

Bounding the Interleaving Distance for Mapper Graphs with a Loss Function

Erin Wolf Chambers¹, Elizabeth Munch^{2*}, Sarah Percival³,
Bei Wang⁴

¹University of Notre Dame, Notre Dame, IN, 46556, USA.

^{2*}Michigan State University, East Lansing, MI, 48824, USA.

³University of New Mexico, Albuquerque, NM, 87106, USA.

⁴University of Utah, Salt Lake City, UT, 84112, USA.

*Corresponding author(s). E-mail(s): muncheli@msu.edu;

Contributing authors: echambe2@nd.edu; spercival@unm.edu;
beiwang@sci.utah.edu;

Abstract

Data consisting of a graph with a function mapping into \mathbb{R}^d arise in many data applications, encompassing structures such as Reeb graphs, geometric graphs, and knot embeddings. As such, the ability to compare and cluster such objects is required in a data analysis pipeline, leading to a need for distances between them. In this work, we study the interleaving distance on discretization of these objects, called mapper graphs when $d = 1$, where functor representations of the data can be compared by finding pairs of natural transformations between them. However, in many cases, computation of the interleaving distance is NP-hard. For this reason, we take inspiration from recent work by Robinson to find quality measures for families of maps that do not rise to the level of a natural transformation, called assignments. We then endow the functor images with the extra structure of a metric space and define a loss function which measures how far an assignment is from making the required diagrams of an interleaving commute. Finally we show that the computation of the loss function is polynomial with a given assignment. We believe this idea is both powerful and translatable, with the potential to provide approximations and bounds on interleavings in a broad array of contexts.

Keywords: Interleaving distance, Mapper graph, Reeb graph, geometric graph, topological data analysis

1 Introduction

Graphs with additional geometric information arise in many contexts in data analysis. For instance, a *geometric graph* is generally defined as an abstract graph along with a well-behaved embedding of the graph into \mathbb{R}^2 , while a graph with a well-behaved map into \mathbb{R} is called a *Reeb graph*. In particular from the viewpoint of the Reeb graph, these types of input data can arise by studying connected component structures from more general input \mathbb{R}^d -spaces, meaning a topological space \mathbb{X} with a function $f : \mathbb{X} \rightarrow \mathbb{R}^d$. Such graphs are a fundamental object used to model a wide range of data sets, ranging from maps and trajectories [1, 2], to commodity networks (e.g. transportation networks [3, 4]) and shape skeletons for object recognition [5, 6]. The ability to compare, cluster, and simplify such representative objects is essential in a data analysis pipeline, leading to a need for theoretically motivated and computable distances. In this paper, we study a distance for a discretization of the input data, known as mapper [7]. That is, starting from a topological space \mathbb{X} with a function $f : \mathbb{X} \rightarrow \mathbb{R}^d$ (resp. a point cloud P with a function $f : P \rightarrow \mathbb{R}^d$), mapper is an encoding of the connected components (resp. clusters) of $f^{-1}(U_\alpha)$ for some cover $\mathcal{U} = \{U_\alpha\}$ of \mathbb{R}^d defined as the nerve of the pullback cover. When $d = 1$, this results in a graph structure called a *mapper graph* (see Fig. 1). Since there are higher dimensional cells, we call the resulting construction for $d > 1$ an \mathbb{R}^d -mapper complex; however we abuse terminology and generally call this construction a mapper graph whether the dimension is 1 or not.

There has been extensive work on metrics for general graphs, geometric graphs, and Reeb graphs (see surveys [8–11], [2], and [12, 13] respectively). In this paper, we will draw inspiration from the interleaving distance; specifically, we develop a natural extension of the interleaving distance on Reeb graphs [14] to the setting of mapper graphs. Interleaving distances arose in the context of generalizing the bottleneck distance for persistence modules [15] and were subsequently translated to more general categorical frameworks in [14, 16]. With the exception of 1-parameter persistence [17], the interleaving distance is NP-hard in many contexts including multi-parameter persistence [18, 19], and Reeb graphs [14]. However, some additional structural information can give better algorithms such as FPT computation for merge trees [20], and polynomial time for formigrams [21] and labeled merge trees [22, 23]. Indeed, the closest work to our approach is work providing bounds for the interleaving distance restricted to merge trees: [24] use the Gromov-Wasserstein distance to find a leaf labeling that gives an upper bound using the easy to compute labeled interleaving distance [22, 23]; while [25] uses the map formulation of [20, 26] with an integer linear program to provide a bound. See [18] for a recent summary of interleaving distance complexity results.

When $d = 1$, there is already work using the interleaving distance to relate the Reeb graph and its mapper graph [27–31]. We will encode the structure of our more general \mathbb{R}^d -mapper complexes in a discretized setting by imposing a grid structure K on \mathbb{R}^d . Then we can represent the input data $f : X \rightarrow \mathbb{R}^d$ as a cosheaf of the form $F : \mathbf{Open}(K) \rightarrow \mathbf{Set}$ where we store the path-connected components of inverse images of open sets $\pi_0(f^{-1}(U))$. The idea of the interleaving distance, in this context, is to compare two cosheaves $F, G : \mathbf{Open}(K) \rightarrow \mathbf{Set}$ using a pair of natural transformations $\varphi : F \Rightarrow G^m$ and $\psi : G \Rightarrow F^n$ mapping into relaxations of the original inputs.

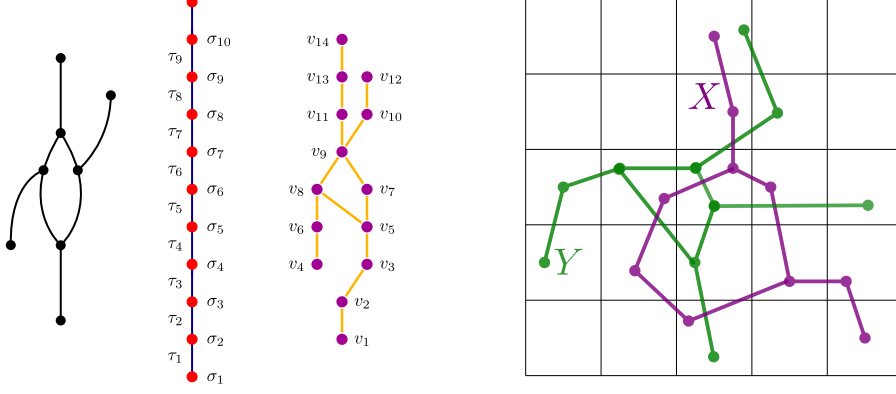


Fig. 1 (Left) An example of a Reeb graph ($d = 1$) where the discretization of \mathbb{R} is used to generate a mapper graph. (Right) An example of a pair of geometric graphs ($d = 2$) which we study using the connected components restricted to the grid.

The complexity of computing this distance then relies on finding the smallest n with available φ and ψ maps, which in our setting immediately connects to hard underlying problems such as graph isomorphism.

The ideas building this distance are rooted in previous work that study interleavings in related contexts. In some sense, we can view this distance as a discretized cosheaf version of the continuous sheaf version of the interleaving distance that was previously studied [32]. It can fit into the more general framework of an interleaving distance arising from a category with a flow [33], or as an interleaving distance on generalized persistence modules with a family of translations [16], but prior work in these areas focused on theoretical properties and did not address computational aspects, as the more general framework makes such study incredibly difficult. Perhaps the closest version of this distance is mentioned as a special case of a general categorical framework [31]; however, that setting keeps the thickening of the open sets structure tightly bound to thickening in \mathbb{R}^d , whereas we choose to define the distance entirely over the combinatorial structures.

While all this prior work is powerful in theory, the computational complexity of the construction in more general settings has meant a lack of the use of the interleaving distance in practice. To circumvent these issues, we take inspiration from recent work of Robinson [34] to define quality measures for families of maps that do not rise to the level of a natural transformation, in order to allow for non-optimal maps φ and ψ in this framework. We then apply these quality measures to \mathbb{R}^d -mapper complexes, in the hopes of utilizing algorithms from geometry and graph theory to make computation more feasible.

In particular, in [34] the object of study is a single input assignment of data of the form $P : \mathbf{Open}(X) \rightarrow \mathbf{Set}$ and, with the added structure of a pseudometric for each set $P(U)$, provides a measurement for how far the input data is from having a global section. In our work, we instead work with a pair of functors $F, G : \mathbf{Open}(K) \rightarrow \mathbf{Set}$ as input, and study collections of maps $\varphi = \{\varphi_U : F(U) \rightarrow G(U^n) \mid U\}$ and $\psi = \{\psi_U : G(U) \rightarrow F(U^n) \mid U\}$ which we call an *assignment* when they do not necessarily form a

true interleaving. We then endow the image with the extra structure of a metric space, so that we have pairs $(F(U), d_U)$ for every open set U . Using this metric structure, we define a loss function $L(\varphi, \psi)$ which measures how far the required diagrams of an interleaving are from commuting, given any input assignment (Thm. 3.8). We modify this bound by only focusing on the loss function computed for a basis of the topology, $L_B(\varphi, \psi)$ (Thm. 3.16), which not only improves the computational complexity but also improves the bound. Then, we show that the computation of the bound is polynomial, opening up the potential for algorithmic approximation of the interleaving distance. Throughout, we show examples encoding the data of a *geometric graph* (i.e. a graph G with a straight line embedding $f : G \rightarrow \mathbb{R}^2$) or a Reeb graph (a graph G with a straight line map to \mathbb{R}); see Fig. 1 for an illustration. However, this kind of input is not a requirement for our framework.

We note that this paper is the first step in a larger project. That is, the paper here presents a loss function that can be computed given an input n -assignment φ, ψ and results in a bound on the distance by explicitly constructing an $(n + L_B(\varphi, \psi))$ -interleaving. As with many garbage-in-garbage-out settings, this bound is only as good as the input, but this study seeks to determine the most general possible bound with no guarantees on the input at all. In the followup work [35], we include this loss function as part of an optimization strategy to update a given assignment in order to find better bounds as well as provide further study on how close to optimal is possible.

Outline. In Sec. 2 we provide the necessary technical background to set up the interleaving distance for mapper graph inputs. In Sec. 3, we define the loss functions and bounds. We discuss algorithmic requirements of the bound in Sec. 4. Next, we show how this loss function can be used to similarly bound the Reeb graph interleaving distance by approximating the Reeb graph with a mapper graph in Sec. 5. We include all technical proofs in Sec. 6. Finally, we discuss broader implications of this work in Sec. 7.

2 Technical Background

We will assume several example types of inputs in this paper. All are tied together by having data of the form $f : \mathbb{X} \rightarrow \mathbb{R}^d$, where \mathbb{X} is a topological space. In particular, we will require that $\pi_0(f^{-1}(U))$ is a finite set for some reasonable collection of open sets $U \subset \mathbb{R}^d$. We thus assume that $\mathbb{X} \subset \mathbb{R}^k$ is a semi-algebraic set and that f is a semi-algebraic map since this results in our desired restrictions.

2.1 Functors and Cosheaves

We give basic definitions for the category theoretic notions required in this paper, and direct the interested reader to [32, 36] for further details. A *category* \mathcal{C} consists of a collection of objects X, Y, Z, \dots and morphisms f, g, h, \dots with the following requirement: morphisms $f : X \rightarrow Y$ have designated domain X and codomain Y ; every object has a designated identity morphism $\mathbb{1}_X : X \rightarrow X$, and any pair of morphisms $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ has a composite morphism $g \circ f : X \rightarrow Z$. These objects and morphisms satisfy an identity axiom, where $f : X \rightarrow Y$ is the same as the $\mathbb{1}_Y \circ f$ and $f \circ \mathbb{1}_X$; and composition (denoted by \circ but often dropped when unnecessary) is

associative, so $h \circ (g \circ f) = (h \circ g) \circ f$. Some example categories are **Set** where objects are finite sets and morphisms are set maps; **Top** where objects are topological spaces and morphisms are continuous functions; and **Open**(\mathbb{X}) for a given topological space \mathbb{X} , where the objects are open sets and morphisms are given by inclusion.

A *functor* $F : \mathcal{C} \rightarrow \mathcal{D}$ is a map between categories preserving the relevant structures. Specifically, for every object $X \in \mathcal{C}$ there is an object $F(X) \in \mathcal{D}$, and for every morphism $f : X \rightarrow Y$, there is a morphism $F[f] : F(X) \rightarrow F(Y)$. To be a functor, F must further satisfy that for any $X \in \mathcal{C}$, $F[1_X] = 1_{F(X)}$ and for any composable pair $f, g \in \mathcal{C}$, we have $F[gf] = F[g]F[f]$. Given functors $F, G : \mathcal{C} \rightarrow \mathcal{D}$, a *natural transformation* $\eta : F \Rightarrow G$ consists of a map $\eta_X : F(X) \rightarrow G(X)$ for every $X \in \mathcal{C}$ (called the components) so that for any morphism $f : X \rightarrow Y$ in \mathcal{C} , the following diagram

$$\begin{array}{ccc} X & F(X) & \xrightarrow{\eta_X} G(X) \\ \downarrow f & F[f] \downarrow & \downarrow G[f] \\ Y & F(Y) & \xrightarrow{\eta_Y} G(Y) \end{array}$$

commutes. One example is $\pi_0 : \mathbf{Top} \rightarrow \mathbf{Set}$, where $\pi_0(\mathbb{X})$ is the set of path-connected components of \mathbb{X} , and morphisms are set maps $\pi_0[f] : \pi_0(\mathbb{X}) \rightarrow \pi_0(\mathbb{Y})$ sending a connected component A in \mathbb{X} to the connected component of $f(A)$ in \mathbb{Y} . Note that throughout the paper, we use the term *component* to mean *path-connected component*.

A diagram is a functor $F : J \rightarrow \mathcal{C}$ where J is a small category. In essence, this construction picks out a collection of objects $F(j)$ and a collection of morphisms $F(j) \rightarrow F(k)$. A cocone on a given diagram is a natural transformation $\lambda : F \rightarrow c$ where we abuse notation to write that $c : J \rightarrow \mathcal{C}$ is the constant functor returning the object $c(j) = c \in \mathcal{C}$ for all $j \in J$. We often call the components $\lambda_j : F(j) \rightarrow c$ the *legs*, and note that this requirement says that for any $f : j \rightarrow k$ in J , $\lambda_k \circ F[f] = \lambda_j$. A cocone $\lambda : F \rightarrow c$ is called a colimit if for any other cocone $\lambda' : F \rightarrow c'$, there is a unique morphism $u : c \rightarrow c'$ such that

$$\begin{array}{ccc} F(j) & \xrightarrow{F[f]} & F(k) \\ \lambda_j \searrow & & \swarrow \lambda_k \\ & c & \\ \lambda'_j \searrow & \downarrow u & \swarrow \lambda'_k \\ & c' & \end{array}$$

commutes for all $f : j \rightarrow k$ in J .

We will be particularly interested in functors of the form $F : \mathbf{Open}(X) \rightarrow \mathbf{Set}$, which can also be called *pre-cosheaves*. A pre-cosheaf is a *cosheaf* if it satisfies a gluing axiom, meaning that $F(U)$ is entirely determined by $F(U_\alpha)$ for any cover $\{U_\alpha\}_\alpha$. Specifically, given an open set U and a cover $\{U_\alpha \mid \alpha \in A\}$ of U , define a category $\mathcal{U} = \{U_\alpha \cap U_{\alpha'} \mid \alpha, \alpha' \in A\}$ with morphisms given by inclusion. Then we have a diagram $F : \mathcal{U} \rightarrow \mathbf{Set}$, and as such can consider its colimit $\lambda : F \rightarrow L$. If the unique

map $L \rightarrow F(U)$ given by the colimit definition is an isomorphism, then F is called a *cosheaf*.

2.2 Functorial Representation of Embedded Data

Assume we are given as input a pair of compact topological spaces with valued functions $f : \mathbb{X} \rightarrow \mathbb{R}^d$ and $g : \mathbb{Y} \rightarrow \mathbb{R}^d$. We will construct a cover controlled by diameter $\delta \geq 0$ of the images $f(\mathbb{X}) \cup g(\mathbb{Y})$ in order to define the discretized map-per complex. Assume that a bounding box containing $f(\mathbb{X}) \cup g(\mathbb{Y})$ can be written as $[-B, B]^d = [-L\delta, L\delta]^d$.

Following [37], δ induces a discretization of $[-L\delta, L\delta]^d$ into a cubical complex in the following way. An *elementary interval* is an open interval in \mathbb{R} of the form¹ $(\ell\delta, (\ell+1)\delta)$ or a single point viewed as a degenerate interval $[\ell] := [\ell\delta, \ell\delta]$ for $\ell \in [-L, \dots, L] \subset \mathbb{Z}$. These are called non-degenerate and degenerate intervals, respectively. An elementary (open) cube Q is a finite product of d elementary intervals, i.e. $\sigma = I_1 \times I_2 \times \dots \times I_d \subset [-B, B]^d$. The dimension of a cube σ is given by the number of intervals used which are non-degenerate. This means that 0-cubes are vertices at grid locations $(i\delta, j\delta, \dots, k\delta) \in \delta \cdot \mathbb{Z}^d$, 1-cubes are edges (not including their endpoints), 2-cubes are squares (not including their boundaries), 3-cubes are voxels, etc. The collection of elementary cubes discretizing $[-B, B]^d$ is a finite cubical complex K . This construction comes with a face relation which gives a poset structure, where we write $\sigma \leq \tau$ iff $\sigma \subseteq \bar{\tau}$, where $\bar{\tau}$ denotes the closure of the set. In order to differentiate between the combinatorial and continuous settings, we write $|\sigma|$ for the geometric realization in \mathbb{R}^d of a combinatorial object $\sigma \in K$.

This complex K induces a cover \mathcal{U} of $[-B, B]^d$ as follows. For any cube $\sigma \in K$, we can find the upper closure of σ using the face relation, i.e. $\sigma^\uparrow = \{\tau \in K \mid \tau \geq \sigma\}$. The *cover element associated to σ* is $U_\sigma = \bigcup_{\tau \in \sigma^\uparrow} |\tau|$. Then we write the cover as $\mathcal{U} = \{U_\sigma \mid \sigma \in K\}$. Note that there is a poset relation on \mathcal{U} given by inclusion, and in particular, $U_\sigma \subseteq U_\tau$ iff $\tau \leq \sigma$.

We next endow the poset (\mathcal{U}, \subseteq) with the Alexandroff topology, following [38]. For any set $S \subseteq \mathcal{U}$, the upper closure, or *upset*, is $S^\uparrow = \{U \in \mathcal{U} \mid V \subseteq U, V \in S\}$ and the downward closure, or *downset*, is $S^\downarrow = \{U \in \mathcal{U} \mid U \subseteq V, V \in S\}$. We give (\mathcal{U}, \subseteq) the Alexandroff topology $\mathbf{Open}(\mathcal{U})$, where a set $S \subseteq K$ is open iff the following holds: for any $U \in S$ and any $V \subseteq U$, $V \in S$. Equivalently, this means that S is its own downset, i.e. $S = S^\downarrow$. See Fig. 2 for an example of this notation in the case of $d = 1$, where the open set U_{σ_i} is associated to the point $\sigma_i = i\delta$, and open set U_{τ_i} is associated to the edge $\tau_i = (i\delta, (i+1)\delta)$.

It can be checked that this topology has a basis given by the collection $\{S_\sigma\}_{\sigma \in K}$ where we write

$$S_\sigma := \{U_\sigma\}^\downarrow = \{U_\tau \in \mathcal{U} \mid U_\tau \subseteq U_\sigma\} = \{U_\tau \in \mathcal{U} \mid \tau \in \sigma^\uparrow\} \quad (2.1)$$

and call S_σ a *basic open set*. This complex is constructed so that for any subset $S \subset \mathcal{U}$, the geometric realization $|S| := \bigcup_{U \in S} U \subset \mathbb{R}^d$ is an open set; and further the notation

¹Note our use of open intervals here in order to have open sets in our cover later, which differs from the definition given in [37].

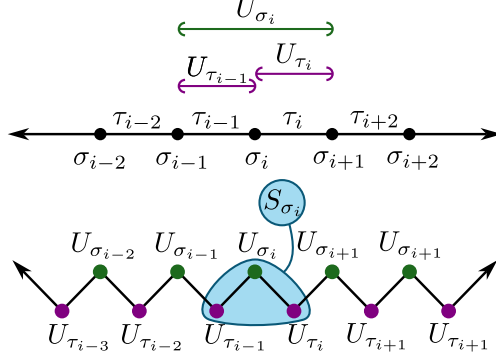


Fig. 2 Example of the notation for the case $d = 1$, where a discretization of the real line (top) induces a cover, and the poset representation of this cover can be seen in the bottom row. The basic open associated to U_{σ_i} is the three element set $S_{\sigma_i} = \{U_{\tau_{i-1}}, U_{\sigma_i}, U_{\tau_i}\}$.

is reasonable since the geometric realization of the basis set associated to σ is the same as the open set associated to σ , i.e. $|S_\sigma| = U_\sigma$ for all $\sigma \in K$. Again, see Fig. 2 for an example of this notation.

We assume that the inputs $f : \mathbb{X} \rightarrow \mathbb{R}^d$ and $g : \mathbb{Y} \rightarrow \mathbb{R}^d$ are semi-algebraic maps defined on semi-algebraic sets in \mathbb{R}^k , as well as being compact as assumed earlier. Recall that the class of semi-algebraic sets is the smallest class of sets defined by a finite number of polynomial (in)equalities $\{x \in \mathbb{R}^k \mid p(x) \geq 0\}$ that is closed under complement, finite union, and finite intersection. A map $f : \mathbb{X} \rightarrow \mathbb{R}^d$ is semi-algebraic if the graph of f is a semi-algebraic set in $\mathbb{R}^k \times \mathbb{R}^d$. A semi-algebraic set is *semi-algebraically connected* if it cannot be written as the disjoint union of two non-empty open semi-algebraic sets. Analogously, a semi-algebraic set X is *path connected* if for any two points $x, x' \in X$ there is a continuous semi-algebraic map $\gamma : [0, 1] \rightarrow X$ such that $\gamma(0) = x'$ and $\gamma(1) = x$. Note that, by definition, a semi-algebraically path connected semi-algebraic set is also path connected. From this observation, combined with Theorem 2.4.5 and Proposition 2.5.13 in [39], we see that any connected semi-algebraic set is also path-connected. We make this restriction so that we have the following property.

Lemma 2.2. *Given a semi-algebraic map $f : \mathbb{X} \rightarrow \mathbb{R}^d$ with \mathbb{X} semi-algebraic, and any semi-algebraic open set $U \subset \mathbb{R}^d$, $\pi_0(f^{-1}(U))$ is finite.*

We refer the reader to [40] for an overview of the basic properties of semi-algebraic sets and maps, from which this lemma easily follows.

Because each grid cell itself is a semi-algebraic set, we have that for any open set $S \in \mathbf{Open}(\mathcal{U})$, the set of (path) connected components $\pi_0 f^{-1}(|S|)$ is finite.

Then we have a functor $F : \mathbf{Open}(\mathcal{U}) \rightarrow \mathbf{Set}$ given by

$$\begin{aligned} F : \mathbf{Open}(\mathcal{U}) &\rightarrow \mathbf{Set} \\ S &\mapsto \pi_0 f^{-1}(|S|). \end{aligned}$$

Functoriality of π_0 means that for $S \subseteq T$, there is an induced map

$$F[S \subseteq T] : \pi_0 f^{-1}(|S|) \rightarrow \pi_0 f^{-1}(|T|),$$

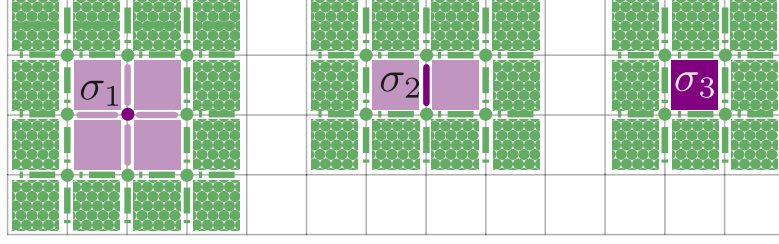


Fig. 3 Examples of open sets for $d = 2$. The basic open S_σ is given in light purple for each cell type shown in dark purple (vertex σ_1 , edge σ_2 and square σ_3). The 1-thickening of the open sets are given by including the green portion.

so that F satisfies the requirements of a functor. Indeed, this functor is actually a cosheaf so moving forward, we assume that F and G are cosheaves, even if they were not obtained from some input topological space. When the sets involved are obvious in the notation, we will write the induced map as $F[\subseteq] : F(S) \rightarrow F(T)$.

2.3 Thickenings

Given any set $S \in \mathbf{Open}(\mathcal{U})$, the 1-thickening² is defined by taking the downward closure of the upper closure of S , written as $S^1 = (S^\uparrow)^\downarrow$. This operation can be thought of as taking the star of the closure of the set; see Fig. 3 for examples. The n -thickening is defined to be n repetitions of the process given recursively as

$$S^n = \begin{cases} S & n = 0 \\ (S^{n-1})^\uparrow^\downarrow & n \geq 1. \end{cases}$$

Each S^n is itself an open set in $\mathbf{Open}(\mathcal{U})$, and if $S \subseteq T$, then $S^n \subseteq T^n$. Thus we can view this operation as a functor on the category $\mathbf{Open}(\mathcal{U})$ with morphisms given by inclusion:

$$\begin{array}{ccc} (-)^n : \mathbf{Open}(\mathcal{U}) & \rightarrow & \mathbf{Open}(\mathcal{U}) \\ S & \mapsto & S^n. \end{array}$$

In Sec. 6, we show that $(-)^n$ is a functor. Because of the cosheaf assumption, we also have that $F(S^n) = \varinjlim_{S_\sigma \subset S^n} F(S_\sigma)$. Another useful property of this thickening, proved in Sec. 6, is described in Lem. 2.3.

Lemma 2.3. *For any $n, n' \geq 0$ and $S \in \mathbf{Open}(\mathcal{U})$,*

$$(S^n)^{n'} = S^{n+n'}.$$

We can use this thickening to build an interleaving distance on cosheaves of the form $F : \mathbf{Open}(\mathcal{U}) \rightarrow \mathbf{Set}$. The first necessary ingredient is the composition of functors $F \circ (-)^n : \mathbf{Open}(\mathcal{U}) \rightarrow \mathbf{Set}$, which we denote by F^n . This means $F^n(S) = F(S^n)$, followed by a similar setup for G^n . With this notation, an interleaving is a pair of

²We note that this definition is distinct from the morphological operation of thickening that is a tool in image processing, although quite similar to the concept of dilation from that field.

natural transformations $\varphi : F \Rightarrow G^n$ and $\psi : G \Rightarrow F^n$, so a component of φ is a set-map $\varphi_S : F(S) \rightarrow G(S^n)$. There is another component at S^n , $\varphi_{S^n} : F(S^n) \rightarrow G(S^{2n})$, which can also be viewed as a component of a different natural transformation $\varphi^n : F^n \Rightarrow G^{2n}$. For this reason, we use the notation φ_{S^n} and φ_S^n interchangeably when φ is indeed a natural transformation.³

We are now ready to introduce our notion of interleaving distance.

Definition 2.4. *Given cosheaves $F, G : \mathbf{Open}(\mathcal{U}) \rightarrow \mathbf{Set}$ and $n \geq 0$, an n -interleaving is a pair of natural transformations $\varphi : F \Rightarrow G^n$ and $\psi : G \Rightarrow F^n$ such that the diagrams*

$$\begin{array}{ccc} F(S) & \xrightarrow{F[S \subseteq S^{2n}]} & F(S^{2n}) \\ \varphi_S \searrow & & \nearrow \psi_{S^n} \\ & G(S^n) & \end{array} \qquad \begin{array}{ccc} & F(S^n) & \\ \psi_S \nearrow & & \searrow \varphi_{S^n} \\ G(S) & \xrightarrow{G[S \subseteq S^{2n}]} & G(S^{2n}) \end{array}$$

commute for all $S \in \mathbf{Open}(\mathcal{U})$. The interleaving distance is given by

$$d_I(F, G) = \inf\{n \geq 0 \mid \text{there exists an } n\text{-interleaving}\},$$

and is set to be $d(F, G) = \infty$ if there is no interleaving for any n .

As shown in Sec. 6, this definition fits in the framework built by Bubenik et al. [16] and thus it is an extended pseudometric.

3 Loss Function and Bounds

In this section, we introduce a loss function for interleavings on \mathbb{R}^d -mapper complexes. We give the definition of the loss function (Defn. 3.4) in Sec. 3.1, and present our first version of the bound as Thm. 3.8 in Sec. 3.2. However, this version of the bound requires checking diagrams for all possible open sets $S \in \mathbf{Open}(K)$ which creates a combinatorial explosion that is counterproductive in practice. Thus, in Sec. 3.3, we prove this loss function can be replaced with an improved loss function which only needs to check the open sets for a basis of $\mathbf{Open}(\mathcal{U})$.

3.1 Loss Function Definition

We start by turning each non-empty $F(S)$ (similarly $G(S)$) into a metric space, as follows.

Definition 3.1. *Define the distance $d_S^F(A, B)$ for $A, B \in F(S)$ to be the smallest n such that A and B represent the same connected component when included into S^n . Specifically,*

$$d_S^F(A, B) = \min\{n \geq 0 \mid F[S \subset S^n](A) = F[S \subset S^n](B)\}.$$

If no such n exists, then we set $d_S^F(A, B) = \infty$.

³We are implicitly using Lem. 2.3 to write the maps this way.

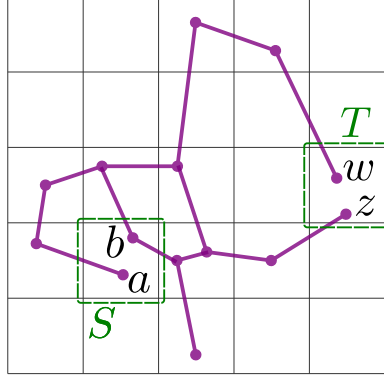


Fig. 4 An example geometric graph used to calculate example distances in d_S^F and d_T^F .

It is easy to see that this definition satisfies the definition of an extended metric. Indeed, it is actually an extended ultrametric since $d_S^F(A, C) \leq \max\{d_S^F(A, B), d_S^F(B, C)\}$, although we will not need that additional structure here.

Consider the example of Fig. 4 with a single input graph encoded by a cosheaf $F : \mathbf{Open}(\mathcal{U}) \rightarrow \mathbf{Set}$. The set $F(S)$ has two elements, which we denote by A and B as they represent the connected components containing the points a and b respectively. Then $d_S^F(A, B) = 1$, since thickening the set S by 1 puts a and b in the same connected component. Likewise, denoting the elements of $F(T)$ by W and Z , we see that $d_T^F(W, Z) = 2$ since we must expand the set T twice before w and z are in the same connected component.

As a first useful property of this distance, thickening a set implies that the distance between components will only decrease. For an example, consider $W, Z \in F(T)$ representing points w and z in Fig. 4. As noted previously, $d_T^F(W, Z) = 2$. However, if the elements $W', Z' \in F(T^1)$ represent the connected components in the 1-thickening of T , then $d_{T^1}^F(W', Z') = 1$, and in particular, $d_T^F(W, Z) \geq d_{T^1}^F(W', Z')$. This idea is formalized in the following lemma:

Lemma 3.2. *Fix $k \geq 0$ and any $A, B \in F(S)$ with images $A' = F[S \subseteq S^k](A)$ and $B' = F[S \subseteq S^k](B)$ in $F(S^k)$. Then*

$$d_{S^k}^F(A', B') = \max\{0, d_S^F(A, B) - k\} = \begin{cases} 0 & \text{if } k \geq n \\ d_S^F(A, B) - k & \text{if } 0 \leq k < n \end{cases}$$

and in particular, $d_S^F(A, B) \geq d_{S^k}^F(A', B')$.

Proof. Let $n = d_S^F(A, B)$, so that we know the image of A and B in $F(S^n)$ is the same. If $k \geq n$, then we use the functor maps $F(S) \rightarrow F(S^n) \rightarrow F(S^k)$ to see that the images of A and B are the same in $F(S^n)$ so they are the same in $F(S^k)$. Then $d_{S^k}^F(A', B') = 0$. If $k < n$, then we have the maps $F(S) \rightarrow F(S^k) \rightarrow F(S^n)$. Because we know that A and B do not map to the same thing prior to n , we have $d_{S^k}^F(A', B') = n - k$, completing the proof. \square

We use this framework as follows: first, assume we are given F and G but our attempts at finding an interleaving do not necessarily satisfy the requirements of a natural transformation. Normally, a natural transformation $\eta : H \Rightarrow H'$ is a collection of component morphisms $\eta : H(S) \rightarrow H'(S)$ which commute with the inclusions:

$$\begin{array}{ccc} H(S) & \xrightarrow{H[\subseteq]} & H(T) \\ \eta_S \downarrow & & \downarrow \eta_T \\ H'(S) & \xrightarrow{H'[\subseteq]} & H'(T). \end{array}$$

The following definitions, inspired by [34] and [41], give names to collections of component morphisms used to define an interleaving where the square might not commute.

Definition 3.3. *Given functors $H, H' : \mathbf{Open}(\mathcal{U}) \rightarrow \mathbf{Set}$, an unnatural transformation⁴ $\eta : H \rightarrow H'$ is a collection of maps $\eta_S : H(S) \rightarrow H'(S)$ with no additional promise of commutativity.*

For a fixed $n \geq 0$ and cosheaves F and G , an assignment, or more specifically an n -assignment, is a pair of unnatural transformations $\varphi : F \Rightarrow G^n$ and $\psi : G \Rightarrow F^n$.

In order to clarify notation, for the remainder of the paper, we will be using n -assignments to build $(n+k)$ -interleavings, which by definition will be required to be natural transformations. When the n -assignment might not commute, we denote its maps by lower case φ and ψ ; for $(n+k)$ -assignments which are constructed to be natural transformations, we denote them by Φ and Ψ .

In addition, we assume for the remainder of the paper that n is large enough for an assignment to exist. That is, it is possible that for some given $F(S)$, $G(S^n)$ might be empty for n small enough and thus there is no available map from one to the other. However, because we have assumed a compact input, $f(\mathbb{X})$ and $g(\mathbb{Y})$ is contained in a compact interval, and thus, we have that the σ for which $F(S_\sigma)$ is not empty is contained in some interval (in the poset sense). So long as n is large enough that the Hausdorff distance between the images $f(\mathbb{X})$ and $g(\mathbb{Y})$ is at most δn , $G(S^n)$ will be non-empty for any non-empty $F(S)$ (and vice versa).

In the spirit of [34], we measure the quality of a choice of an n -assignment φ, ψ using the collections of distances $\{d_S^F \mid S \in \mathbf{Open}(\mathcal{U})\}$ and $\{d_S^G \mid S \in \mathbf{Open}(\mathcal{U})\}$. First, note that checking that φ and ψ are natural transformations means ensuring the diagrams

$$\begin{array}{ccccc} F(S) & \xrightarrow{F[\subseteq]} & F(T) & & F(S^n) & \xrightarrow{F[\subseteq]} & F(T^n) \\ \varphi_S \searrow & & \varphi_T \searrow & & \psi_S \nearrow & & \psi_T \nearrow \\ & G(S^n) & \xrightarrow{G[\subseteq]} & G(T^n) & G(S) & \xrightarrow{G[\subseteq]} & G(T) \end{array}$$

⁴A natural transformation is an unnatural transformation which just happens to follow commutativity properties. In other words, natural and unnatural transformations are not mutually exclusive. This vocabulary follows from [41] so we accept no responsibility for the linguistic implications.

commute. As we use them repeatedly, we will denote these diagrams by $\nabla_\varphi(S, T)$ and $\triangleleft_\psi(S, T)$, dropping the subscript when it is clear from context. Then checking whether the pair constitutes an interleaving involves checking commutativity of the diagrams

$$\begin{array}{ccc}
 F(S) & \xrightarrow{F[S \subseteq S^{2n}]} & F(S^{2n}) \\
 \searrow \varphi_S & & \nearrow \psi_{S^n} \\
 & G(S^n) &
 \end{array}
 \qquad
 \begin{array}{ccc}
 & F(S^n) & \\
 \nearrow \psi_S & & \searrow \varphi_{S^n} \\
 G(S) & \xrightarrow{G[S \subseteq S^{2n}]} & G(S^{2n})
 \end{array}$$

which we denote by $\nabla_{\varphi, \psi}(S)$ and $\triangleleft_{\varphi, \psi}(S)$ respectively, again dropping the subscripts when unnecessary. We measure quality of the given assignments by checking how far these four diagrams are from commuting in the sense of the distances defined at the terminal vertex of the shape.

Definition 3.4. Fix an n -assignment (φ, ψ) . We define four diagram loss functions:

$$\begin{aligned}
 L_{\nabla}^{S, T}(\varphi) &= \max_{\alpha \in F(S)} d_{T^n}^G(\varphi_T \circ F[S \subseteq T](\alpha), G[S^n \subseteq T^n] \circ \varphi_S(\alpha)) \\
 L_{\triangleleft}^{S, T}(\psi) &= \max_{\alpha \in G(S)} d_{T^n}^F(\psi_T \circ G[S \subseteq T](\alpha), F[S^n \subseteq T^n] \circ \psi_S(\alpha)) \\
 L_{\nabla}^S(\varphi, \psi) &= \max_{\alpha \in F(S)} \left[\frac{1}{2} \cdot d_{S^{2n}}^F(F[S \subseteq S^{2n}](\alpha), \psi_{S^n} \circ \varphi_S(\alpha)) \right] \\
 L_{\triangleleft}^S(\varphi, \psi) &= \max_{\alpha \in G(S)} \left[\frac{1}{2} \cdot d_{S^{2n}}^G(G[S \subseteq S^{2n}](\alpha), \varphi_{S^n} \circ \psi_S(\alpha)) \right].
 \end{aligned}$$

Then the loss for the given assignment is defined to be

$$L(\varphi, \psi) = \max_{S \subseteq T} \left\{ L_{\nabla}^{S, T}, L_{\triangleleft}^{S, T}, L_{\nabla}^S, L_{\triangleleft}^S \right\}.$$

These loss functions are defined in a way so that while the diagram in question might not commute, pushing n forward by the loss value will send the elements to the same place. For example, if $L_{\nabla}^{S, T}(\varphi) = k$, then in the diagram

$$\begin{array}{ccccc}
 F(S) & \xrightarrow{F[\subseteq]} & F(T) & & \\
 \searrow \varphi_S & & \searrow \varphi_T & & \\
 & G(S^n) & \xrightarrow{G[\subseteq]} & G(T^n) & \longrightarrow G(T^{n+k})
 \end{array} \tag{3.5}$$

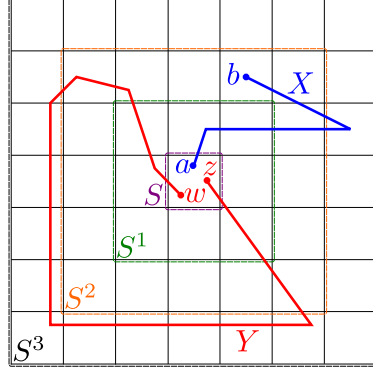


Fig. 5 An example of two input geometric graphs, X and Y .

the image of a point from $F(S)$ is the same in $G(T^{n+k})$ following both paths. Similarly, if $L_{\nabla}^S(\varphi, \psi) = k$, then in the diagram

$$\begin{array}{ccccc}
 F(S) & \xrightarrow{F[\subseteq]} & F(S^{2n}) & \longrightarrow & F(S^{2(n+k)}) \\
 \searrow \varphi_S & & \nearrow \psi_{S^n} & & \\
 & & G(S^n) & &
 \end{array} \tag{3.6}$$

the image of a point in $F(S)$ is the same (following both paths) in $F(S^{2(n+k)})$ even if not in $F(S^{2n})$.

An Example: Consider Fig. 5 and fix $n = 1$. Denote the connected component of the point a in $F(S)$, $F(S^1)$, and $F(S^2)$ by A , A' , and A'' , respectively. Similarly, the connected component of the point b is denoted by $B'' \in G(S^2)$. Follow the same form for the connected components of points w and z in G . The interleaving diagrams can be collected together as

$$\begin{array}{ccccc}
 \{A\} & \xrightarrow{F[S \subseteq S^1]} & \{A'\} & \xrightarrow{F[S^1 \subseteq S^2]} & \{A'', B''\} \\
 \searrow \varphi_S & & \nearrow \varphi_{S^n} & & \searrow \varphi_{S^1} \\
 \psi_S & & \psi_{S^1} & & \\
 \{W, Z\} & \xrightarrow{G[S \subseteq S^1]} & \{W', Z'\} & \xrightarrow{G[S^1 \subseteq S^2]} & \{W'', Z''\}
 \end{array} \tag{3.7}$$

noting that the horizontal maps are determined by sending a letter to the same letter with an additional prime. The distances between the points in their respective sets are

$$\begin{aligned}
 d_{S^2}^F(A'', B'') &= 1; \\
 d_S^G(W, Z) &= 3, \quad d_{S^1}^G(W', Z') = 2, \quad d_{S^2}^G(W'', Z'') = 1.
 \end{aligned}$$

Consider the following example assignment:

$$\begin{aligned}\varphi_S : A &\mapsto W', & \psi_S : W, Z &\mapsto A', \\ \varphi_{S^1} : A' &\mapsto W'', & \psi_{S^1} : W' &\mapsto A''.\end{aligned}$$

In this case, we then have that $L_{\sqcup}^{S, S^n} = 0$, $L_{\sqcup}^{S, S^n} = 1$, $L_{\sqcup}^S = 0$, and $L_{\sqcup}^S = 1$, so again $L(\varphi, \psi) \geq 1$. For this particular example, no $n = 1$ interleaving is possible so any choice of assignment will have a non-zero loss (the easiest check is to see that any choice of assignment will force $L_{\sqcup}^S = 1$).

3.2 Bounding the Interleaving Distance

We now use the loss function to give an upper bound for the interleaving distance.

Theorem 3.8. *For an n -assignment, $\varphi : F \Rightarrow G^n$ and $\psi : G \Rightarrow F^n$,*

$$d_I(F, G) \leq n + L(\varphi, \psi).$$

To prove this, we require the following technical lemma, proved in Sec. 6.

Lemma 3.9. *Assume we are given an n -assignment $\varphi : F \Rightarrow G^n$ and $\psi : G \Rightarrow F^n$. For a fixed k , define $(n+k)$ -assignments $\Phi_S = G[S^n \subseteq S^{n+k}] \circ \varphi_S$ and $\Psi_S = F[S^n \subseteq S^{n+k}] \circ \psi_S$ for all $S \in \mathbf{Open}(\mathcal{U})$. Then the following hold:*

1. $L_{\sqcup}^{S, T}(\varphi) \leq k$ implies $\sqcup_{\Phi}(S, T)$ commutes, and thus $L_{\sqcup}^{S, T}(\Phi) = 0$.
2. $L_{\sqcup}^{S, T}(\psi) \leq k$ implies $\sqcup_{\Psi}(S, T)$ commutes, and thus $L_{\sqcup}^{S, T}(\Psi) = 0$.
3. $L_{\sqcup}^S(\varphi, \psi) \leq k$ and $L_{\sqcup}^{S^n, S^{n+k}}(\psi) \leq k$ imply $\sqcup_{\Phi, \Psi}(S)$ commutes, and thus $L_{\sqcup}^S(\Phi, \Psi) = 0$.
4. $L_{\sqcup}^S(\varphi, \psi) \leq k$ and $L_{\sqcup}^{S^n, S^{n+k}}(\varphi) \leq k$ imply $\sqcup_{\Phi, \Psi}(S)$ commutes, and thus $L_{\sqcup}^S(\Phi, \Psi) = 0$.

In particular, if φ and ψ have $L(\varphi, \psi) = 0$, then φ and ψ constitute an interleaving, and so $d_I(F, G) \leq n$.

Proof of Thm. 3.8. Set $k = L(\varphi, \psi)$, so by definition, $L_{\sqcup}^{S, T}(\varphi) \leq k$, $L_{\sqcup}^{S, T}(\psi) \leq k$, $L_{\sqcup}^S(\varphi, \psi) \leq k$, and $L_{\sqcup}^S(\varphi, \psi) \leq k$. As in Lem. 3.9, construct two $(n+k)$ -assignments: Φ given by $\Phi_S = G[S^n \subseteq S^{n+k}] \circ \varphi_S$, and Ψ given by $\Psi_S = F[S^n \subseteq S^{n+k}] \circ \psi_S$. By Lem. 3.9, this means the diagrams $\sqcup_{\Phi}(S, T)$, $\sqcup_{\Psi}(S, T)$, $\sqcup_{\Phi, \Psi}(S)$, and $\sqcup_{\Phi, \Psi}(S)$ commute for all pairs $S \subseteq T$. This implies that Φ and Ψ are an $(n+k)$ -interleaving, giving the theorem. \square

First, notice that this proof works by explicitly constructing an interleaving from a given n -assignment. Second, we have no reason to believe that this bound is tight. In particular, in Sec. 3.3 we improve the bound by way of restricting the computation to the basis for the topology of K but even that is depending on input quality and gives no guarantee.

We include one additional note about when this loss function can be promised to be finite. Define the diameter of a metric space to be the largest distance between

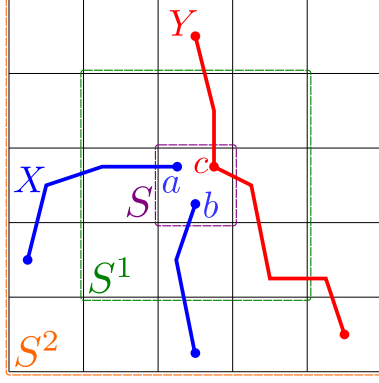


Fig. 6 An example of the comparison of two geometric graphs with different numbers of connected components. In this case, because X has one connected component and Y has two, the loss function will be infinite for any assignment.

points, which we denote by $\text{diam}(X, d) = \sup\{d(a, b) \mid a, b \in X\}$, and note that here, the sup can be replaced with a max since we are working in finite metric spaces. To simplify statements, we define the diameter of the empty set to be zero.

Lemma 3.10. *The loss function for an n -assignment (φ, ψ) is bounded above; specifically,*

$$L(\varphi, \psi) \leq \max \left(\left\{ \text{diam}(F(S^k), d_F^{S^k}) \mid S \in \mathbf{Open}(\mathcal{U}), k \in \{n, 2n\} \right\} \cup \left\{ \text{diam}(G(S^k), d_G^{S^k}) \mid S \in \mathbf{Open}(\mathcal{U}), k \in \{n, 2n\} \right\} \right).$$

In particular, if the inputs come from $f : \mathbb{X} \rightarrow \mathbb{R}$ and $g : \mathbb{Y} \rightarrow \mathbb{R}$ with both \mathbb{X} and \mathbb{Y} connected, then $L(\varphi, \psi)$ is finite.

Proof. The parallelogram portions of the loss function L_{∇} and L_{\triangleright} take values from distances in $F(S^n)$ and $G(S^n)$. The triangle portions L_{∇} and L_{\triangle} take values from distances in $F(S^{2n})$ and $G(S^{2n})$. So, the maximum for the loss function must be attained on one of these sets, giving the inequality. For the second statement, if the input graphs each have a single connected component, then any pair of elements $a, b \in F(S)$ map to the same element under the inclusion $F(S) \rightarrow F(S^K)$ for a large enough K . This in turn implies that the diameter of d_S^F is finite for every S . \square

Consider the example in Fig. 6. Let $\{A, B\}$, $\{A', B'\}$, and $\{A'', B''\}$ be the representatives of the connected components of the points a and b in $F(S)$, $F(S^1)$ and $F(S^2)$ respectively. Because there is no n for which the two points are the same connected component of X , the distance between A and B is ∞ in all three sets. Then no matter the choice of 1-assignment, $L_{\nabla} = \infty$, making the loss function infinite.

3.3 Restriction to Basis Elements

We have so far measured the loss function by studying all possible open sets S . While this is helpful for definitions, it does not make for a reasonable computational setting. To that end, we now focus on a basis of the topology, and prove that this basis suffices.

Recall that an open set $S_\sigma \in \mathbf{Open}(\mathcal{U})$ (Eqn. (2.1)) given by the downset of U_σ for some $\sigma \in K$ is called a *basic open set*. Note that $\{S_\sigma \mid \sigma \in K\}$ is a basis for the Alexandroff topology. We next give a name to the case where we are only given n -assignment information for basis elements, or equivalently, if we are given a full assignment but ignore the maps for non-basis open sets.

Definition 3.11. *A basis unnatural transformation for functors H and H' is a collection of maps $\eta_{S_\sigma} : H(S_\sigma) \rightarrow H'(S_\sigma)$ for all basis elements S_σ from $\sigma \in K$. A basis n -assignment (or simply a basis assignment) is a pair of basis unnatural transformations*

$$\{\varphi_{S_\sigma} : F(S_\sigma) \rightarrow G(S_\sigma^n) \mid \sigma \in K\} \quad \text{and} \quad \{\psi_{S_\sigma} : G(S_\sigma) \rightarrow F(S_\sigma^n) \mid \sigma \in K\}$$

In this section, we prove that we can focus our loss function efforts on only those diagrams associated to basic opens, and the solution can be extended to any open set.

Definition 3.12. *The basis loss function is defined to be*

$$L_B(\varphi, \psi) = \max_{\sigma \leq \tau} \left\{ L_{\sqcup}^{S_\tau, S_\sigma}, L_{\sqsupset}^{S_\tau, S_\sigma}, L_{\triangle}^{S_\sigma}, L_{\nabla}^{S_\sigma} \right\}.$$

It is immediate from the definitions that $L_B \leq L$ as the L_B maximum is taken over a subset of those used to determine L . This means, in particular, that if $L = 0$ then $L_B = 0$. These values are not always equal; for instance, we might have chosen a basis assignment for which every diagram commutes (making $L_B = 0$), but φ_T defined on non-basis elements causes a non-zero loss function so $L > 0$. However in the special case where $L_B = 0$, and thus the basis open diagrams are commutative, we do have the ability to extend the information checked to a full interleaving. This can be seen in the following lemma, proved in Sec. 6.

Lemma 3.13. *Given a basis unnatural transformation*

$$\{\Phi_{S_\sigma} : F(S_\sigma) \rightarrow G(S_\sigma^N) \mid \sigma \in K\}$$

with $L_{\sqcup}^{S_\tau, S_\sigma} = 0$ for all $\sigma \leq \tau$, we can extend this to a full natural transformation Φ ; i.e. we can define Φ_S for all S such that $L_{\sqcup}^{S, T} = 0$ for all $S \subseteq T$.

Note that the symmetric version extending a basis unnatural transformation Ψ to a natural transformation $\Psi : G \Rightarrow F^N$ is obtained in exactly the same way. Next, we can take these natural transformations and ensure the triangles commute (thus giving an interleaving) by only checking the basis set triangles, again proved in Sec. 6.

Lemma 3.14. *Given natural transformations $\Phi : F \Rightarrow G^N$ and $\Psi : G^N \Rightarrow F$ such that $L_{\sqcup}^{S_\sigma} = 0$ for all $\sigma \in K$, then $L_{\sqcup}^S = 0$ for all open sets S .*

Taken together, we immediately have the following proposition.

Proposition 3.15. *Fix a basis N -assignment (Φ, Ψ) . If $L_B(\Phi, \Psi) = 0$, then Φ and Ψ can be extended to natural transformations with $L(\Phi, \Psi) = 0$, and thus constitute an interleaving.*

Finally, we arrive at our main theorem, where we can use the provided basis n -assignment and the calculated loss function to give a bound for the interleaving distance.

Theorem 3.16. *Given a basis n -assignment*

$$\varphi = \{\varphi_{S_\sigma} \mid \sigma \in K\} \text{ and } \psi = \{\psi_{S_\sigma} \mid \sigma \in K\},$$

we have

$$d_I(F, G) \leq n + L_B(\varphi, \psi).$$

Proof. This proof proceeds in the same way as that of Thm. 3.8 with some minor modifications of input assumptions. First, let $k = L_B(\varphi, \psi)$; and define a basis $(n + k)$ -assignment by

$$\{\Phi_{S_\sigma} = G[\subseteq] \circ \varphi_{S_\sigma} \mid \sigma \in K\} \quad \text{and} \quad \{\Psi_{S_\sigma} = F[\subseteq] \circ \psi_{S_\sigma} \mid \sigma \in K\}.$$

By Lem. 3.9, we know that $L_{\nabla}^{S_\tau, S_\sigma}(\Phi) = 0$ and $L_{\nabla}^{S_\tau, S_\sigma}(\Psi) = 0$ for all $\tau \leq \sigma$. Then by Lem. 3.13, we can extend Φ and Ψ to full natural transformations defined for all $S \in \mathbf{Open}(\mathcal{U})$.

To show that Φ and Ψ constitute an $(n + k)$ -interleaving, we must check triangles; i.e. ensure that $L_{\nabla}^S(\Phi, \Psi) = L_{\Delta}^S(\Phi, \Psi) = 0$. With the goal of using part 3 of Lem. 3.9, first note that $L_{\nabla}^{S_\sigma}(\varphi, \psi) \leq k$ for basis elements. We can see that $L_{\nabla}^{S_\sigma^{n+k}} \leq k$ by using the (non-commutative) diagram

$$\begin{array}{ccccc} & & F(S_\sigma^{2n}) & \xrightarrow{F[\subseteq]} & F(S_\sigma^{2n+k}) & \xrightarrow{F[\subseteq]} & F(S_\sigma^{2(n+k)}) \\ & \nearrow \psi_\bullet & & & & & \\ G(S_\sigma^n) & \xrightarrow{G[\subseteq]} & G(S_\sigma^{n+k}) & & & & \\ & \nwarrow \psi_\bullet & & & & & \\ & & F(S_\sigma^{2n}) & \xrightarrow{F[\subseteq]} & F(S_\sigma^{2n+k}) & \xrightarrow{F[\subseteq]} & F(S_\sigma^{2(n+k)}) \end{array}$$

(Note: The diagram above is a simplified representation of the commutative diagram in the image. The image shows a more complex diagram with multiple paths and labels Ψ_\bullet and ψ_\bullet on the arrows.)

The leftmost and rightmost triangles commute by definition of Ψ , and the orange parallelogram commutes because Ψ is a natural transformation. Then chasing any $x \in G(S_\sigma^n)$ up to the top right $F(S_\sigma^{2(n+k)})$ results in the same element, giving the required bound on $L_{\nabla}^{S_\sigma^{n+k}}(\varphi, \psi)$. Using Lem. 3.14 for Φ and Ψ , $L_{\nabla}^S(\Phi, \Psi) = 0$ for all open sets S . The proof that $L_{\Delta}^S(\Phi, \Psi) = 0$ is similar. Therefore Φ and Ψ are an $(n + k)$ -interleaving, giving the bound. \square

What is surprising about this bound is that despite checking fewer open sets, the loss function for L_B is actually lower than that found using L . One reason for this is that when we work with the smaller set of input maps, we extend the collection to a “better” full assignment, potentially getting rid of some of the causes of a nonzero loss function in the first place. For example, a full assignment would be required to

provide a map φ_S for a S with multiple connected components, say $S = T_1 \cup T_2$. Since no requirements were made of this map based on the φ_{T_1} and φ_{T_2} maps, there is a reasonable chance that the loss function contribution from the $L_{\sqcup}^{T_1, S}$ is higher than necessary. However, in the basis version, we can build the best possible φ_T given the information over φ_{S_1} and φ_{S_2} , providing a potentially better, but certainly no worse, bound.

4 Computation

In this section, we show that given an n -assignment φ and ψ , we can compute the loss function $L_B(\varphi, \psi)$ in polynomial time. For simplicity, we describe the algorithm explicitly in the case where $d = 1$ for clarity of exposition, before addressing the run time in higher dimensions.

4.1 Data Structures for $d = 1$

In this section, we build a pair of graphs representing a pair of input functors F, G , and use pointers between the vertex and edge sets to represent a given n -assignment φ and ψ . For ease of exposition, we will carefully focus on the case $d = 1$, following the example of Fig. 7 to illustrate our construction. While we acknowledge an overuse of the term “graph” throughout this paper as it is used in many different contexts, for this section we use the term graph to mean a network; i.e. a combinatorial pair consisting of vertices and edges. Throughout, we denote edges interchangeably by either (x, y) or xy depending on the context and notational complexity.

At a high level, we will construct graphs for F and G , which we denote by (V_F, E_F) and (V_G, E_G) . Then we will build data structures to encode the natural transformations φ and ψ . For clarity, we use **phi** and **psi** to denote a collection of pointers that will store the information of φ and ψ , respectively. These will be viewed as set maps⁵ **phi** : $(V_F, E_F) \rightarrow (V_G, E_G)$ and **psi** : $(V_G, E_G) \rightarrow (V_F, E_F)$, which map each vertex to a vertex in the other graph and each edge to an edge in the other graph. We give explicit constructions of **phi** and **psi** as well as further restrictions on allowable maps in what follows.

Focusing on $d = 1$, the complex K , i.e., the discretization of \mathbb{R} , consists of vertices which we write as $\sigma_{-L}, \dots, \sigma_L$ with heights in our bounding box $[-L\delta, L\delta]$, and with edges $\tau_j = (\sigma_j, \sigma_{j+1})$. Then we construct the graph for $F : \mathbf{Open}(\mathcal{U}) \rightarrow \mathbf{Set}$ by generating a vertex for every object in every $F(S_{\sigma_i})$ and connect them using the morphisms of the functor. This results in a vertex set

$$V_F = \prod_{i=1}^B F(S_{\sigma_i}),$$

⁵This notation is meant to imply that **phi** is composed of two maps, **phi** : $V_F \rightarrow V_G$ and **phi** : $E_F \rightarrow E_G$.

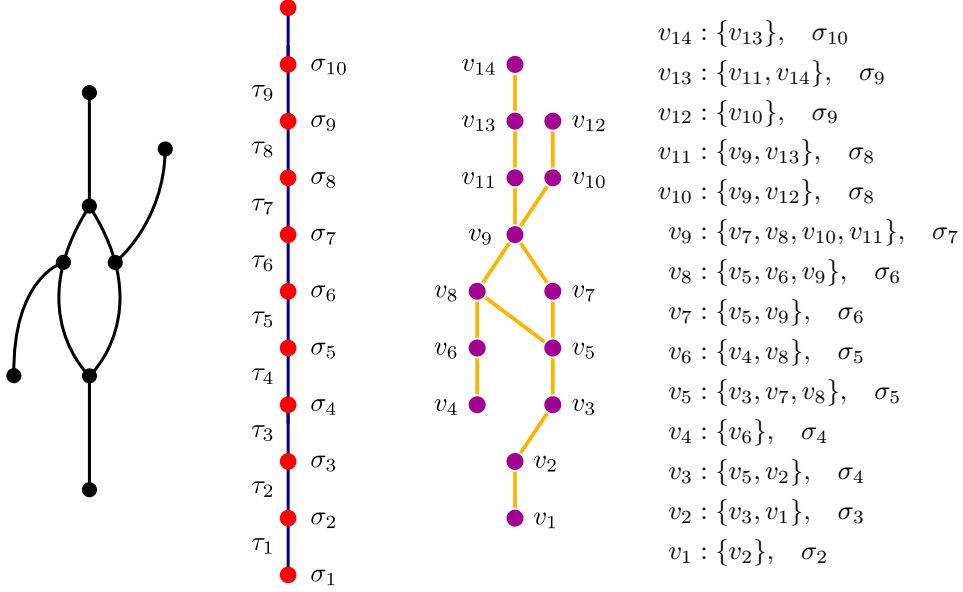


Fig. 7 From left to right: an example input Reeb graph ($d = 1$), a discretization of \mathbb{R} , the generated mapper graph, and the data structure encoding the mapper graph.

and an edge for every object in every $F(S_{\tau_i})$, giving edge set

$$E_F = \prod_{i=1}^{B-1} F(S_{\tau_i}).$$

The endpoints of any edge $e \in F(S_{\tau_i}) \subseteq E_F$ can be found via the attaching maps:

$$\begin{aligned} F[S_{\tau_i} \subseteq S_{\sigma_i}](e) &\in F(S_{\sigma_i}) \text{ and} \\ F[S_{\tau_i} \subseteq S_{\sigma_{i+1}}](e) &\in F(S_{\sigma_{i+1}}). \end{aligned}$$

For example, $e = (v_4, v_6) \in F(S_{\tau_4})$ in Fig. 7 has endpoints $v_6 \in F(S_{\sigma_4})$ and $v_4 \in F(S_{\sigma_5})$. We store this data in a standard adjacency list. In addition, each vertex also keeps track of its height, so a vertex $v \in F(S_{\sigma_i})$ also stores the value i as a representation of its height.

Next, we encode the information for an assignment (φ, ψ) between $F, G : \mathbf{Open}(\mathcal{U}) \rightarrow \mathbf{Set}$ by constructing the set maps **phi** and **psi** using the graphs (V_F, E_F) and (V_G, E_G) . Assume we are given n -assignments φ and ψ . We start by focusing on the vertex set. For this, we need to represent the map $\varphi_{S_{\sigma_i}} : F(S_{\sigma_i}) \rightarrow G(S_{\sigma_i}^n)$. The elements of $F(S_{\sigma_i})$ are already given as vertices, however the elements of $G(S_{\sigma_i}^n)$ are not. But, because of the cosheaf structure of G , the elements of $G(S_{\sigma_i}^n)$ can be seen as the connected components of particular subsets of the (V_G, E_G) .

We emphasize that we are using the term “subset” because the resulting objects will not be subgraphs in the usual sense. These subsets will consist of a subset of the vertex set and a subset of the edge set (V', E') , but without the promise that both endpoints of an edge are included in the set. However, we can still define connected components in this setting to consist of vertices and edges which can be connected by paths.

For some vertex σ_i of K , let

$$V_{G, \sigma_i, n} = \{v \mid v \in G(S_{\sigma_j}), j \in [i - n, i + n]\}$$

and

$$E_{G, \sigma_i, n} = \{e \mid e \in G(S_{\tau_j}), j \in [i - n - 1, i + n]\}$$

and write $(V_G, E_G)_{\sigma_i, n} := (V_{G, \sigma_i, n}, E_{G, \sigma_i, n})$. Similarly for the edges of K , we can define

$$V_{G, \tau_i, n} = \{v \mid v \in G(S_{\sigma_j}), j \in [i - n + 1, i + n]\}$$

and

$$E_{G, \tau_i, n} = \{e \mid e \in G(S_{\tau_j}), j \in [i - n, i + n]\}$$

with notation $(V_G, E_G)_{\tau_i, n} := (V_{G, \tau_i, n}, E_{G, \tau_i, n})$. Note that because of the endpoints, these are not induced subgraphs; see Fig. 8 for examples. We call either $(V_G, E_G)_{\tau_i, n}$ or $(V_G, E_G)_{\sigma_i, n}$ a *slice* of the graph since they are each a portion of the graph which gives the connected components over an open interval as seen in the following lemma.

Lemma 4.1. *The elements of $G(S_{\sigma_i}^n)$ (resp. $G(S_{\tau_i}^n)$) are in one-to-one correspondence with the connected components of the subset of the graph $(V_G, E_G)_{\sigma_i, n}$ (resp. $(V_G, E_G)_{\tau_i, n}$).*

Proof. This lemma is immediate from noting that $G(S_{\sigma_i}^n)$ is the colimit of the diagram given by $G(S_\sigma)$ for $S_\sigma \subset S_{\sigma_i}^n$ where σ is taken over all cells in K (of both dimension 0 and 1), and then carefully tracking indices of these cells from the above notation. The edge version is similar. \square

With this, we can return to storing a given n -assignment φ and ψ . To store the unnatural transformation φ , for each $v \in F(S_{\sigma_i})$, we choose a vertex $\mathbf{phi}(v) \in V_{G, \sigma_i, n}$, where $\mathbf{phi}(v)$ is in the connected component of $(V_G, E_G)_{\sigma_i, n}$ represented by $\varphi_{S_{\sigma_i}}(v) \in G(S_{\sigma_i}^n)$. We note that given a collection of choices of vertices

$$\{\mathbf{phi}(v) \in G(S_{\sigma_j}) \subset V_G \mid v \in F(S_{\sigma_i}) \subset V_F, |i - j| \leq n\}$$

and edges

$$\{\mathbf{phi}(e) \in G(S_{\tau_j}) \mid e \in F(S_{\tau_i}) \subset V_F, |i - j| \leq n\}$$

we can immediately reconstruct an unnatural transformation φ by setting $\varphi_{S_{\sigma_i}}(v)$ to be the element of $G(S_{\sigma_i}^n)$ representing the connected component of $(V_G, E_G)_{\sigma_i, n}$ containing $\mathbf{phi}(v)$. As these processes are inverse of each other and the parallel setup can be done for ψ and \mathbf{psi} , we are justified in using this data structure to represent the assignment.

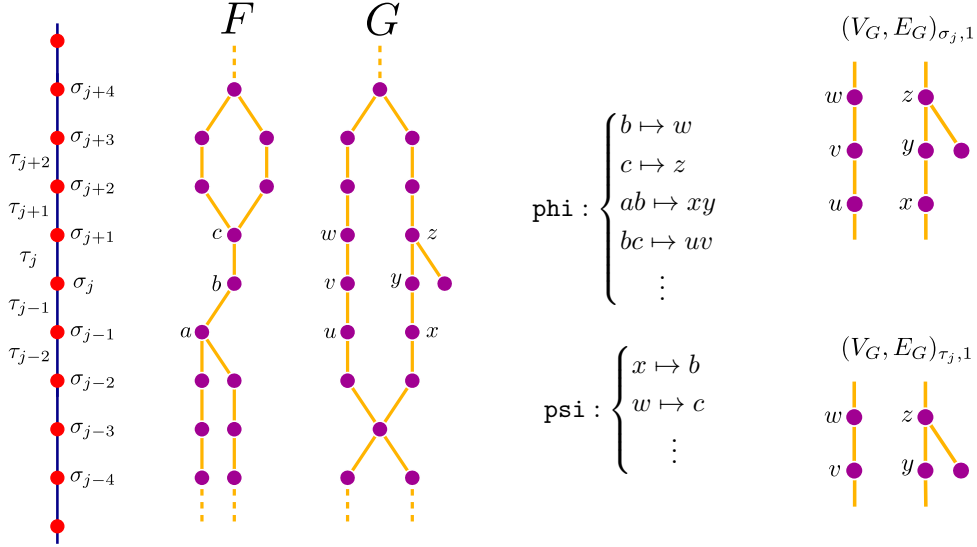


Fig. 8 At left are two example input mapper graphs. In the middle are a subset of an example **phi** and **psi** input assignment used throughout the text. At right are two example slices of the graphs used when checking connectivity.

For an example, consider Fig. 8 where we assume $n = 1$. If $\mathbf{phi}(b) = w$, then $\varphi_{S_{\sigma_i}}(b)$ is the connected component that includes w of $(V_G, E_G)_{\sigma_i, 1}$ as shown on the right. We can similarly find the edge map $\mathbf{phi}(e)$ for $e \in F(S_{\tau_i})$ by setting it to be an edge in $E_{G, \tau_i, n}$ representing the connected component of $\varphi_{S_{\tau_i}}(e) \in G(S_{\tau_i}^n)$ in $(V_G, E_G)_{\tau_i, n}$. So, for example, in Fig. 8 where $n = 1$, the input data might have $\mathbf{phi}(ab) = (xy) \in E_G$ and $\mathbf{phi}(bc) = (uv) \in E_G$.

4.2 Algorithm and Complexity for $d = 1$

In this section, we discuss the complexity of determining $L_B(\varphi, \psi)$ given **phi** and **psi**. First, we will proceed using a binary search on $k \in [0, \dots, 2L]$ where the maximum is determined by the diameter of the bounding box. For a fixed k , we will determine if $L_B(\varphi, \psi) \leq k$ by checking if $L_{\nabla}^{S_{\tau}, S_{\sigma}}, L_{\nabla}^{S_{\sigma}}, L_{\nabla}^{S_{\tau}, S_{\sigma}}$ and $L_{\Delta}^{S_{\sigma}}$ are all less than k for all σ and τ in K . We will describe the cases for $L_{\nabla}^{S_{\tau}, S_{\sigma}}$ and $L_{\nabla}^{S_{\sigma}}$, as $L_{\nabla}^{S_{\tau}, S_{\sigma}}$ and $L_{\Delta}^{S_{\sigma}}$ are symmetric.

Start with $L_{\nabla}^{S_{\tau}, S_{\sigma}}$ and note that in the case where $d = 1$, there are two pairs necessary to check for each edge: τ_j, σ_j and τ_j, σ_{j+1} . Fix σ_ℓ to be either σ_j or σ_{j+1} . For each edge $e \in F(S_{\tau_j})$, we need to check if the two possible images in $G(S_{\sigma_\ell}^{n+k})$

under the diagram

$$\begin{array}{ccccc}
F(S_{\tau_i}) & \xrightarrow{F[\sqsubseteq]} & F(S_{\sigma_\ell}) & & e \xrightarrow{\quad} v \\
& \searrow \varphi_{S_{\tau_i}} & \searrow \varphi_{S_{\tau_i}^n} & & \swarrow \quad \searrow \\
& & G(S_{\tau_i}^n) & \xrightarrow{G[\sqsubseteq]} & G(S_{\sigma_\ell}^n) \longrightarrow G(S_{\sigma_\ell}^{n+k}) \\
& & & & [e'] \xrightarrow{\quad} [w] \rightleftharpoons [w] \\
& & & & [e'] \xrightarrow{\quad} [e']
\end{array} \tag{4.2}$$

are the same. Note that we use $[-]$ to represent the connected component in the relevant slice of the graph containing that edge or vertex. Following the top of diagram Eq. (4.2), we know that e has a unique endpoint vertex $v \in F(S_{\sigma_\ell})$, and that vertex has an image under $\varphi_{S_{\tau_i}^n}$, which is a connected component represented by $\mathbf{phi}(v) = w \in V_G$. Following down, the edge e has an edge image $\mathbf{phi}(e) = e' \in E_G$. So the question becomes: are e' and w in the same connected component of the slice $(V_G, E_G)_{\sigma_\ell, n+k}$, whose components represent the elements of $G(S_{\sigma_\ell}^{n+k})$? In order to answer this question easily for all starting edges in $F(S_{\tau_i})$, we use a breadth first search algorithm to label all the connected components of the slice once. Then the two images of each edge starting from $F(S_{\tau_i})$ can be checked in constant time [42, Sec. 5.6]. This results in a total time (when $d = 1$) of $O(|V_G| + |E_G|)$ time taken for checking the parallelogram. There are $2L$ dimension 1 cells in K , and after computing connected components for the two parallelogram diagrams, all edges in a cell can have each parallelogram diagrams checked in $O(1)$ time, thus the time to determine if $\max_{\sigma \leq \tau} L_{\nabla}^{S_\tau, S_\sigma} \leq k$ is $O(L \cdot (|V_G| + |E_G|))$.

In the example of Fig. 8, assume $n = k = 1$ and assume the given input \mathbf{phi} is as noted. Then for the diagram of Eq. (4.2) with $\ell = j$ and chasing $bc \in F(S_{\tau_j})$, this comes down to checking if the connected component of $\mathbf{phi}(b) = w$ and $\mathbf{phi}(bc) = xy$ are the same in the portion of $(V_G, E_G)_{\sigma_j, 2}$. In this particular example, there are two connected components in this slice and the images are not in the same component. Then we know that $L_{\nabla}^{S_{\tau_j}, S_{\sigma_j}} > k$ so we would skip all other commutative diagram checks and immediately move on in our binary search. If it were the case that the two images were in the same connected component, then $L_{\nabla}^{S_{\tau_j}, S_{\sigma_j}} \leq k$ and thus we would move on to the next commutative diagram check.

Checking if $L_{\nabla}^{S_\tau} \leq k$ is similar so we briefly highlight the differences. First, there are two types of basis elements in our case where $d = 1$, so we need to check $L_{\nabla}^{S_{\sigma_i}} \leq k$ (meaning checking vertices) and $L_{\nabla}^{S_{\tau_i}} \leq k$ (meaning checking edges). We focus on the case of vertices since the edge version is similar. For any vertex $v \in F(S_{\sigma_i})$, we need

to chase it around the diagram

$$\begin{array}{ccccc}
F(S_{\sigma_i}) & \xrightarrow{F[S_{\sigma_i} \subseteq S_{\sigma_i}^{2n}]} & F(S_{\sigma_i}^{2n}) & \longrightarrow & F(S_{\sigma_i}^{2(n+k)}). \\
\searrow \varphi_{S_{\sigma_i}} & & \nearrow \psi_{S_{\sigma_i}^{2n}} & & \\
& G(S_{\sigma_i}^n) & & & \\
& \swarrow & \searrow & & \\
& v & & & w
\end{array}
\quad \begin{array}{c}
\begin{array}{ccc}
\begin{array}{c} [v] \\ [v'] \end{array} & \xleftrightarrow{\quad} & \begin{array}{c} [v] \\ [v'] \end{array} \\
\end{array}
\end{array} \quad (4.3)$$

If $\mathbf{phi} : v \mapsto w$, and $\mathbf{psi} : w \mapsto v'$, the question again becomes: are v and v' in the same connected component of $(V_G, E_G)_{\sigma_j, 2(n+k)}$? Similar to the parallelogram case, we take the relevant slice of the graph and check this connectivity question by finding connected components once in the slice in $O(|V_G| + |E_G|)$ time, and then the check for each vertex in $F(S_{\sigma_i})$ is done in $O(1)$ time. As before, either the elements checked are in the same connected component of the relevant slice of the graph, in which case we move to the next diagram; or it does not, and we move to a different k in our binary search. There are $O(L)$ cells (counting both 0- and 1-dimensional cells) in K , meaning there are $O(L)$ triangle diagrams to check. Thus, checking if $\max_{\sigma \in K} L_{\nabla}^{S_{\sigma}} \leq k$ can also be done in $O(L \cdot (|V_G| + |E_G|))$ time.

In our example case of Fig. 8 with $n = k = 1$, we have $2(n + k) = 4$. Then chasing b , we need to check that b and $\mathbf{psi} \circ \mathbf{phi}(b) = c$ are in the same connected component of $(V_G, E_G)_{\sigma_j, 4}$. As this slice has one connected component, this triangle commutes.

We can check another triangle $L_{\Delta}^{S_{\sigma_j}} \leq k$ by chasing w . In this case, we must check if w and $\mathbf{phi} \circ \mathbf{psi}(w) = z$ are in the same component of $(V_G, E_G)_{\sigma_i, 4}$, which again, they both are. In either case, if they were not, we would know the loss function is at least k and continue in the binary search.

Putting this together, this means that if the graph representations of F and G are (V_F, E_F) and (V_G, E_G) respectively, the time for computing the loss function is

$$O\left(L \log L \cdot \max\left\{|V_F| + |E_F|, |V_G| + |E_G|\right\}\right)$$

where the $\log L$ term comes from the binary search.

4.3 Generalization for $d > 1$

The generalization to higher dimensions makes relatively minor modifications for the algorithm, with the expected curse of dimensionality result in the running time. In this case, we build a graph with vertex set $V_F = \coprod_{\sigma \in K} F(S_{\sigma})$ so that vertices represent all dimensional cubes rather than only 0 as earlier. An edge is given between every pair of vertices v and w for which $F[S_{\tau} \subseteq S_{\sigma}](v) = w$. Denote the sizes of these sets by $|V_F|$ and $|E_F|$. Note that these sizes are in some sense already hiding an exponential term in d since the number of cells in the grid K is $O(L^d)$.

As in $d = 1$, we proceed using a binary search on $[0, \dots, 2L]$ where $[-L\delta, L\delta]^d$ is the bounding box of the images of $f : \mathbb{X} \rightarrow \mathbb{R}^d$ and $g : \mathbb{Y} \rightarrow \mathbb{R}^d$. Again, for a fixed k , we will determine if $L_B(\varphi, \psi) \leq k$ by checking if $L_{\nabla}^{S_\tau, S_\sigma}$, $L_{\nabla}^{S_\sigma}$, $L_{\nabla}^{S_\tau, S_\sigma}$ and $L_{\nabla}^{S_\sigma}$ are all less than k for all σ and τ in K .

If we count in terms of the open sets S_σ , every set $F(S_\sigma)$ needs to be checked as the starting point for one triangle diagram $\nabla_{\Phi, \Psi}(S_\sigma)$, and as the starting point for one parallelogram diagram $\nabla_{\Phi}(S_\sigma, S_\tau)$ for every $\tau \geq \sigma$. The grid structure means there are worst case $O(2^d)$ adjacent cells, so this results in $1 + O(2^d)$ diagram checks to be done per vertex. Each of these checks involves determining if two vertices are in the same connected component of the higher dimensional analogue of a slice for $\sigma_{\vec{\ell}}$ with indices $\vec{\ell} \in \mathbb{Z}^d$ involves checking a portion of the graph with indices in a d -dimensional box $[\ell_1 - (n+k), \ell_1 + (n+k)] \times \dots \times [\ell_d - (n+k), \ell_d + (n+k)]$ and hence takes worst case $O(|V_F| + |E_F|)$ time. However as before, this connected component needs to only be found once per diagram. The result is a running time of

$$O(\log L \cdot 2^d \cdot \max\{|V_F| + |E_F|, |V_G| + |E_G|\}).$$

5 Extension to Reeb Graphs

We now take a brief diversion into understanding how the loss function framework can be used to approximate the Reeb graph interleaving distance. In this case, we consider a 1-dimensional mapper graph to be an approximation of the Reeb graph [27–31]. We show that in order to bound the Reeb graph interleaving distance, we can compute the mapper graph for a resolution δ , and then use the loss function to provide a similar bound.

5.1 Definitions

Given input data $f : \mathbb{X} \rightarrow \mathbb{R}$, the Reeb graph of (\mathbb{X}, f) is computed as follows. Define an equivalence relation by setting $x \sim y$ iff x and y are in the same path-connected component of the levelset $f^{-1}(a)$. With enough restrictions on the space and function (for example, a Morse function on a manifold), the resulting Reeb graph is a topological graph; i.e. a 1-dimensional stratified space. Similar to the vantage taken for the mapper graphs in this paper, the data of a Reeb graph can be stored in a cosheaf.

Definition 5.1. *For a given (\mathbb{X}, f) , the associated Reeb cosheaf is given by*

$$\begin{array}{ccc} \tilde{F} : \mathbf{Int} & \rightarrow & \mathbf{Set} \\ I & \mapsto & \pi_0 f^{-1}(I) \\ \downarrow \cap & & \downarrow \pi_0 [\subseteq] \\ J & \mapsto & \pi_0 f^{-1}(J) \end{array}$$

where morphisms are induced by the π_0 functor.

For clarity, we write the Reeb cosheaf with a tilde to distinguish it from the mapper cosheaf without a tilde. Given this input, we have the Reeb graph interleaving distance [33], given as follows.

Definition 5.2. Define the functor $(-)^{\varepsilon} : \mathbf{Int} \rightarrow \mathbf{Int}$ by $(a, b) \mapsto (a - \varepsilon, b + \varepsilon)$ with morphisms induced by inclusion. Then $\tilde{F}_{\varepsilon} : \mathbf{Int} \rightarrow \mathbf{Set}$ is given by $\tilde{F}_{\varepsilon}(J) = \tilde{F}(J^{\varepsilon})$.

For given $\tilde{F}, \tilde{G} : \mathbf{Int} \rightarrow \mathbf{Set}$, an ε -interleaving is a pair of natural transformations $\tilde{\varphi} : \tilde{F} \Rightarrow \tilde{G}_{\varepsilon}$ and $\tilde{\psi} : \tilde{G} \Rightarrow \tilde{F}_{\varepsilon}$ such that

$$\begin{array}{ccc} \tilde{F}(I) & \xrightarrow{\tilde{F}[I \subseteq I^{2n}]} & \tilde{F}(I^{2n}) \\ \tilde{\varphi}_I \searrow & & \nearrow \tilde{\psi}_{I^n} \\ & \tilde{G}(I^n) & \end{array} \quad \begin{array}{ccc} & \tilde{F}(I^n) & \\ \tilde{\psi}_I \nearrow & & \searrow \tilde{\varphi}_{I^n} \\ \tilde{G}(I) & \xrightarrow{\tilde{G}[I \subseteq I^{2n}]} & \tilde{G}(I^{2n}) \end{array}$$

commute for all $I \in \mathbf{Int}$. The (categorical) Reeb graph interleaving distance is given by

$$d_R(\tilde{F}, \tilde{G}) = \inf\{\varepsilon \geq 0 \mid \text{there exists an } \varepsilon\text{-interleaving}\}.$$

Fix a δ . Following Sec. 2, denote the vertices of K by $\{\sigma_{-L}, \dots, \sigma_L\}$ where σ_i is at the point $i\delta \in \mathbb{R}$. Denote the edges by $\tau_i = (i\delta, (i+1)\delta)$ which has faces σ_i and σ_{i+1} . Given some input data $f : \mathbb{X} \rightarrow \mathbb{R}$, we can either construct its Reeb cosheaf $\tilde{F} : \mathbf{Int} \rightarrow \mathbf{Set}$, or by fixing some choice of δ , we can construct its mapper cosheaf $F : \mathbf{Open}(K) \rightarrow \mathbf{Set}$, $F(S) = f^{-1}(|S|)$.

We next show that the loss function we have computed here on the mapper version d_I can be used to similarly bound the Reeb interleaving distance d_R . We do this by showing that d_I is an approximation of d_R , which can be viewed as a special case of [31, Thm. 5.15]; however, for clarity, we include a direct proof in Sec. 6 as our setting allows a proof with considerably less use of category theoretic machinery.

Proposition 5.3. For inputs $f : \mathbb{X} \rightarrow \mathbb{R}$ and $g : \mathbb{Y} \rightarrow \mathbb{R}$, denote the respective Reeb cosheaves as $\tilde{F}, \tilde{G} : \mathbf{Int} \rightarrow \mathbf{Set}$, and the respective mapper cosheaves as $F, G : \mathbf{Open}(K) \rightarrow \mathbf{Set}$. Then

$$d_R(\tilde{F}, \tilde{G}) \leq (d_I(F, G) + 1) \delta.$$

Given this bound, we combine Prop. 5.3 with Thm. 3.16 to show that the loss function for the mapper graph discretization bounds the Reeb graph interleaving as well and that, in particular, this bound is controlled by the diameter δ chosen for K .

Corollary 5.4. Given a basis n -assignment $\varphi = \{\varphi_{U_{\sigma}} \mid \sigma \in K\}$ and $\psi = \{\psi_{U_{\sigma}} \mid \sigma \in K\}$ for $F, G : \mathbf{Open}(K) \rightarrow \mathbf{Set}$, we have that

$$d_R(\tilde{F}, \tilde{G}) \leq \delta(d_I(F, G) + 1) \leq \delta(n + L_B(\varphi, \psi) + 1).$$

6 Technical Proofs

In this section, we include the technical proofs from the previous sections.

6.1 Proofs from Sec. 2

Lemma 6.1. $(-)^n$ is a functor.

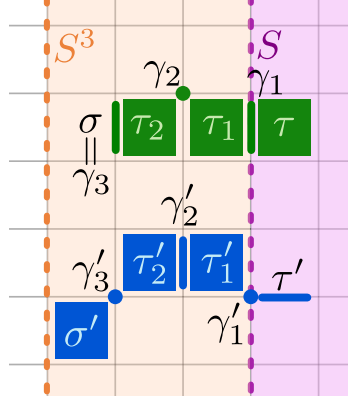


Fig. 9 Given the purple set U , we have an edge σ and a square σ' which are elements of U^3 . Then we provide an example sequence for each satisfying Eq. (6.2) leading to τ and τ' in U .

Proof. First, we check that the images of morphisms are well defined, which is to say that if $S \subseteq T$, then $S^n \subseteq T^n$. The statement is clear if $n = 0$, so by induction, we assume that $S^{n-1} \subseteq T^{n-1}$. Given an arbitrary $U_\sigma \in S^n$, the statement is immediate if $U_\sigma \in S^{n-1} \subseteq S^n$, so we assume $U_\sigma \in S^n \setminus S^{n-1}$. For this to happen, there must be a $U_\gamma \in S^{n-1}$ and $\tau \in K$ with $\gamma \geq \tau \leq \sigma$ and thus $U_\gamma \subseteq U_\tau \subseteq U_\sigma$. But as $U_\gamma \in S^{n-1} \subseteq T^{n-1}$, this sequence also implies that $U_\sigma \in T^n$, finishing the well-defined check.

To ensure this is a functor, we need to check that the identity morphism is sent to the identity, and that composition holds. For the former, we see that $S \subseteq S$ gets sent to $S^n \subseteq S^n$, and each is an identity. The latter is immediate from the property that $\mathbf{Open}(\mathcal{U})$ is a poset category, meaning that there is at most one morphism between any pair of objects. \square

One property of this construction that will be useful is as follows. For any $U_\sigma \in S^n$, there is a $U_\tau \in S$ and a sequence of cells of K

$$\tau \geq \gamma_1 \leq \tau_1 \geq \gamma_2 \leq \tau_2 \geq \cdots \geq \gamma_n \leq \sigma \quad (6.2)$$

and thus also a sequence of sets in \mathcal{U}

$$U_\tau \geq U_{\gamma_1} \leq U_{\tau_1} \geq U_{\gamma_2} \leq U_{\tau_2} \geq \cdots \geq U_{\gamma_n} \leq U_\sigma. \quad (6.3)$$

Further, given such a sequence with $\tau \in U$, we know that $\sigma \in U^n$. Two examples of this can be seen in Fig. 9, where σ and σ' from U^3 are given, along with a path satisfying Eq. (6.2). Of course, the choice of sequence for Eq. (6.2) is not unique, so other options are possible.

Next we show that the distance of Eq. (2.4) is indeed a distance using the super-linear family of translations framework of [16]. This construction can be generalized to the concept of a category with a flow [14], but the added generality is not needed here. **Definition 6.4** ([16]). *Let $P = (P, \leq)$ be a preordered set. A translation on P is a functor $\Gamma : P \rightarrow P$ along with a natural transformation $\eta : 1_P \Rightarrow \Gamma$. A super-linear*

family of translations is a collection $\{\Gamma_\varepsilon\}_{\varepsilon \geq 0}$ such that $\Gamma_\varepsilon \Gamma_{\varepsilon'}(p) \leq \Gamma_{\varepsilon+\varepsilon'}(p)$ for all $p \in P$, and $\varepsilon, \varepsilon' \geq 0$.

Lemma 2.3. For any $n, n' \geq 0$ and $S \in \mathbf{Open}(\mathcal{U})$,

$$(S^n)^{n'} = S^{n+n'}.$$

Proof. First, we check that $(-)^n$ is indeed a translation using the above terminology. In particular, we define $\gamma^n : \mathbf{1}_{\mathbf{Open}(\mathcal{U})} \Rightarrow (-)^n$ to have components $\gamma_S^n : S \rightarrow S^n$ as simply the inclusion, and we can easily check that this satisfies the naturality requirements.

Fix $S \in \mathbf{Open}(\mathcal{U})$. We need to show that $(S^n)^{n'} = S^{n+n'}$. Let $U_\sigma \in (S^n)^{n'}$. By previous remarks, this is true if and only if there is a sequence in K

$$\tau \geq \gamma_1 \leq \tau_1 \geq \gamma_2 \leq \tau_2 \geq \cdots \geq \gamma_{n'} \leq \sigma$$

with $U_\tau \in S^n$. But this property of τ happens iff there is also a sequence in K

$$\tau' \geq \gamma'_1 \leq \tau'_1 \geq \gamma'_2 \leq \tau'_2 \geq \cdots \geq \gamma'_{n'} \leq \tau$$

with $U_{\tau'} \in S$. Concatenating the two sequences gives a sequence of length $(n + n')$ from $U_\tau \in S$ to U_σ . Thus $U_\sigma \in S^{n+n'}$ iff $U_\sigma \in (S^n)^{n'}$, and hence $(S^n)^{n'} = S^{n+n'}$. \square

Theorem 6.5. The interleaving distance of Defn. 2.4 is an extended pseudometric.

Proof. Because Lem. (2.3) is a stronger requirement than needed for Defn. 6.4, the collection $\{(-)^n\}_{n \geq 0}$ forms a super-linear family of translations. Then the result is immediate from [16, Theorem 3.21]. \square

6.2 Proofs from Sec. 3

To simplify notation, throughout the proofs of this section we often use a \bullet symbol to represent the set indexing a particular map when the subscript would be obvious from the given map. For example, we write $\varphi_\bullet : F(S^n) \rightarrow G(S^{2n})$ rather than writing φ_{S^n} .

Lemma 3.9. Assume we are given an n -assignment $\varphi : F \Rightarrow G^n$ and $\psi : G \Rightarrow F^n$. For a fixed k , define $(n+k)$ -assignments $\Phi_S = G[S^n \subseteq S^{n+k}] \circ \varphi_S$ and $\Psi_S = F[S^n \subseteq S^{n+k}] \circ \psi_S$ for all $S \in \mathbf{Open}(\mathcal{U})$. Then the following hold:

1. $L_{\nabla}^{S,T}(\varphi) \leq k$ implies $\nabla_\Phi(S, T)$ commutes, and thus $L_{\nabla}^{S,T}(\Phi) = 0$.
2. $L_{\square}^{S,T}(\psi) \leq k$ implies $\square_\Psi(S, T)$ commutes, and thus $L_{\square}^{S,T}(\Psi) = 0$.
3. $L_{\nabla}^S(\varphi, \psi) \leq k$ and $L_{\square}^{S^n, S^{n+k}}(\psi) \leq k$ imply $\nabla_{\Phi, \Psi}(S)$ commutes, and thus $L_{\nabla}^S(\Phi, \Psi) = 0$.
4. $L_{\triangle}^S(\varphi, \psi) \leq k$ and $L_{\nabla}^{S^n, S^{n+k}}(\varphi) \leq k$ imply $\triangle_{\Phi, \Psi}(S)$ commutes, and thus $L_{\triangle}^S(\Phi, \Psi) = 0$.

In particular, if φ and ψ have $L(\varphi, \psi) = 0$, then φ and ψ constitute an interleaving, and so $d_I(F, G) \leq n$.

Proof of Lem. 3.9. We prove the lemma for the first and third entries only as the other arguments are symmetric. Assume $L_{\sqsubseteq}^{S,T}(\varphi) \leq k$ and consider the diagram

$$\begin{array}{ccccc}
F(S) & \xrightarrow{F[\sqsubseteq]} & F(T) & & \\
\searrow \Phi_S & \searrow \varphi_S & \searrow \Phi_T & \searrow \varphi_T & \\
& & G(S^n) & \xrightarrow{G[\sqsubseteq]} & G(T^n) \\
& & \downarrow G[\sqsubseteq] & & \downarrow G[\sqsubseteq] \\
& & G(S^{n+k}) & \xrightarrow{G[\sqsubseteq]} & G(T^{n+k}).
\end{array} \tag{6.6}$$

Note that the top of the diagram (Eq. 6.6) given by

$$\begin{array}{ccc}
F(S) & \xrightarrow{F[\sqsubseteq]} & F(T) \\
\downarrow \varphi_S & & \downarrow \varphi_T \\
G(S^n) & \xrightarrow{G[\sqsubseteq]} & G(T^n)
\end{array} \tag{6.7}$$

does not necessarily commute in the case that $k \geq 1$, and the bottom of the diagram (Eq. 6.6) given by

$$\begin{array}{ccc}
F(S) & \xrightarrow{F[\sqsubseteq]} & F(T) \\
\downarrow \Phi_S & & \downarrow \Phi_T \\
G(S^{n+k}) & \xrightarrow{G[\sqsubseteq]} & G(T^{n+k})
\end{array} \tag{6.8}$$

is $\sqsubseteq_{\Phi}(S, T)$, for which we wish to check for commutativity. For any $x \in F(S)$, following around the top square, Eq. 6.7, gives

$$\begin{array}{ccc}
x & \xrightarrow{\quad} & x' \\
\downarrow & & \downarrow \\
a & \xrightarrow{\quad} & a'
\end{array}$$

with $d_{T^n}^G(a', b') \leq k$. By definition, the image of a' and b' is the same under the map $G(T^n) \rightarrow G(T^{n+k})$. Then since the front square of Eq. 6.6 given by

$$\begin{array}{ccc}
G(S^n) & \xrightarrow{G[\sqsubseteq]} & G(T) \\
\downarrow G[\sqsubseteq] & & \downarrow G[\sqsubseteq] \\
G(S^n) & \xrightarrow{G[\sqsubseteq]} & G(T^n)
\end{array}$$

commutes by functoriality of G , and the side triangles of Eq. 6.6 given by

$$\begin{array}{ccc} F(S) & \xrightarrow{\varphi_S} & G(S^n) \\ & \searrow \Phi_S & \downarrow G[\subseteq] \\ & & G(S^{n+k}) \end{array} \quad \begin{array}{ccc} F(T) & \xrightarrow{\varphi_T} & G(T^n) \\ & \searrow \Phi_T & \downarrow G[\subseteq] \\ & & G(T^{n+k}) \end{array}$$

commute by definition of Φ , we have that the image of x under either direction of the back square, Eq. 6.8, commutes, proving claim (1).

Turning to claim (3), consider the noncommutative diagram

$$\begin{array}{ccccccc} F(S) & \xrightarrow{F[\subseteq]} & F(S^{2n}) & \xrightarrow{F[\subseteq]} & F(S^{2n+k}) & \xrightarrow{F[\subseteq]} & F(S^{2(n+k)}) \\ & \searrow \Phi_\bullet & \searrow \psi_\bullet & \searrow \psi_\bullet & \searrow \psi_\bullet & \searrow \Psi_\bullet & \\ & & G(S^n) & \xrightarrow{G[\subseteq]} & G(S^{n+k}) & & \end{array} \quad (6.9)$$

The two yellow triangles

$$\begin{array}{ccccc} F(S) & & & F(S^{2n+k}) & \xrightarrow{F[\subseteq]} & F(S^{2(n+k)}) \\ & \searrow \Phi_\bullet & & \searrow \psi_\bullet & & \searrow \Psi_\bullet \\ & & G(S^n) & \xrightarrow{G[\subseteq]} & G(S^{n+k}) & \end{array}$$

commute by definition of Φ and Ψ . The blue parallelogram

$$\begin{array}{ccc} G(S^n) & \xrightarrow{G[\subseteq]} & G(S^{n+k}) \\ \downarrow \psi_\bullet & & \downarrow \psi_\bullet \\ F(S^{2n}) & \xrightarrow{G[\subseteq]} & F(S^{2n+k}) \end{array}$$

is the diagram $\square_{\psi}(S^n, S^{n+k})$ which also has loss function bounded by k , thus elements of $G(S^n)$ are not necessarily the same in the image of $F(S^{2n+k})$ following the parallelogram, but are the same in $F(S^{2(n+k)})$.

Checking that $\nabla_{\Phi, \Psi}(S)$ commutes amounts to a diagram chase. For an arbitrary $\alpha \in F(S)$, consider the following elements

$$\begin{array}{ccccccc} \alpha & \xrightarrow{\quad} & a & \xrightarrow{\quad} & b \\ & \searrow & a' & \xrightarrow{\quad} & b' \\ & & & \xrightarrow{\quad} & b'' \end{array} \quad \left. \begin{array}{c} \\ \\ \end{array} \right\} \xrightarrow{\quad} c$$

$$\begin{array}{ccc} & & \\ & \searrow & \\ & & x \end{array} \quad \begin{array}{ccc} & \searrow & \\ & & x' \end{array} \quad \begin{array}{ccc} & \searrow & \\ & & \end{array}$$

aligning with the diagram of Eq. 6.9. Both α and x map to x' because of the yellow triangle commuting, and both b'' and x' map to the same c for the same reason. Even if α and x map to different elements in $F(S^{2n})$, they must map to the same element in $F(S^{2(n+k)})$, and this element must be c , since both b' and b'' map to the same element by the bound on the blue parallelogram. As this was done for an arbitrary α , we have that $\nabla_{\Phi, \Psi}(S)$ commutes.

Claims (2) and (4) are similar with appropriate choices of diagrams. The final statement is immediate since $L(\varphi, \psi) = 0$ implies all diagrams needed for an interleaving commute. \square

Lemma 3.13. *Given a basis unnatural transformation*

$$\{\Phi_{S_\sigma} : F(S_\sigma) \rightarrow G(S_\sigma^N) \mid \sigma \in K\}$$

with $L_{\sqsubseteq}^{S_\tau, S_\sigma} = 0$ for all $\sigma \leq \tau$, we can extend this to a full natural transformation Φ ; i.e. we can define Φ_S for all S such that $L_{\sqsubseteq}^{S, T} = 0$ for all $S \subseteq T$.

Proof of Lem. 3.13. We start by defining Φ_S for arbitrary open sets. Note that since $L_{\sqsubseteq}^{S_\tau, S_\sigma} = 0$, for any $\sigma \leq \tau$, the diagram of the form

$$\begin{array}{ccc} F(S_\tau) & \xrightarrow{F[\sqsubseteq]} & F(S_\sigma) \\ \Phi_{S_\tau} \downarrow & & \downarrow \Phi_{S_\sigma} \\ G(S_\tau^n) & \xrightarrow{G[\sqsubseteq]} & G(S_\sigma^n) \end{array}$$

commutes.

For an arbitrary open S , define $\mathcal{U}_S = \{S_\sigma \mid U_\sigma \in S\}$. Any nonempty intersection $U_\sigma \cap U_\tau$ is also an element of \mathcal{U} , and so any nonempty intersection of $S_\sigma \cap S_\tau$ is an element of \mathcal{U}_S , so it is a cover of S . Then we use the fact that F is a cosheaf, and in particular this means that $F(S)$ is the coequalizer of the diagram

$$\coprod_{\sigma, \sigma'} F(S_\sigma \cap S_{\sigma'}) \xrightarrow[F[S_\sigma \cap S_{\sigma'} \subseteq S_{\sigma'}]]{F[S_\sigma \cap S_{\sigma'} \subseteq S_\sigma]} \coprod_{\tau} F(S_\tau).$$

Rephrased, this means that for any set Q with maps $F(S_\sigma) \rightarrow Q$ such that the solid arrow diagrams of the form

$$\begin{array}{ccc} F(S_\sigma \cap S_{\sigma'}) & \xrightarrow{F[\sqsubseteq]} & F(S_\sigma) \\ F[\sqsubseteq] \downarrow & & \downarrow F[\sqsubseteq] \\ F(S_{\sigma'}) & \xrightarrow{F[\sqsubseteq]} & F(S) \end{array} \quad \begin{array}{c} \curvearrowright \\ \text{---} \exists! \text{---} \\ \downarrow \\ Q \end{array}$$

commute for any σ, σ' , then there is a unique map $F(S) \rightarrow Q$ whose addition still has all diagrams commute. In our case, set $Q = G(S^n)$, and define the legs of the cocone to be $G[\sqsubseteq] \circ \Phi_{S_\sigma}$ as seen in the bold purple arrows of the diagram

$$\begin{array}{ccccc}
F(S_\sigma \cap S_{\sigma'}) & \xrightarrow{\Phi_\bullet} & G((S_\sigma \cap S_{\sigma'})^n) & & \\
\downarrow F[\sqsubseteq] & \searrow F[\sqsubseteq] & \downarrow G[\sqsubseteq] & \searrow G[\sqsubseteq] & \\
F(S_{\sigma'}) & \xrightarrow{\Phi_\bullet} & G(S_{\sigma'}^n) & & \\
\downarrow F[\sqsubseteq] & \searrow F[\sqsubseteq] & \downarrow G[\sqsubseteq] & \searrow G[\sqsubseteq] & \\
F(S_\sigma) & \xrightarrow{\Phi_\bullet} & G(S_\sigma^n) & & \\
\downarrow F[\sqsubseteq] & \searrow F[\sqsubseteq] & \downarrow G[\sqsubseteq] & \searrow G[\sqsubseteq] & \\
F(S) & \xrightarrow{\exists! \Phi_S} & G(S^n) & &
\end{array} \tag{6.10}$$

Note that the diagram prior to the inclusion of the dotted line commutes, since we can check the relevant faces as follows. The left and right squares commute because F and G are functors. The back and top panels commute because they involve only basis opens; equivalently, because we assumed $L_{\sqsubseteq}^{S_\sigma \cap S_{\sigma'}, S_\sigma} = L_{\sqsubseteq}^{S_\sigma \cap S_{\sigma'}, S_{\sigma'}} = 0$. Then, because $F(S)$ is a colimit of the diagram, there exists a unique map $\Phi_S : F(S) \rightarrow G(S^n)$ as noted, making any diagram of this form commute.

To ensure that the resulting Φ_S maps make diagrams of the form

$$\begin{array}{ccc}
F(S) & \longrightarrow & G(S^n) \\
\downarrow & & \downarrow \\
F(T) & \longrightarrow & G(T^n)
\end{array}$$

commute for arbitrary $S \subseteq T$, fix such a pair and an $x \in F(S)$. Because $F(S)$ is the colimit, there is a σ and an $x_\sigma \in F(S_\sigma)$ such that $x_\sigma \mapsto x$. In this case we have the diagram

$$\begin{array}{ccc}
F(S_\sigma) & \xrightarrow{\Phi_\bullet} & G(S_\sigma^n) \\
\searrow & & \searrow \\
& F(S) & \xrightarrow{\Phi_\bullet} G(S^n) \\
\searrow & \downarrow & \searrow \\
& F(T) & \xrightarrow{\Phi_\bullet} G(T^n)
\end{array}$$

The top and bottom squares of the wedge commute because they are the front of the cube of the diagram in Eq. (6.10). The left and right triangles commute since F and G are functors. Thus the front square commutes. This means the resulting Φ is a natural transformation, and thus $L_{\sqsubseteq}^{S, T} = 0$. \square

Lemma 3.14. *Given natural transformations $\Phi : F \Rightarrow G^N$ and $\Psi : G^N \Rightarrow F$ such that $L_{\sqsubseteq}^{S_\sigma} = 0$ for all $\sigma \in K$, then $L_{\sqsubseteq}^S = 0$ for all open sets S .*

Proof of Lem. 3.14. Because $L_{\nabla}^{S_\sigma} = 0$ for all basis elements, diagrams of the form

$$\begin{array}{ccc} F(S_\sigma) & \xrightarrow{F[\subseteq]} & F(S_\sigma^{2n}) \\ & \searrow \Phi_{S_\sigma} & \nearrow \Psi_{S_\sigma^n} \\ & G(S_\sigma^n) & \end{array}$$

commute for any $\sigma \in K$. Given an arbitrary open set S , let $x \in F(S)$ be given. As in the proof of Lem. 3.13, there is a σ and an $x_\sigma \in F(S_\sigma)$ with $x_\sigma \mapsto x$. Then consider the diagram

$$\begin{array}{ccccc} F(S_\sigma) & \xrightarrow{F[\subseteq]} & F(S_\sigma^{2n}) & & \\ & \searrow F[\subseteq] & \nearrow \Psi_\bullet & & \\ & & F(S) & \xrightarrow{F[\subseteq]} & F(S^{2n}) \\ & \searrow \Phi_\bullet & \nearrow & \searrow \Phi_\bullet & \nearrow \Psi_\bullet \\ & & G(S_\sigma^n) & \xrightarrow{G[\subseteq]} & G(S^n) \end{array}$$

The top square commutes because F is a functor. The back triangle commutes by this lemma's assumption. The left and right squares commute because Φ and Ψ are natural transformations. Taken together, this means that the front triangle commutes as required. \square

6.3 Proof from Sec. 5

Proposition 5.3. *For inputs $f : \mathbb{X} \rightarrow \mathbb{R}$ and $g : \mathbb{Y} \rightarrow \mathbb{R}$, denote the respective Reeb cosheaves as $\tilde{F}, \tilde{G} : \mathbf{Int} \rightarrow \mathbf{Set}$, and the respective mapper cosheaves as $F, G : \mathbf{Open}(K) \rightarrow \mathbf{Set}$. Then*

$$d_R(\tilde{F}, \tilde{G}) \leq (d_I(F, G) + 1) \delta.$$

Proof. Let φ, ψ be an n -interleaving for $F, G : \mathbf{Open}(\mathcal{U}) \rightarrow \mathbf{Set}$. We will construct an $\varepsilon = \delta(n+1)$ -interleaving $\tilde{\varphi}, \tilde{\psi}$ for $\tilde{F}, \tilde{G} : \mathbf{Int} \rightarrow \mathbf{Set}$.

We start by defining $\tilde{\varphi} : \tilde{F} \Rightarrow \tilde{G}^\varepsilon$ as $\tilde{\psi}$ is analogous. Given an arbitrary interval $I = (a, b)$, let $J = (j\delta, k\delta)$ be the smallest grid-aligned interval containing I ; i.e. $j\delta \leq a < (j+1)\delta$ and $(k-1)\delta < b \leq k\delta$.

Note that $I \subseteq J \subseteq J^{\delta n} \subseteq I^{(n+1)\delta} = I^\varepsilon$. Let $S = \{S_{\tau_i} \mid j \leq i \leq k-1\} \cup \{S_{\sigma_i} \mid j < i < k\}$. A quick check shows that $S \in \mathbf{Open}(\mathcal{U})$, that $J = |S|$, and that $J^{\delta n} = |S^n|$. Chasing definitions, this means that $\tilde{F}(J) = \pi_0(f^{-1}(J))$ and $F(S) = \pi_0(f^{-1}(|S|))$ are equal; similarly $\tilde{F}(J^{\delta n}) = F(S^n)$. Then define $\tilde{\varphi}_I$ to be the map defined by the

composition

$$\begin{array}{ccc}
\tilde{F}(I) & \xrightarrow{\tilde{\varphi}_I} & \tilde{G}(I^{(n+1)\delta}) \\
\tilde{F}[\subseteq] \downarrow & & \nearrow \tilde{G}[\subseteq] \\
\tilde{F}(J) & \xrightarrow{\tilde{\varphi}_J} & \tilde{G}(J^{\delta n}) \\
= \downarrow & & \uparrow = \\
F(S) & \xrightarrow{\varphi_S} & G(S^n).
\end{array}$$

Notice that setting I to be an axis aligned interval J gives the map $\tilde{\varphi}_J$ marked.

Now that we have built $\tilde{\varphi}$ and $\tilde{\psi}$, we need to check (i) that each is a natural transformation and (ii) that they satisfy the triangle diagrams of Defn. 5.2. For (i) we check only $\tilde{\varphi}$ as, again, $\tilde{\psi}$ is symmetric. To this end, assume we have $I \subseteq I'$ with minimal grid-aligned intervals J and J' , and let $S, S' \in \mathbf{Open}(\mathcal{U})$ be such that $|S| = J$ and $|S'| = J'$. Then consider the diagram

$$\begin{array}{ccccc}
\tilde{F}(I) & \xrightarrow{\tilde{\varphi}_I} & \tilde{G}(I^{(n+1)\delta}) & & \\
\tilde{F}[\subseteq] \downarrow & & \tilde{G}[\subseteq] \uparrow & & \\
\tilde{F}(J) & \searrow \tilde{F}[\subseteq] & \tilde{G}(J^{\delta n}) & \searrow \tilde{G}[\subseteq] & \\
= \downarrow & & \uparrow = & & \\
F(S) & \xrightarrow{\varphi_S} & G(S^n) & & \\
F[\subseteq] \searrow & & \nearrow G[\subseteq] & & \\
\tilde{F}(I') & \xrightarrow{\tilde{\varphi}_{I'}} & \tilde{G}((I')^{(n+1)\delta}) & & \\
\tilde{F}[\subseteq] \downarrow & & \tilde{G}[\subseteq] \uparrow & & \\
\tilde{F}(J') & \searrow \tilde{F}[\subseteq] & \tilde{G}((J')^{\delta n}) & \searrow \tilde{G}[\subseteq] & \\
= \downarrow & & \uparrow = & & \\
F(S') & \xrightarrow{\varphi_{S'}} & G((S')^n) & &
\end{array}$$

Note that the front and back panels of the cube are the diagrams that were used to define $\tilde{\varphi}_I$ and $\tilde{\varphi}_{I'}$, so they commute. The bottom panel commutes because φ is a natural transformation. The left and right panels commute because F and \tilde{F} arise from computing connected components on the same underlying input data. Thus, the top square commutes, and this is exactly what is needed to say that $\tilde{\varphi}$ is a natural transformation.

To check (ii), fix an interval I with grid aligned $J \subseteq I$ and $S \in \mathbf{Open}(\mathcal{U})$ with $|S| = J$. Then consider the diagram

$$\begin{array}{ccccc}
\tilde{F}(I) & \xrightarrow{\tilde{\varphi}_I} & \tilde{G}(I^{(n+1)\delta}) & \xrightarrow{\tilde{\psi}_{I^\varepsilon}} & \tilde{F}(I^{2(n+1)\delta}) \\
\tilde{F}[\subseteq] \downarrow & & \tilde{G}[\subseteq] \nearrow & & \tilde{F}[\subseteq] \nearrow \\
\tilde{F}(J) & & \tilde{G}(J^{\delta n}) & \xrightarrow{\tilde{G}[\subseteq]} & \tilde{G}(J^{(n+1)\delta}) & & \tilde{F}(J^{\delta(2n+1)}) \\
= \downarrow & & \uparrow = & & = \downarrow & & \uparrow = \\
F(S) & \xrightarrow{\varphi_S} & G(S^n) & \xrightarrow{G[\subseteq]} & G(S^{n+1}) & \xrightarrow{\psi_{S^{n+1}}} & F(S^{2n+1}) \\
& & \searrow \psi_{S^n} & & \nearrow \psi_{S^{n+1}} & & \\
& & F(S^{2n}) & & & &
\end{array}$$

$F[\subseteq]$ (from $F(S)$ to $F(S^{2n})$) and $F[\subseteq]$ (from $F(S^{2n})$ to $F(S^{2n+1})$)

The left and right hexa-laterals commute by definition of $\tilde{\varphi}$ and $\tilde{\psi}$ respectively. The middle top triangle commutes because \tilde{G} is a functor, and the middle square commutes because \tilde{G} and G are defined as connected components of the same input data. The bottom left triangle commutes because φ and ψ are an n -interleaving. The right quadrilateral commutes because ψ is a natural transformation. All this shows that the outside boundary of the diagram commutes. Swapping out the interior, we have

$$\begin{array}{ccc}
& \tilde{G}(I^{(n+1)\delta}) & \\
\tilde{\varphi}_I \nearrow & & \searrow \tilde{\psi}_{I^\varepsilon} \\
\tilde{F}(I) & \xrightarrow{\tilde{F}[\subseteq]} & \tilde{F}(I^{2(n+1)\delta}) \\
\tilde{F}[\subseteq] \downarrow & & \uparrow \tilde{F}[\subseteq] \\
\tilde{F}(J) & \xrightarrow{\tilde{F}[\subseteq]} & \tilde{F}(J^{\delta(2n+1)}) \\
= \downarrow & & \uparrow = \\
F(S) & \xrightarrow{F[\subseteq]} & F(S^{2n+1}) \\
& \searrow F[\subseteq] & \nearrow F[\subseteq] \\
& F(S^{2n}) &
\end{array}$$

The bottom triangle commutes because F is a functor, the next square up commutes by definition of F and \tilde{F} , and the top square commutes because \tilde{F} is a functor. Combining this with the outside ring commuting means that the top triangle commutes, which is the final ingredient needed for the definition of an interleaving. \square

7 Discussion

In this paper, we defined a loss function that quantifies how far a diagram is from being commutative, and used such a loss function to bound the interleaving distance, both

for mapper and Reeb graph settings. This work provides a way to evaluate a particular set of maps, which immediately suggests the question of utilizing this quantification to iteratively improve our comparison. Here, the quality of the bound is dependent on the quality of the input n -assignment, but we assume no control over that input in this paper and so we cannot evaluate the tightness of the bound. In the followup work [35], we will use this bound in the context of an ILP framework, where an input n -assignment can be improved incrementally thus finding a better bound on the distance. The potential for not only getting better approximations but also returning the actual interleaving maps used in the bound is an exciting step toward computing interleaving distances for graph-based signatures available in practice. Of course, we know that deciding if two Reeb graphs are ε -interleaved (for $\varepsilon \geq 1$) is NP-hard [18], so our ILP has no guarantee of reaching the global optimal solution. This is related to other available work providing bounds similarly lacking tightness guarantees for the interleaving distance restricted to merge trees [24, 25]. The first of these ([24]) uses the labeled merge tree distance [22, 23] which has more limited properties when generalizing to Reeb graphs [43] so it is unclear if this would generalize. The latter uses the map formulations on the topological spaces that arise in the special case of merge trees, so major modifications would need to be made for this to apply to the Reeb graph setting as seen here.

We believe that our loss function based framework is applicable in a broader context where data are modeled as sheaves or cosheaves in the category of sets, as sheaf theory is emerging as a tool in data science to study, e.g., distributed systems [44, 45], sensor networks [46], model fit [47], and uncertainty quantification [48]. In particular, one interesting next step is to study how to extend our framework to work with persistence modules as cosheaves in the category of vector spaces (e.g., [49]). As the interleaving distance for multiparameter persistence modules is similarly NP-hard [19], this would be an exciting step toward computational efforts in this broad class of topological signatures.

Declarations

Acknowledgments. The authors would like to thank Vidit Nanda for helpful discussions, and the careful feedback of the anonymous reviewers which greatly increased the clarity of the manuscript.

Conflict of interest. On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical Approval. Not applicable.

Funding. This work was funded in part by the National Science Foundation through grants CCF-1907591, CCF-1907612, CCF-2106578, CCF-2142713, CCF-2106672, DMS-2301361, DMS-2135960, DGE-2152014, and IIS-2145499. Research reported in this publication was supported by the National Cancer Institute of the National Institutes of Health under Award Number U54CA272167. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Availability of data and materials. Not applicable.

References

- [1] Ahmed, M., Karagiorgou, S., Pfoser, D., Wenk, C.: Map Construction Algorithms. Springer, Switzerland (2015). <https://doi.org/10.1007/978-3-319-25166-0>
- [2] Buchin, M., Chambers, E., Fang, P., Fasy, B.T., Gasparovic, E., Munch, E., Wenk, C.: Distances between immersed graphs: Metric properties. *La Matematica* **2**(1), 197–222 (2023) <https://doi.org/10.1007/s44007-022-00037-8>
- [3] Jiang, W., Luo, J.: Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications* **207**, 117921 (2022) <https://doi.org/10.1016/j.eswa.2022.117921>
- [4] Myers, A., Muñoz, D., Khasawneh, F.A., Munch, E.: Temporal network analysis using zigzag persistence. *EPJ Data Science* **12**(6) (2023) <https://doi.org/10.1140/epjds/s13688-023-00379-5> [arXiv:2205.11338](https://arxiv.org/abs/2205.11338) [cs.CG]
- [5] Ayzenberg, V., Kamps, F.S., Dilks, D.D., Lourenco, S.F.: Skeletal representations of shape in the human visual cortex. *Neuropsychologia* **164**, 108092 (2022) <https://doi.org/10.1016/j.neuropsychologia.2021.108092>
- [6] Zeng, D., Li, M., Jiang, N., Ju, Y., Schreiber, H., Chambers, E., Letscher, D., Ju, T., Topp, C.N.: Toporoot: a method for computing hierarchy and fine-grained traits of maize roots from 3D imaging. *Plant Methods* **17**(1) (2021) <https://doi.org/10.1186/s13007-021-00829-z>
- [7] Singh, G., Mémoli, F., Carlsson, G.: Topological methods for the analysis of high dimensional data sets and 3D object recognition. In: *Eurographics Symposium on Point-Based Graphics* (2007). <https://doi.org/10.2312/SPBG/SPBG07/091-100>
- [8] Deza, M.M., Deza, E.: *Encyclopedia of Distances*, 2nd edn. Springer, Heidelberg New York Dordrecht London (2013). <https://doi.org/10.1007/978-3-642-30958-8>
- [9] Conci, A., Kubrusly, C.: Distances between sets—a survey. *Advances in Mathematical Sciences and Applications* **26**(1), 1–18 (2017)
- [10] Donnat, C., Holmes, S.: Tracking network dynamics: A survey using graph distances. *The Annals of Applied Statistics* **12**(2) (2018) <https://doi.org/10.1214/18-aos1176>
- [11] Wills, P., Meyer, F.G.: Metrics for graph comparison: A practitioner’s guide. *PLOS ONE* **15**(2), 0228728 (2020) <https://doi.org/10.1371/journal.pone.0228728>
- [12] Yan, L., Masood, T.B., Sridharamurthy, R., Rasheed, F., Natarajan, V., Hotz,

- I., Wang, B.: Scalar field comparison with topological descriptors: Properties and applications for scientific visualization. *Computer Graphics Forum (CGF)* **40**(3), 599–633 (2021) <https://doi.org/10.1111/cgf.14331>
- [13] Bollen, B., Chambers, E.W., Levine, J., Munch, E.: Reeb graph metrics from the ground up (2021) [arXiv:2110.05631](https://arxiv.org/abs/2110.05631) [cs.CG]
 - [14] de Silva, V., Munch, E., Stefanou, A.: Theory of interleavings on categories with a flow. *Theory and Applications of Categories* **33**(21), 583–607 (2018)
 - [15] Chazal, F., Cohen-Steiner, D., Glisse, M., Guibas, L.J., Oudot, S.Y.: Proximity of persistence modules and their diagrams. In: *Proceedings of the 25th Annual Symposium on Computational Geometry. SCG '09*, pp. 237–246. ACM, New York, NY, USA (2009). <https://doi.org/10.1145/1542362.1542407>
 - [16] Bubenik, P., Silva, V., Scott, J.: Metrics for generalized persistence modules. *Foundations of Computational Mathematics* **15**(6), 1501–1531 (2014) <https://doi.org/10.1007/s10208-014-9229-5>
 - [17] Lesnick, M.: The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics* **15**(3), 613–650 (2015) <https://doi.org/10.1007/s10208-015-9255-y>
 - [18] Bjerkevik, H.B., Botnan, M.B.: Computational complexity of the interleaving distance. In: Speckmann, B., Tóth, C.D. (eds.) *34th International Symposium on Computational Geometry (SoCG 2018)*. *Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 99, pp. 13–11315. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2018). <https://doi.org/10.4230/LIPIcs.SoCG.2018.13>
 - [19] Bjerkevik, H.B., Botnan, M.B., Kerber, M.: Computing the interleaving distance is NP-hard. *Foundations of Computational Mathematics* **20**(5), 1237–1271 (2019) <https://doi.org/10.1007/s10208-019-09442-y>
 - [20] Farahbakhsh Touli, E., Wang, Y.: FPT-algorithms for computing Gromov-Hausdorff and interleaving distances between trees. In: Bender, M.A., Svensson, O., Herman, G. (eds.) *27th Annual European Symposium on Algorithms (ESA 2019)*. *Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 144, pp. 83–18314. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2019). <https://doi.org/10.4230/LIPIcs.ESA.2019.83>
 - [21] Kim, W., Mémoli, F., Stefanou, A.: Interleaving by parts: Join decompositions of interleavings and join-assemblage of geodesics. *Order* (2023) <https://doi.org/10.1007/s11083-023-09643-9>
 - [22] Munch, E., Stefanou, A.: The ℓ^∞ -cophenetic metric for phylogenetic trees as an interleaving distance. In: *Research in Data Science*. Association for Women

- in Mathematics Series, pp. 109–127. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11566-1_5
- [23] Gasparovic, E., Munch, E., Oudot, S., Turner, K., Wang, B., Wang, Y.: Intrinsic interleaving distance for merge trees. *La Matematica* (2024) <https://doi.org/10.1007/s44007-024-00143-9>
 - [24] Curry, J., Hang, H., Mio, W., Needham, T., Okutan, O.B.: Decorated merge trees for persistent topology. *Journal of Applied and Computational Topology* **6**(3), 371–428 (2022) <https://doi.org/10.1007/s41468-022-00089-3>
 - [25] Pegoraro, M.: A graph-matching formulation of the interleaving distance between merge trees (2024) <https://doi.org/10.48550/ARXIV.2111.15531> [arXiv:2111.15531v6](https://arxiv.org/abs/2111.15531v6) [math.CO]
 - [26] Morozov, D., Beketayev, K., Weber, G.: Interleaving Distance Between Merge Trees. In: *Proceedings of TopoInVis* (2013)
 - [27] Carrière, M., Oudot, S.: Structure and stability of the one-dimensional mapper. *Foundations of Computational Mathematics* **18**(6), 1333–1396 (2017) <https://doi.org/10.1007/s10208-017-9370-z>
 - [28] Carrière, M., Michel, B., Oudot, S.: Statistical analysis and parameter selection for mapper. *Journal of Machine Learning Research* **19**(12), 1–39 (2018)
 - [29] Munch, E., Wang, B.: Convergence between categorical representations of Reeb space and Mapper. In: Fekete, S., Lubiw, A. (eds.) *32nd International Symposium on Computational Geometry (SoCG 2016)*. *Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 51, pp. 53–15316. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2016). <https://doi.org/10.4230/LIPIcs.SoCG.2016.53>
 - [30] Brown, A., Bobrowski, O., Munch, E., Wang, B.: Probabilistic convergence and stability of random mapper graphs. *Journal of Applied and Computational Topology* (2020) <https://doi.org/10.1007/s41468-020-00063-x>
 - [31] Botnan, M.B., Curry, J., Munch, E.: A relative theory of interleavings (2020) [arXiv:2004.14286](https://arxiv.org/abs/2004.14286) [math.CT]
 - [32] Curry, J.: *Sheaves, cosheaves and applications*. PhD thesis, University of Pennsylvania (2014)
 - [33] Silva, V., Munch, E., Patel, A.: Categorified Reeb graphs. *Discrete & Computational Geometry* **55**(4), 854–906 (2016) <https://doi.org/10.1007/s00454-016-9763-9>
 - [34] Robinson, M.: Assignments to sheaves of pseudometric spaces. *Compositionality*

- 2 (2020) <https://doi.org/10.32408/compositionality-2-2>
- [35] Chambers, E.W., Ghosh, I., Munch, E., Percival, S., Wang, B.: Towards an optimal bound for the interleaving distance on mapper graphs (2025) <https://doi.org/10.48550/ARXIV.2504.03865> arXiv:2504.03865 [cs.CG]
 - [36] Riehl, E.: Category Theory in Context. Courier Dover Publications, New York, NY, USA (2017)
 - [37] Kaczynski, T., Mischaikow, K., Mrozek, M.: Computational Homology. Springer, New York, NY, USA (2004). <https://doi.org/10.1007/b97315>
 - [38] Barmak, J.A.: Algebraic Topology of Finite Topological Spaces and Applications. Springer, Berlin, Heidelberg (2011). <https://doi.org/10.1007/978-3-642-22003-6>
 - [39] Bochnak, J., Coste, M., Roy, M.-F.: Real Algebraic Geometry. Springer, Heidelberg (1998). <https://doi.org/10.1007/978-3-662-03718-8> . <http://dx.doi.org/10.1007/978-3-662-03718-8>
 - [40] Basu, S., Pollack, R., Roy, M.-F.: Algorithms in Real Algebraic Geometry. Springer, Heidelberg (2006). <https://doi.org/10.1007/3-540-33099-2>
 - [41] nLab authors: unnatural transformation. <https://ncatlab.org/nlab/show/unnatural+transformation>. Revision 1 (2023)
 - [42] Erickson, J.: Algorithms, (2019). <http://jeffe.cs.illinois.edu/teaching/algorithms/>
 - [43] Lan, F., Parsa, S., Wang, B.: Labeled interleaving distance for Reeb graphs. Journal of Applied and Computational Topology **8**(8), 2367–2399 (2024) <https://doi.org/10.1007/s41468-024-00193-6>
 - [44] Malcolm, G.: Sheaves, objects, and distributed systems. Electronic Notes in Theoretical Computer Science **225**(2), 3–19 (2009)
 - [45] Mansourbeigi, S.: Sheaf theory approach to distributed applications: Analysing heterogeneous data in air traffic monitoring. International Journal of Data Science and Analysis **3**(5), 34–39 (2017) <https://doi.org/10.11648/j.ijdsa.20170305.11>
 - [46] Robinson, M.: Sheaves are the canonical data structure for sensor integration. Information Fusion **36**, 208–224 (2017) <https://doi.org/10.1016/j.inffus.2016.12.002>
 - [47] Kvinge, H., Jefferson, B., Joslyn, C., Purvine, E.: Sheaves as a framework for understanding and interpreting model fit. IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) (2021) <https://doi.org/10.1109/ICCVW54120.2021.00469>

- [48] Joslyn, C.A., Charles, L., DePerno, C., Gould, N., Nowak, K., Praggastis, B., Purvine, E., Robinson, M., Strules, J., Whitney, P.: A sheaf theoretical approach to uncertainty quantification of heterogeneous geolocation information. *Sensors* **20**(12) (2020) <https://doi.org/10.3390/s20123418>
- [49] Bubenik, P., Milićević, N.: Homological algebra for persistence modules. *Foundations of Computational Mathematics* **21**, 1233–1278 (2021) <https://doi.org/10.1007/s10208-020-09482-9>