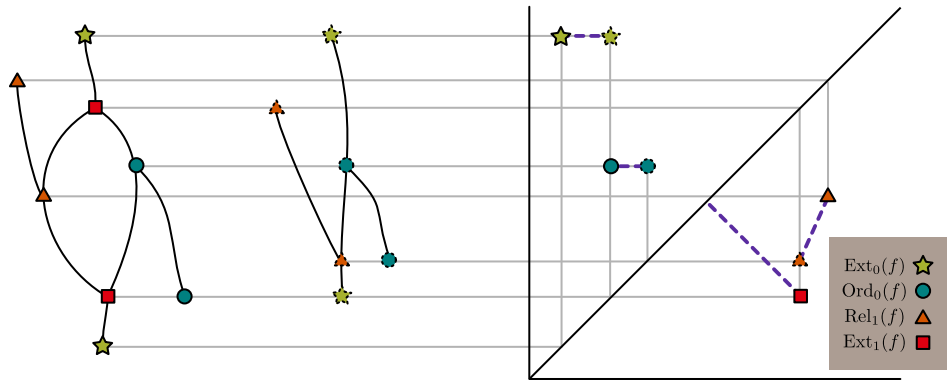


Figure 5 A visualization of the bottleneck distance between the extended persistence diagrams of two Reeb graphs. Note that the pairing from the bottleneck distance must match points of the same type, so even though the solid square and dashed triangle at the bottom right of the diagram are close, the bottleneck matching cannot pair them together.

For example, an up-split with more than two up-edges will contribute to more than one persistence point. For this reason we will state our main theorem in the context of generic Reeb graphs, although with some bookkeeping it can be modified to the case of non-generic Reeb graphs.

2.4 Bottleneck distance

The bottleneck distance first arose in the context of persistent homology, where it was used to assess stability in persistence diagrams [11]. Intuitively, this distance between diagrams considers all pairings of points of the same dimension and type in the two diagrams and calculates the maximum distance among all matched pairs under the L_∞ norm. Note that points are also allowed to match to the diagonal to compensate for the potential of having a different number of off-diagonal points in the diagrams under consideration. In a sense, we can think of this as overlaying the extended persistence diagrams of both graphs and then matching points either to a point of the same type, or to the diagonal; see Fig. 5.

▶ **Definition 3.** *Let $(X, f), (Y, g)$ be two Reeb graphs. We define the* ***bottleneck distance*** *d_b between their extended persistence diagrams $D(X, f)$ and $D(Y, g)$ as*

$$d_b(\mathrm{D}(X, f), \mathrm{D}(Y, g)) = \inf_m \sup_{x \in \mathrm{D}(X,f)} ||x - m(x)||_\infty,$$

where m is a bijection between the multiset of points of $\mathrm{D}(X, f)$ and $\mathrm{D}(Y, g)$ (including points on the diagonal), and the bijection must match points of the same type ($Ord_0(f)$ to $Ord_0(g)$, $Rel_1(f)$ to $Rel_1(g)$, etc.)

3 Geometric analysis of smoothing

In this section, we analyze the smoothing operation more carefully and precisely characterize the behavior of the critical set during the smoothing based on which part of the extended persistence diagram it contributes to. Geometrically, this implies that smoothing eliminates cycles whose height is less than 2ε, and that smoothing eventually results in the graph becoming a forest. The proof of this theorem depends upon a more detailed geometric

classification of how smoothing affects down- and up-forks. We first need the following lemma, proven in [16], which specifies properties of how the projection map works during smoothing.

▶ **Lemma 4.** *[16, Lemma 4.22] Let $p : X \times [-\varepsilon, \varepsilon] \to X$ be the projection map onto the first factor. Then the map p restricts to a homotopy equivalence $f_\varepsilon^{-1}(I) \simeq f^{-1}(I^\varepsilon)$ where I^ε denotes the ε-thickening of the interval I.*

This lemma allows us to completely characterize vertices in the smoothed Reeb graph by looking at inverse images from our original Reeb graph rather than needing to work with the thickened version, $X \times [-\varepsilon, \varepsilon]$. In particular, we can investigate $S_\varepsilon(X, f)$ at function value b to characterize when there is a vertex at that value, and what its up- and down-degrees are.

For the sake of notation, denote by $[b]^\varepsilon = [b - \varepsilon, b + \varepsilon]$ and $(b)^\varepsilon = (b - \varepsilon, b + \varepsilon)$ the closed and open intervals of width 2ε, respectively. Taking the limit of Lem. 4 over intervals $(b)^\delta$ as $\delta \to 0$, we have that $\pi_0 f_\varepsilon^{-1}(b) \simeq \pi_0 f^{-1}([b]^\varepsilon)$; this gives the points in $S_\varepsilon(X, f)$ at the function value b. Because (X, f) is constructible, for a small enough $\delta > 0$ the points in $S_\varepsilon(X, f)$ immediately below are elements of $\pi_0 f^{-1}([b - \delta]^\varepsilon)$, while the points immediately above are elements of $\pi_0 f^{-1}([b + \delta]^\varepsilon)$.

To determine which elements of these sets correspond to a vertex (i.e. those for which up and down degree is not both 1), we can keep track of the attaching maps as follows. First, note that by assumption, $[b]^\varepsilon$ and $[b]^{\varepsilon+\delta}$ intersect the same collection of critical values of (X, f), thus by constructibility, the map induced by inclusion $\pi_0 f^{-1}([b]^\varepsilon) \to \pi_0 f^{-1}([b]^{\varepsilon+\delta})$ is an isomorphism. The point of passing to this larger interval is that now $[b - \delta]^\varepsilon$ and $[b + \delta]^\varepsilon$ are both contained in it, so we have the diagram

$$\begin{array}{ccccc} & & \pi_0 f^{-1}[b]^{\varepsilon+\delta} & & \\ & \overset{\alpha}{\nearrow} & \uparrow & \overset{\beta}{\nwarrow} & \\ \pi_0 f^{-1}[b-\delta]^\varepsilon & & \simeq & & \pi_0 f^{-1}[b+\delta]^\varepsilon \\ & & \pi_0 f^{-1}[b]^\varepsilon. & & \end{array} \tag{1}$$

In $S_\varepsilon(X, f)$, the number of points at b is in the bottom set of Eqn. (1), the lower edges at left, and the upper edges at right. So α and β give attaching information for the lower and upper edges, respectively. Further, there is a vertex at function value b if α or β (or both) are not isomorphisms. See Fig. 6 for an example of these interval representations. We will use this setup repeatedly in the next section to characterize vertices and attaching information to understand how the vertices move in the smoothed version of the Reeb graph.

3.1 Smoothing Reeb Graphs

Next, we consider how to completely determine the combinatorial structure of $S_\varepsilon(X, f)$. In fact, prior work considered this: [16, Corollary 4.25] claims that the set of critical points of $S_\varepsilon(X, f) = \{a \pm \varepsilon \mid a \in S\}$, where S is the set of critical points of the Reeb graph (X, f). In fact, there is some nuance to this issue, as $S_\varepsilon(X, f)$ is a subset of $\{a \in S \mid a \pm \varepsilon\}$, but (assuming we reduce vertices of degree 2 and keep only vertices of degree 1 or of degree 3 or more as critical values) it is never equal to the set $\{a \in S \mid a \pm \varepsilon\}$.

In this paper, we work exclusively with generic Reeb graphs (X, f) and ε values which keep the graph $S_\varepsilon(X, f)$ generic as well. However, even with a generic input Reeb graph, particular choices of ε can result in non-generic $S_\varepsilon(X, f)$, as smoothing intuitively will move

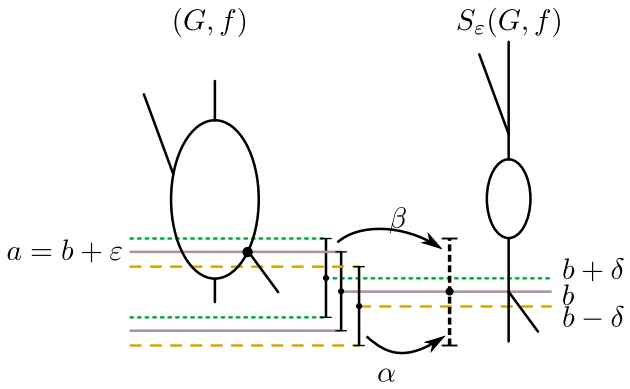


Figure 6 The intervals $[b]^\varepsilon$, $[b-\delta]^\varepsilon$, and $[b+\delta]^\varepsilon$ as well as the dashed interval $[b]^{\varepsilon+\delta}$ shown in between a Reeb graph (left) and its smoothing (right); the inverse images in (X, f) are indicated by dotted lines, and attaching maps α and β as given in Eqn. (1) are the induced maps on the connected components between the noted intervals.

vertices to new function values where they have the potential to "bump into" other vertices. Our proof can be adapted to work for non-generic Reeb graphs and the attaching information described in the last section still completely determines the combinatorial changes; however, we limit our proof to only generic graphs in order to simplify the case analysis. For a generic Reeb graph (X, f), we completely characterize the critical set of $S_\varepsilon(X, f)$ as follows:

▶ **Theorem 5.** *Let (X, f) be a generic Reeb graph with vertex set $V(X) = \{v_1, v_2, \ldots, v_n\}$. We denote the critical set $S = \{a_1 < a_2 < \ldots < a_n\}$ and assume that the vertices are sorted so that $f(v_i) = a_i$. Let $\varepsilon > 0$ be a value such that $2\varepsilon \neq |a_i - a_j|$ for any i, j. Let $W \subseteq V(X)$ be the subset of vertices where each $w \in W$ contributes to a point in Ext_1 which has lifetime at most 2ε. Then there is a bijection $\Phi : V(X) \setminus W \to V(S_\varepsilon(X))$, and $S_\varepsilon(X, f)$ is generic.*

Proof. We will explicitly construct the map Φ, and show it is a bijection. Given a vertex v with $f(v) = a$, we will find a vertex in $S_\varepsilon(X)$ with either function value $a + \varepsilon$ or $a - \varepsilon$ depending on the type of v, and show that this vertex set matching is unique.

First, assume v is a local minimum. Building the diagram of Eq. (1) with $b = a - \varepsilon$ with corresponding maps α and β, we see that the connected component of v in $\pi_0 f^{-1}[b]^{\varepsilon+\delta}$ is not in the image of α. Thus, there is a vertex w in $S_\varepsilon(X, f)$ with $f_\varepsilon(w) = b$, and we define $\Phi(v) = w$. Further, the above construction gives us that the down-degree of w is zero. Noting that δ was chosen sufficiently small in Eq. (1) so that there is no additional vertex between $b + \varepsilon$ and $b + \varepsilon + \delta$, we also have that β is an isomorphism. Therefore, the up-degree of w is 1, implying that w is itself a local minimum vertex. We can use the symmetric argument with $b = a + \varepsilon$ in the case where v is a local maximum to find that its connected component is not in the image of β, and thus there is a local maximum vertex w at height $a + \varepsilon$ in $S_\varepsilon(X, f)$. We again define $\Phi(v) = w$.

Next, assume that v is a down fork which is not part of an Ext_1 pair with lifetime less than 2ε; i.e. $v \notin W$. In this case, we start by building Eqn. (1) with $b = a - \varepsilon$. Consider the two lower edges of v, and the points x and y on these edges at height $a - \delta$. We first show that x and y are in different connected components of $\pi_0 f^{-1}([b - \delta]^\varepsilon)$. If they are in the same connected component, there is a path in $f^{-1}([b - \delta]^\varepsilon)$ connecting the two points, let u be the vertex with lowest function value on this path. By choice of δ, we know that there is no vertex in the interval $[b - \varepsilon - \delta, b - \varepsilon)$, so the height difference between v and u is

at most 2ε. However, in this case, extended persistence would pair the vertex u with v as an Ext_1 pair which has lifetime at most 2ε. This means $v \in W$, contradicting our assumption.

As we now know that x and y are in two different connected components of $f^{-1}([b-\delta]^\varepsilon)$, there are two elements of $\pi_0 f^{-1}([b-\delta]^\varepsilon)$ whose image under α is the component containing v. Because there are no vertices of (X, f) in $f^{-1}((b+\varepsilon, b+\delta+\varepsilon])$ by our choice of δ, there is a single component in $\pi_0 f^{-1}([b+\delta]^\varepsilon)$ mapping to the component of v under β. Thus, we have a vertex w in $S_\varepsilon(X, f)$ at height b with down degree 2 and up degree 1. We set $\Phi(v) = w$. See Fig. 6 for an example. By symmetry, we can use this same argument with $b = a + \varepsilon$ for an up fork vertex, so that for any up fork v in (X, f), there is an up fork w in $S_\varepsilon(X, f)$ at height $a + \varepsilon$, and set $\Phi(v) = w$.

Now that Φ has been defined for all required vertices, we show that $\Phi : V \setminus W \to V(S_\varepsilon(X))$ is a bijection. For surjectivity, let u be a vertex in $S_\varepsilon(X)$ at height b. By assumption on our Reeb graphs and our case analysis of above, u must have at least one of the up or down degree not equal to 1 (i.e. 0 or ≥ 2). Assume first that the down degree is 0 and again construct the diagram of Eqn. (1). Because of vertex u, we must have that there is a connected component $[v] \in \pi_0 f^{-1}([b]^\varepsilon)$ which is not in the image of α; but this is exactly the requirement for having a local minimum vertex at height $b + \varepsilon$ in (X, f), and so $\Phi(v) = u$. The symmetric argument can be made for local maxima. Similarly, assume u is a vertex in $S_\varepsilon(G)$ at height b which has down degree $k \geq 2$. Then there are at least two connected components in $f^{-1}[b-\delta]^\varepsilon$ whose image under α is the component represented by u, and thus there is a vertex in (X, f) at height $b + \varepsilon$ with $\Phi(v) = u$. Again, a symmetric argument can be used in the case of up-degree at least 2; thus Φ is surjective.

For injectivity, recall that the initial Reeb graph is assumed generic, meaning that we cannot have two vertices at the same height. If there was a vertex $u \in S_\varepsilon(X, f)$ at height b and two vertices $v, w \in (X, f)$ with $\Phi(v) = \Phi(w) = u$, then one must be at height $b + \varepsilon$ and the other at height $b - \varepsilon$; without loss of generality assume these are v and w respectively. But then our value $\varepsilon = 2|a_i - a_j|$, where $f(v) = a_i$ and $f(w) = a_j$ are critical values, contradicting our assumptions on ε. Thus Φ is a bijection, and $S_\varepsilon(X, f)$ is generic. ◀

We next show that Φ does not affect the pairing of critical points in the Reeb graph.

▶ **Theorem 6.** *Given a generic Reeb graph, for vertices $u, v \in V(X, f) \setminus W$ which are paired under extended persistence, $\Phi(u)$ and $\Phi(v)$ are paired and of the same type in the extended persistence diagram of $S_\varepsilon(X, f)$.*

Proof. We break this into cases based on the four types of paired points in the extended persistence diagram. In each case, we assume we have paired vertices of (X, f), u and v. For notation, we assume $f(u) = a_i$, $f(v) = a_j$, and $a_i \leq a_j$.

From prior work describing extended persistence of Reeb graphs [1, 12, 14] we have a simple pairing for all the vertices. Namely, the global maximum in any component is paired with its corresponding global minimum (Ext_0 points). A down fork is paired with the highest up fork that spans a loop with it in the graph, if one exists (Ext_1 points). Any remaining downfork is paired with the higher of the two minima for the two components in its sublevel set (Ord_0 points); similarly any remaining upfork is paired with the lower of the two maxima of the two components of its superlevel set (Rel_1 points).

If the pair is in Ext_0, we have a point (a_i, a_j) in the diagram. Further, u and v are the global minimum and maximum respectively of a connected component of X. From Theorem 5, we know that $S_\varepsilon(X, f)$ has vertices $\Phi(u)$ and $\Phi(v)$ at height $a_i - \varepsilon$ and $a_j + \varepsilon$. Moreover, any other critical point b in this component of the original graph has value $a_i < b < a_j$, so any critical point in the related component of $S_\varepsilon(X, f)$ is between $a_i - \varepsilon$ and $a_j + \varepsilon$. This implies

$\Phi(u)$ and $\Phi(v)$ are the global maximum and minimum of a component of the smoothed Reeb graph, and hence they are paired by extended persistence in Ext_0.

If the pair is in Ord_0, we have a point (a_i, a_j) in the diagram, where u is a local minimum, and v is an ordinary down fork. Let C_1 and C_2 be the connected components of $f^{-1}(-\infty, a_j)$ below v. Let u and u' be the minimum function value vertices of these two connected components respectively. Since v is paired with u, this implies that $f(u') < f(u)$. After smoothing (X, f) by ε, by Theorem 5 we know that $S_\varepsilon(X, f)$ has local minimum $\Phi(u)$ at the critical value $a_i - \varepsilon$, local minimum $\Phi(u')$ at $f(u') - \varepsilon$, and downfork $\Phi(v)$ at $a_j - \varepsilon$. Consider the connected component(s) below $\Phi(v)$ in $\tilde{f}^{-1}(-\infty, a_j - \varepsilon)$. Note that by construction, the Reeb quotient map of the thickened space $X \times [-\varepsilon, \varepsilon]$ maintains connected components, so $\pi_0[q] : \pi_0(q^{-1}(\tilde{f}^{-1}(-\infty, a_j - \varepsilon) \simeq \pi_0 \tilde{f}^{-1}(-\infty, a_j - \varepsilon)$ gives an isomorphism.

The sets $C_1 \times \{-\varepsilon\}$ and $C_2 \times \{-\varepsilon\}$ must be disconnected in $q^{-1}(\tilde{f}^{-1}(-\infty, a_j - \varepsilon) \subset X \times [-\varepsilon, \varepsilon]$ because C_1 and C_2 are disconnected in X. Thus, by the isomorphism $\pi_0[q]$, they must map to different connected components below $\Phi(v)$. First, this implies that $\Phi(v)$ must be an essential downfork. Second, we must have $\Phi(u)$ and $\Phi(u')$ in these connected components, and they must be the minimum function value vertices of each of the connected components. We know that $f(\Phi(u)) \geq f(\Phi(u'))$, so $\Phi(u)$ is paired with $\Phi(v)$.

The argument for Rel_1 is the same as that of Ord_0, with super- and sublevel sets switched. Thus our final case is when u and v are paired in Ext_1. We have already shown that all points in $\text{Rel}_1, \text{Ord}_0$, and Ext_0 from X stay paired under Φ, so all that remains is to be sure $\Phi(u)$ and $\Phi(v)$ cannot pair with any other points from Ext_1. Since u is the highest up fork that spans a loop with v, there are exactly two connected components in $f^{-1}((a_i, a_j))$ which attach to u and v in X via the inclusion maps into $f^{-1}([a_i, a_j])$. Therefore, by Lemma 4, we have the commutative diagram

$$\begin{array}{ccc} f_\varepsilon^{-1}(a+\varepsilon, b-\varepsilon) & \xrightarrow{\simeq} & f^{-1}(a,b) \\ \downarrow & & \downarrow \\ f_\varepsilon^{-1}[a+\varepsilon, b-\varepsilon] & \xrightarrow{\simeq} & f^{-1}[a,b] \end{array}$$

and thus there are exactly two connected components in $f_\varepsilon^{-1}((a_i + \varepsilon, a_j - \varepsilon))$ which attach to $\Phi(u)$ and $\Phi(v)$ in $S_\varepsilon(X, f)$ via the inclusion maps. Therefore, $\Phi(u)$ and $\Phi(v)$ will remain paired in Ext_1. ◀

Again, the prior result is only proven here for generic graphs, but the proof can be adapted to work in non-generic graphs as well. In that case, a vertex of the graph simply corresponds to multiple critical values, so that any vertex of degree $d > 2$ in the graph will appear in $d - 2$ persistence pairs in the diagram, making the case analysis more complex.

Finally, we arrive at the true main result of this section, where we can use the characterization of the movement of the critical points to keep track of movement in the persistence diagram. We note that part of this corollary is also implied in [24], via Propositions 7.1 and 7.4 in that work, where they study the movement of points in the persistence diagrams of Reeb graphs under smoothing via the construction of formigrams, although without the geometric analysis that yields our bijection Φ.

▶ **Corollary 7.** *Consider a generic Reeb graph (X, f), a value ε, and the bijection Φ as in Theorem 5. For every point $(a, b) = (f(v), f(u))$ in the persistence diagram of (X, f), the corresponding point $(f(\Phi(v)), f(\Phi(u)))$ in the persistence diagram of $S_\varepsilon(X, f)$ is located (depending on its type) as follows:*

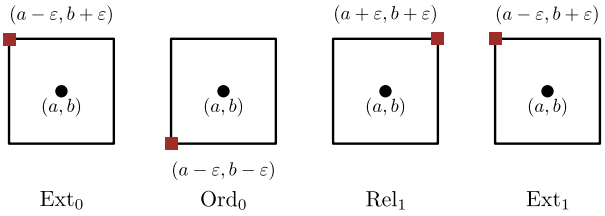


Figure 7 A point (a, b) in the diagram for the Reeb graph (X, f) moves to a point in the diagram of $S_\varepsilon(X, f)$ depending on which type of persistence point it represents.

$$\begin{array}{c|c|c|c} Ext_0 & (a-\varepsilon, b+\varepsilon) & Rel_1 & (a+\varepsilon, b+\varepsilon) \\ Ord_0 & (a-\varepsilon, b-\varepsilon) & Ext_1 & \begin{cases} (a-\varepsilon, b+\varepsilon) & \text{if } a-b > 2\varepsilon \\ \text{removed} & \text{if } a-b \leq 2\varepsilon. \end{cases} \end{array}$$

See Fig. 7 for a visual representation of the movement of the points; and Fig. 9 for the movements of point in a diagram where smoothing without truncation is represented by the case where $\tau = 0$.

Proof. This proof is bookkeeping to keep track of the types of top and bottom points forming each pair in the persistence diagram and noting how they move via Thms. 5 and 6. ◀

3.2 Truncating Smoothed Reeb Graphs

We next consider truncated smoothing, and prove an analogous characterization of its impact on the persistence diagram. We know that $\beta_0(X, f) = \beta_0(S_\varepsilon^\tau(X, f))$, since their π_0 groups are isomorphic [10]. We begin by proving an analogous result on $\pi_1(S_\varepsilon^\tau(X, f))$.

▶ **Lemma 8.** *For $0 \leq \tau < 2\varepsilon$, $\pi_1[\eta] : \pi_1(S_\varepsilon^\tau(X, f)) \to \pi_1(S_\varepsilon(X, f))$ is an isomorphism.*

Proof. Consider a loop in (X, f) with critical values a_i and a_j. By Theorem 5, the loop either disappears under the smoothing functor (if its height is less than $\varepsilon/2$) or is still present in the smoothed graph (under the bijection Φ). By Prop. 4.3 and Lemma 4.4 of [10], for $\tau \leq 2\varepsilon$, truncation of the smoothed graph will not reach any essential fork. Hence each loop maps to a unique loop in the truncated graph, under the natural inclusion from $S_\varepsilon^\tau(X, f) \hookrightarrow S_\varepsilon(X, f)$, and the resulting isomorphism follows since no new loops can be created under truncation. ◀

The goal of the following proposition is to see how truncation affects the four types of persistence points (Ext_0, Ord_0, Rel_1 and Ext_1) after smoothing, with assumptions to ensure that truncation does not change the topology of a given graph. See Fig. 8 for a visualization of the theorem, which constrains the effect of truncated smoothing on the different points of the persistence diagram.

▶ **Proposition 9.** *Fix $0 \leq \tau \leq 2\varepsilon$. Suppose (X, f) is a generic Reeb graph. For every point (a, b) in the persistence diagram of the Reeb graph, the corresponding point in the persistence diagram of the truncated smoothed Reeb graph $S_\varepsilon^\tau(X, f)$ is*

$$\begin{array}{c|c|c|c} Ext_0 & (a-\varepsilon+\tau, b+\varepsilon-\tau) & Rel_1 & (a+\varepsilon-\tau, b+\varepsilon) \\ Ord_0 & (a-\varepsilon+\tau, b-\varepsilon) & Ext_1 & (a-\varepsilon, b+\varepsilon) \end{array}$$

if the new point is on the same side of the diagonal as (a, b), and the point is removed completely if not.

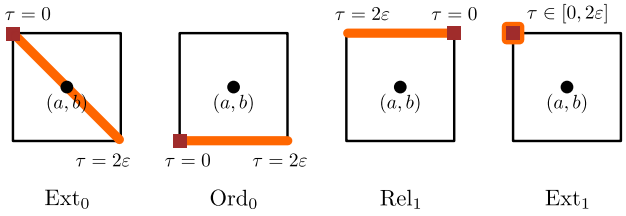


Figure 8 A point (a, b) in the diagram for the Reeb graph (X, f) moves to a point in the diagram of $S_\varepsilon^\tau(X, f)$ depending on which type of persistence point it represents. The red dot again indicates the behavior of smoothing (when $\tau = 0$), and the orange lines indicate the range of possible values in the diagram reachable by different values of $\tau < 2\varepsilon$.

Proof. By Lemma 8 and our assumptions on X, we know that truncation on the smoothed graph moves any critical point that is a local minimum up by τ, any local maximum down by τ, and leaves up- and down-forks unchanged so long as the tail is not removed. The proof then follows from the same bookkeeping as in Corollary 7, after tracking these new critical values. ◀

See Fig. 8 for a visualization of the behavior of different types of points in the diagram. We refer to Fig. 9 for further visualization of how this looks on a full diagram.

4 Inverse problems on Reeb graphs

We are now in a position to make use of this characterization to provide a solution to the time varying inverse problem in a particular restricted setting. In the most general setting, our problem can be stated as follows. We assume we are given a path in persistence diagram space, colloquially known as a vineyard: i.e. a function $\gamma : \mathbb{R}_{\geq 0} \to \mathbf{Pers}$ which is continuous with respect to the topology induced by the bottleneck distance. We further assume we have been provided with an initial Reeb graph $R_0 = (G_0, f_0)$. The goal is to find a path in the space of Reeb graphs $\Gamma : \mathbb{R}_{\geq 0} \to \mathbf{Reeb}$, continuous with respect to the interleaving distance [16], for which $\Gamma(0) = R_0$ and $\mathcal{P}(\Gamma(t)) = \gamma(t)$.

We restrict our view to essentially creating a notion of piecewise linear paths, defined by linear interpolation between the path defined at discrete times $0 = t_0 < t_1 < t_2 < t_3 \cdots$. For the sake of notation, we will denote $\gamma(t_i) = D_i$, and for the Reeb graph path we will construct, $\Gamma(t_i) = R_i = (G_i, f_i)$. Geodesics in persistence diagram space are defined by matchings arising from the bottleneck distance computation. While a minimum cost matching is not unique in this setting, say we have a matching $M_i : D_i \to D_{i+1}$, then a geodesic between diagrams D_i and D_{i+1} is given by sliding each point x at constant speed along the line between $x \in D_i$ and $M_i(x) \in D_{i+1}$.

Following Prop. 9, we define the map $\Psi_\varepsilon^\tau : \mathbf{Pers} \to \mathbf{Pers}$ on diagrams by defining a map on each point $(a, b) \in D$ as follows:

$$\varphi_\varepsilon^\tau(a, b) = \begin{cases} (a - \varepsilon + \tau, b + \varepsilon - \tau) & \text{if the type of } (a, b) \text{ is } Ext_0 \text{ in } D \\ (a - \varepsilon + \tau, b - \varepsilon) & \text{if the type of } (a, b) \text{ is } Ord_0 \text{ in } D \\ (a + \varepsilon - \tau, b + \varepsilon) & \text{if the type of } (a, b) \text{ is } Rel_1 \text{ in } D \\ (a - \varepsilon, b + \varepsilon) & \text{if the type of } (a, b) \text{ is } Ext_1 \text{ in } D. \end{cases}$$

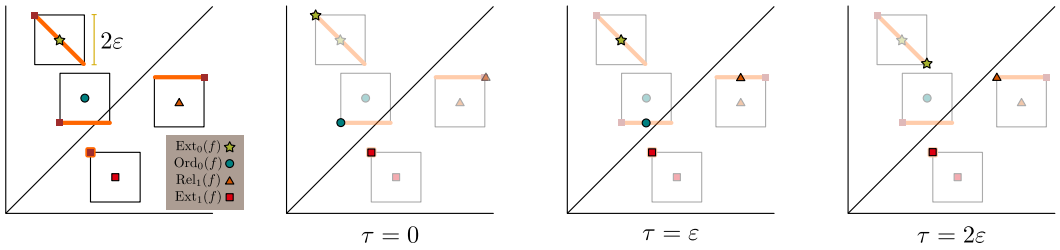


Figure 9 Given an initial extended persistence diagram (left), the three diagrams at right are the resulting diagrams after smoothing by ε and truncating by $\tau = 0, \varepsilon, 2\varepsilon$ respectively. Note that in this example, for $\tau = 2\varepsilon$ the point in Ord_0 is removed completely since its location on the box would be on the other side of the diagonal.

Then the map Ψ_ε^τ is defined by

$$\Psi_\varepsilon^\tau(D) = \{\varphi_\varepsilon^\tau(x,y) \mid (x,y) \in D \text{ with } \varphi_\varepsilon^\tau(x,y) \text{ on the same side of the diagonal } \Delta \text{ as } (x,y)\}.$$

We abuse notation and also view this construction as a matching on the points of related diagrams themselves. Given an ε and τ, all points of the same type (Ord_0, Rel_1, etc) move in the same direction. Thus, for a given direction vector $\vec{v}$, which depends on ε, τ, and point type, we can define an updated single-type diagram $D_{\vec{v}}$ by:

$$D_{\vec{v}} = D + \vec{v} := \{x + \vec{v} \mid x \in D \text{ with } x \text{ and } x + \vec{v} \text{ on the same side of the diagonal}\}.$$

Given this updated diagram, we have a matching defined by:

$$\omega : x \mapsto \begin{cases} x + \vec{v} & \text{if } x + \vec{v} \in D_{\vec{v}} \\ \Delta & \text{otherwise.} \end{cases} \tag{2}$$

With this setup, we begin with the following theorem, which proves directly that our truncated smoothing map in fact is at least locally bottleneck optimal, motivating our use of this operation for inverse problems.

▶ **Theorem 10.** *For a diagram D with only points of a single type, and $\vec{v}$ with magnitude*

$$|\vec{v}| < \frac{1}{2} \min_{\substack{x \in D \\ y \in D \cup \Delta \\ x \neq y}} \{\|x - y\|_\infty\},$$

the matching induced by ω achieves the bottleneck distance between D and $D_{\vec{v}}$.

Proof. Let $\varphi : D \to D_{\vec{v}}$ be the map sending each off-diagonal point in D to its nearest neighbor in $D_{\vec{v}}$ under $\|\cdot\|_\infty$. We first show that $\varphi(x)$ must be $x + \vec{v}$. By way of contradiction, assume $\varphi(x) = x' + \vec{v}$, which implies that $\|x - \varphi(x)\| \leq \|x - (x + \vec{v})\|_\infty = \|\vec{v}\|$. But then

$$\|x - x'\| \leq \|x - (x' + \vec{v}) + \vec{v}\| \leq \|x - \varphi(x)\| + \|\vec{v}\| \leq 2\|\vec{v}\|$$

contradicting the assumption of the theorem. Note that by assumption, no points of D have been removed in $D_{\vec{v}}$, since the magnitude of $\vec{v}$ cannot reach the diagonal. Thus, φ gives a bijection on the off-diagonal points, and further, has a bottleneck matching score of $\|\vec{v}\|_\infty$.

If the bottleneck distance between D and $D_{\vec{v}}$ were $\delta < \|\vec{v}\|_\infty$, there would be a matching sending each point in D to a point at most δ away, but this of course contradicts the nearest neighbor definition of φ. ◀

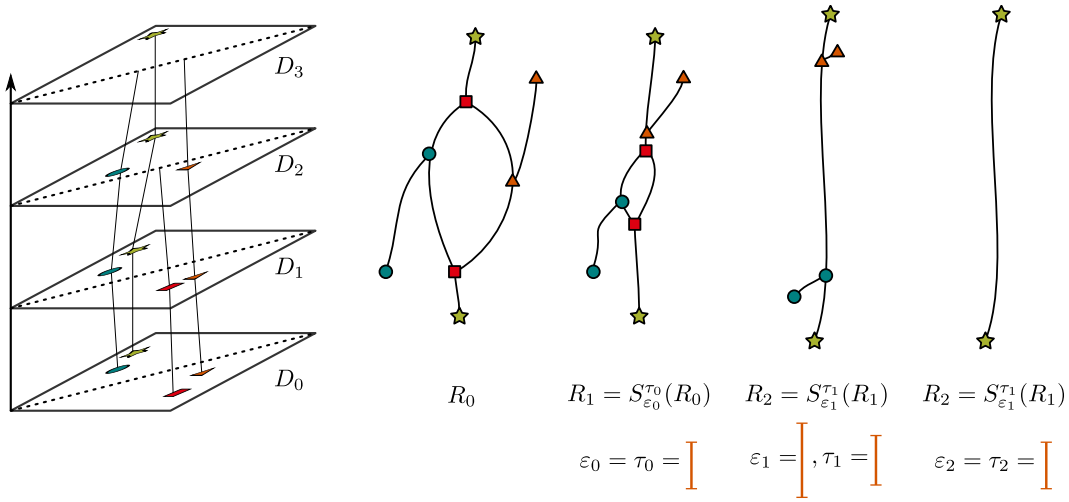


Figure 10 An example of a sequence of admissible diagrams (left) realized by a series of Reeb graphs with ε_i and τ_i shown below.

Next, we turn our attention to when a sequence of diagrams (i.e. a *vineyard*) is realizable by Reeb graphs using smoothing and truncation operations.

▶ **Definition 11.** *A sequence of input diagrams* $\{D_i\}_{i=0}^N$ *is* admissible *if for every* i*, there exists a pair* (ε_i, τ_i) *with* $\tau_i < 2\varepsilon_i$*, such that* $D_{i+1} = \Psi^{\tau_i}_{\varepsilon_i}(D_i)$*.*

For example, see Fig. 10, where pairs of admissible diagrams are shown. This input restriction is built to provide a solution to the inverse problem as follows.

▶ **Theorem 12.** *Assume* $\{D_i\}_{i=0}^N$ *is admissible and a Reeb graph* R_0 *is given so that* $\mathcal{P}(R_0) = D_0$*. Then there is a sequence of Reeb graphs* R_i *for which* $\mathcal{P}(R_i) = D_i$*.*

Proof. The proof proceeds by induction. For $i = 0$, and since the D_i's are admissible, by Proposition 9 we have a pair (ε_0, τ_0) such that D_1 is the diagram of $S^{\tau_0}_{\varepsilon_0}(R_0)$ i.e $\Psi^{\tau_0}_{\varepsilon_0}(D_0) = D_1$. We can then define the first Reeb graph to be $R_1 = S^{\tau_0}_{\varepsilon_0}(R_0)$, which has the property that $\mathcal{P}(R_1) = D_1$ by constructing the truncated smoothing.

Next we for any $i \geq 0$, we assume we have a Reeb graph R_i such that $\mathcal{P}(R_i) = D_i$, and wish to build R_{i+1} such that $\mathcal{P}(R_{i+1}) = D_{i+1}$. We proceed in the same manner as the base case by finding (ε_i, τ_i) using the fact that the D_i collection is admissible, and setting $R_{i+1} = S^{\tau_i}_{\varepsilon_i}(R_i)$. Then by Proposition 9, $\mathcal{P}(R_{i+1}) = \Psi^{\tau_i}_{\varepsilon_i}(D_i) = D_{i+1}$. ◀

▶ **Corollary 13.** *Given an admissible sequence of diagrams* $\{D_i\}_{i=0}^N$*, we can extend this to a piecewise linear vineyard given by* $\gamma(i+t) = D_i + t\vec{v_i}$ *for integer* i *and* $t \in [0, 1)$*, where* $\vec{v_i}$ *is set for the pair* D_i, D_{i+1} *by the formula in Equation* (2) *and is distinct by diagram subtype. Then the path of Reeb graphs* $\Gamma(i+t) = S^{t\tau_i}_{t\varepsilon_i}(R_i)$ *realizes this path; i.e.* $\mathcal{P}(\Gamma(i+t)) = \gamma(i+t)$*.*

Proof. This corollary follows from Theorem 12 as follows. First, note that if the pair D_i, D_{i+1} is admissible when using ε_i and τ_i, we get the direction $\vec{v_i}$ for each type of point as in Eqn. (2). Then, we can linearly interpolate along $\vec{v_i}$ in the diagram to get an intermediate diagram, and Theorems 5 and 6 tell us there are intermediate values of ε and τ that realize the smoothing and diagram for any point along the interpolation. ◀

5 Conclusions and future directions

In this paper, we have provided a complete characterization of the behavior of vertices in a Reeb graph under the truncation and smoothing operations with respect to their extended persistence. We showed how this classification can be used to open doors to further utilization of the smoothing procedure itself. In particular, as we have seen, this characterization can be translated into simple descriptions of the available paths in extended persistence diagram space under these transformations, yielding tractable solutions to (an admittedly rather restricted version of) inverse problems in topological data analysis.

This work suggests many possible future directions to study. While we have only dealt with piecewise linear vineyards in this work, it seems likely that there will be ways to loosen our restrictions in Theorem 12. Our analysis of smoothing suggests many possible relaxations, where for example different sections of the Reeb graphs are smoothed and truncated (or even "untruncated" and "unsmoothed") by varying amounts or where more relaxed constraints on admissible could yield approximate solutions to the inverse problem. This broader framework of piecewise linear paths would be interesting if it can be used to approximate more general vineyards, yielding the potential for approximate solutions of a more general inverse problem. In addition, while we have chosen to analyze using the bottleneck distance in Theorem 10, there are many other metrics on both Reeb graphs and persistence diagrams to consider, which we conjecture may also be locally optimal.

References

1 Pankaj K. Agarwal, Herbert Edelsbrunner, John Harer, and Yusu Wang. Extreme elevation on a 2-manifold. *Discrete & Computational Geometry*, 36(4):553–572, September 2006. doi:10.1007/s00454-006-1265-8.

2 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.

3 Ulrich Bauer, Claudia Landi, and Facundo Mémoli. The Reeb graph edit distance is universal. *Foundations of Computational Mathematics*, 12 2020. doi:10.1007/s10208-020-09488-3.

4 Ulrich Bauer, Elizabeth Munch, and Yusu Wang. Strong equivalence of the interleaving and functional distortion metrics for Reeb graphs. In Lars Arge and János Pach, editors, *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 461–475, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2015/5146, doi:http://dx.doi.org/10.4230/LIPIcs.SOCG.2015.461.

5 S. Biasotti, A. Cerri, A. Bronstein, and M. Bronstein. Recent trends, applications, and perspectives in 3d shape similarity assessment. *Computer Graphics Forum*, 35(6):87–119, October 2015. URL: http://dx.doi.org/10.1111/cgf.12734, doi:10.1111/cgf.12734.

6 S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392(1-3):5–22, February 2008. doi:10.1016/j.tcs.2007.10.018.

7 Brian Bollen, Erin Wolf Chambers, Joshua Levine, and Elizabeth Munch. Reeb graph metrics from the ground up. 2021. arXiv:2110.05631.

8 Mathieu Carrière and Steve Oudot. Local equivalence and intrinsic metrics between Reeb graphs. In Boris Aronov and Matthew J. Katz, editors, *33rd International Symposium on Computational Geometry (SoCG 2017)*, volume 77 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 25:1–25:15, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2017/7179, doi:10.4230/LIPIcs.SoCG.2017.25.

9 Mathieu Carrière and Steve Oudot. Structure and stability of the one-dimensional mapper. *Foundations of Computational Mathematics*, oct 2017. doi:10.1007/s10208-017-9370-z.

10 Erin Wolf Chambers, Elizabeth Munch, and Tim Ophelders. A family of metrics from the truncated smoothing of Reeb graphs. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. Main session acceptance rate: 35%. URL: https://drops.dagstuhl.de/opus/volltexte/2021/13821, arXiv:2007.07795, doi:10.4230/LIPIcs.SoCG.2021.22.

11 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, December 2006. doi:10.1007/s00454-006-1276-5.

12 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Extending persistence using Poincaré and Lefschetz duality. *Foundations of Computational Mathematics*, 9(1):79–103, April 2009. URL: http://dx.doi.org/10.1007/s10208-008-9027-z, doi:10.1007/s10208-008-9027-z.

13 David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. *Proceedings of the twenty-second annual symposium on Computational geometry - SCG '06*, page 119, 2006. URL: http://portal.acm.org/citation.cfm?doid=1137856.1137877, doi:10.1145/1137856.1137877.

14 Kree Cole-McLaughlin, Herbert Edelsbrunner, John Harer, Vijay Natarajan, and Valerio Pascucci. Loops in Reeb graphs of 2-manifolds. In *Proceedings of the nineteenth annual*

symposium on Computational geometry, SCG '03, pages 344–350, New York, NY, USA, 2003. ACM. URL: http://doi.acm.org/10.1145/777792.777844, doi:10.1145/777792.777844.

15 Justin Curry, Haibin Hang, Washington Mio, Tom Needham, and Osman Berat Okutan. Decorated merge trees for persistent topology. *Journal of Applied and Computational Topology*, 6(3):371–428, March 2022. doi:10.1007/s41468-022-00089-3.

16 Vin De Silva, Elizabeth Munch, and Amit Patel. Categorified Reeb graphs. *Discrete & Computational Geometry*, 55(4):854–906, June 2016. doi:10.1007/s00454-016-9763-9.

17 Vin de Silva, Elizabeth Munch, and Anastasios Stefanou. Theory of interleavings on categories with a flow. *Theory and Applications of Categories*, 33(21):583–607, 2018. URL: http://www.tac.mta.ca/tac/volumes/33/21/33-21.pdf.

18 Tamal K Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 2021.

19 Barbara Di Fabio and Claudia Landi. The edit distance for Reeb graphs of surfaces. *Discrete & Computational Geometry*, 55(2):423–461, 1 2016. doi:10.1007/s00454-016-9758-6.

20 Harish Doraiswamy and Vijay Natarajan. Efficient algorithms for computing Reeb graphs. *Computational Geometry*, 42(6-7):606–616, August 2009. doi:10.1016/j.comgeo.2008.12.003.

21 Charles Gueunet, Pierre Fortin, Julien Jomier, and Julien Tierny. Task-based Augmented Reeb Graphs with Dynamic ST-Trees. In Hank Childs and Steffen Frey, editors, *Eurographics Symposium on Parallel Graphics and Visualization*. The Eurographics Association, 2019. doi:10.2312/pgv.20191107.

22 William Harvey, Yusu Wang, and Rephael Wenger. A randomized $O(m \log m)$ time algorithm for computing Reeb graphs of arbitrary simplicial complexes. In *Proceedings of the 2010 annual symposium on Computational geometry*, SoCG '10, pages 267–276, New York, NY, USA, 2010. ACM. URL: http://doi.acm.org/10.1145/1810959.1811005, doi:10.1145/1810959.1811005.

23 Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

24 Woojin Kim and Facundo Mémoli. Extracting persistent clusters in dynamic data via möbius inversion. *Discrete & Computational Geometry*, 71(4):1276–1342, October 2023. doi:10.1007/s00454-023-00590-1.

25 J.R. Munkres. *Topology*. Topology. Prentice-Hall, 2000. URL: https://books.google.com.sa/books?id=NnCjQgAACAAJ.

26 Steve Oudot and Elchanan Solomon. *Inverse Problems in Topological Persistence*, pages 405–433. Springer International Publishing, 2020. doi:10.1007/978-3-030-43408-3_16.

27 Steve Y. Oudot. *Persistence Theory - From Quiver Representations to Data Analysis*, volume 209 of *Mathematical surveys and monographs*. American Mathematical Society, 2015. URL: http://bookstore.ams.org/surv-209/.

28 Salman Parsa. A deterministic $O(m \log m)$ time algorithm for the Reeb graph. In *Proceedings of the 28th annual ACM symposium on Computational geometry*, SoCG '12. ACM, 2012.

29 Georges Reeb. Sur les points singuliers d'une forme de pfaff complèment intégrable ou d'une fonction numérique. *Comptes Rendus de L'Académie ses Séances*, 222:847–849, 1946.

30 Lin Yan, Talha Bin Masood, Raghavendra Sridharamurthy, Farhan Rasheed, Vijay Natarajan, Ingrid Hotz, and Bei Wang. Scalar field comparison with topological descriptors: Properties and applications for scientific visualization. *Computer Graphics Forum*, 40(3):599–633, jun 2021. doi:10.1111/cgf.14331.

31 Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, November 2004. doi:10.1007/s00454-004-1146-y.