

Computing the Gromov-Hausdorff Distance for Metric Trees*

Pankaj K. Agarwal
Duke University
pankaj@cs.duke.edu

Kyle Fox
Duke University
kylefox@cs.duke.edu

Abhinandan Nath
Duke University
abhinath@cs.duke.edu

Anastasios Sidiropoulos
Ohio State University
sidiropoulos.1@osu.edu

Yusu Wang
Ohio State University
yusu@cse.ohio-state.edu

Abstract

The Gromov-Hausdorff (GH) distance is a natural way to measure distance between two metric spaces. We prove that it is NP-hard to approximate the Gromov-Hausdorff distance better than a factor of 3 for geodesic metrics on a pair of trees. We complement this result by providing a polynomial time $O(\min\{n, \sqrt{rn}\})$ -approximation algorithm for computing the GH distance between a pair of metric trees, where r is the ratio of the longest edge length in both trees to the shortest edge length. For metric trees with unit length edges, this yields an $O(\sqrt{n})$ -approximation algorithm.

*Work on this paper by P. K. Agarwal, K. Fox and A. Nath was supported by NSF under grants CCF-09-40671, CCF-10-12254, CCF-11-61359, and IIS-14-08846, and by Grant 2012/229 from the U.S.-Israel Binational Science Foundation. A. Sidiropoulos was supported by NSF under grants CAREER-1453472 and CCF-1423230. Y. Wang was supported by NSF under grant CCF-1319406.

1 Introduction

The Gromov-Hausdorff distance (or GH distance for brevity) [10] is one of the most natural distance measures between metric spaces, and has been used, for example, for matching deformable shapes [3, 15], and for analyzing hierarchical clustering trees [5]. Informally, the Gromov-Hausdorff distance measures the *additive* distortion suffered when mapping one metric space to another using a correspondence between their points. Multiple approaches have been proposed to estimate the Gromov-Hausdorff distance [3, 14, 15].

Despite much effort, the problem of computing, either exactly or approximately, GH distance has remained elusive. The problem is not known to be NP-hard, and computing the GH distance, even approximately, for *graphic metrics*¹ is at least as hard as the graph isomorphism problem. Indeed, the metrics for two graphs have GH distance 0 if and only if the two graphs are isomorphic. Motivated by this trivial hardness result, it is natural to ask whether GH distance becomes easier in more restrictive settings such as geodesic metrics over trees, where efficient algorithms are known for checking isomorphism [1].

Related work. Most work on associating points between two metric spaces involves *embedding* a given high dimensional metric space into an infinite host space of lower dimensional metric spaces. However, there is some work on finding a bijection between points in two given finite metric spaces that minimizes typically multiplicative distortion of distances between points and their images, with some limited results on additive distortion.

Kenyon *et al.* [13] give an optimal algorithm for minimizing the multiplicative distortion of a bijection between two equal-sized finite metric spaces, and a parameterized polynomial time algorithm that finds the optimal bijection between an arbitrary unweighted graph metric and a bounded-degree tree metric.

Papadimitriou and Safra [17] show that it is NP-hard to approximate the multiplicative distortion of any bijection between two finite 3-dimensional point sets to within any additive constant or to a factor better than 3.

Hall and Papadimitriou [11] discuss the *additive distortion problem* – given two equal-sized point sets $S, T \subset \mathbb{R}^d$, find the smallest Δ such that there exists a bijection $f : S \rightarrow T$ such that $d(x, y) - \Delta \leq d(f(x), f(y)) \leq d(x, y) + \Delta$. They show that it is NP-hard to approximate by a factor better than 3 in \mathbb{R}^3 , and also give a 2-approximation for \mathbb{R}^1 and a 5-approximation for the more general problem of embedding an arbitrary metric space onto \mathbb{R}^1 . However, their setting differs from ours in two major ways – firstly, they consider finite metric spaces of equal size, whereas in this paper the metric spaces may be uncountably infinite; secondly, they consider bijections between metric spaces, whereas in our work we deal with correspondences between metric spaces which are more general than bijections. Thus, their approach cannot be easily extended to our setting.

The interleaving distance between merge trees [16] was proposed as a measure to compare functions over topological domains that is stable to small perturbations in a function. Distances for the more general Reeb graphs are given in [2, 7]. These concepts are related to the GH distance (Section 4), which we will leverage to design an approximation algorithm for the GH distance for metric trees.

Our results. In this paper, we give the first non-trivial results on approximating the GH distance between metric trees. First, we prove (in Section 3) that the problem remains NP-hard even for metric trees via a reduction from 3-PARTITION. In fact, we show that there exists no algorithm with approximation ratio less than 3 unless $P = NP$. As noted above, we are not aware of any result that shows the GH distance problem being NP-hard even for general graphic metrics.

To complement our hardness result, we give an $O(\sqrt{n})$ -approximation algorithm for the GH distance between metric trees with n nodes and *unit length* edges. Our algorithm works with arbitrary edge lengths as well; however, the approximation ratio becomes $O(\min\{n, \sqrt{rn}\})$ where r is the ratio of the longest edge length in both trees to the shortest edge length. Even achieving the $O(n)$ -approximation ratio presented here for arbitrary r is a non-trivial task.

¹A graphic metric measures the shortest path distance between vertices of a graph with unit length edges.

Our algorithm uses a reduction, described in Section 4, to the similar problem of computing the *interleaving distance* [16] between two *merge trees*. Given a function $f : \mathbb{X} \rightarrow \mathbb{R}$ over a topological space \mathbb{X} , the merge tree T_f describes the connectivity between components of the sublevel sets of f (see Section 2 for a more formal definition). Morozov et al. [16] proposed the interleaving distance as a way to compare merge trees and their associated functions². We describe, in Section 5, an $O(\min\{n, \sqrt{rn}\})$ -approximation algorithm for interleaving distance between merge trees, and our reduction provides a similar approximation for computing the GH distance between two metric trees.

2 Preliminaries

Metric Spaces and the Gromov-Hausdorff Distance. A *metric space* $\mathcal{X} = (X, \rho)$ consists of a (potentially infinite) set X and a function $\rho : X \times X \rightarrow \mathbb{R}_{\geq 0}$ such that the following hold: $\rho(x, y) = 0$ iff $x = y$; $\rho(x, y) = \rho(y, x)$; and $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$.

Given sets A and B , a *correspondence* between A and B is a set $\mathcal{C} \subseteq A \times B$ such that: (i) for all $a \in A$, there exists $b \in B$ such that $(a, b) \in \mathcal{C}$; and (ii) for all $b \in B$, there exists $a \in A$ such that $(a, b) \in \mathcal{C}$. We use $\Pi(A, B)$ to denote the set of all correspondences between A and B .

Let $\mathcal{X}_1 = (X_1, \rho_1)$ and $\mathcal{X}_2 = (X_2, \rho_2)$ be two metric spaces. The *distortion* of a correspondence $\mathcal{C} \in \Pi(X_1, X_2)$ is defined as:

$$\text{Dist}(\mathcal{C}) = \sup_{(x, y), (x', y') \in \mathcal{C}} |\rho_1(x, x') - \rho_2(y, y')|.$$

The *Gromov-Hausdorff distance* [14], d_{GH} , between \mathcal{X}_1 and \mathcal{X}_2 is defined as:

$$d_{GH}(\mathcal{X}_1, \mathcal{X}_2) = \frac{1}{2} \inf_{\mathcal{C} \in \Pi(X_1, X_2)} \text{Dist}(\mathcal{C}).$$

Intuitively, d_{GH} measures how close can we get to an *isometric* (distance-preserving) embedding between two metric spaces. We note that there are different equivalent definitions of the Gromov-Hausdorff distance; see e.g. Theorem 7.3.25 of [4] and Remark 1 of [14].

Given a tree $T = (V, E)$ and a length function $l : E \rightarrow \mathbb{R}_{\geq 0}$, we associate a metric space $\mathcal{T} = (|T|, d)$ with T as follows. $|T|$ is a geometric realization of T . The metric space is extended to points in an edge such that each edge of length l is isometric to the interval $[0, l]$. For $x, y \in |T|$, define $d(x, y)$ to be the length of the path $\pi(x, y) \in |T|$ which is simply the sum of the lengths of the restrictions of this path to edges in T . It is clear that d is a metric. The metric space thus obtained is a *metric tree*. We often do not distinguish between T and $|T|$ and write $\mathcal{T} = (T, d)$.

Merge Trees and the Interleaving Distance. Let $f : \mathbb{X} \rightarrow \mathbb{R}$ be a continuous function from a connected topological space \mathbb{X} to the set of real numbers. The *sublevel set* at a value $a \in \mathbb{R}$ is defined as $f_{\leq a} = \{x \in \mathbb{X} \mid f(x) \leq a\}$. A *merge tree* M_f captures the evolution of the topology of the sublevel sets as the function value is increased continuously from $-\infty$ to $+\infty$. Formally, it is obtained as follows. Let $\text{epi } f = \{(x, y) \in \mathbb{X} \times \mathbb{R} \mid y \geq f(x)\}$. Let $\tilde{f} : \text{epi } f \rightarrow \mathbb{R}$ be such that $\tilde{f}((x, y)) = y$. We may say $\tilde{f}((x, y))$ is the *height* of point $(x, y) \in \mathbb{X} \times \mathbb{R}$. For two points (x, y) and (x', y') in $\mathbb{X} \times \mathbb{R}$ with $y = y'$, let $(x, y) \sim (x', y')$ denote them lying in the same component of $\tilde{f}^{-1}(y) (= \tilde{f}^{-1}(y'))$. Then \sim is an equivalence relation, and the merge tree M_f is defined as the quotient space $(\mathbb{X} \times \mathbb{R}) / \sim$.

Since two components of \tilde{f}^{-1} at a certain height can only merge at a higher height and a component can never split as height increases, we get a rooted tree where the internal nodes represent the points where two components merge and the leaves represent the birth of a new component at a local minimum. Figure 1 shows an example of a merge tree for a 1-dimensional function. Note that the merge tree extends to a height

²In fact, our hardness result can be easily extended to the GH distance between graphic metrics for trees and the interleaving distance between merge trees.

of ∞ , and our assumption that \mathbb{X} is connected implies we have only one component in $F_{\leq \infty}$. We define the *root* of merge tree M_f to be the internal node with the highest function value (if there are no internal nodes, the only leaf is defined to be the root).

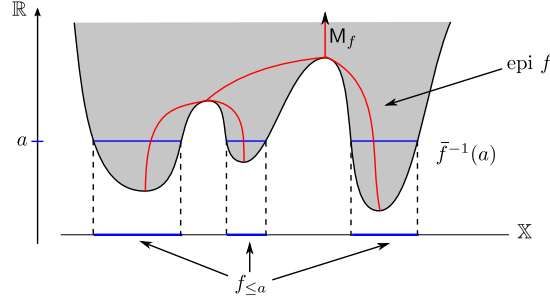


Figure 1. Merge tree M_f (shown in red) for a function $f : \mathbb{X} \rightarrow \mathbb{R}$, where $\mathbb{X} = \mathbb{R}$. $\text{epi } f$ is shown in grey, and $f^{-1}(a)$ is the grey region below the blue horizontal lines in $\text{epi } f$.

Since each point $x \in M_f$ represents a component of a sublevel set at a certain height, we can associate this height value with x , denoted by $\hat{f}(x)$. Given a merge tree M_f and $\varepsilon \geq 0$, an ε -shift map $\sigma_f^\varepsilon : M_f \rightarrow M_f$ is the map that maps a point $x \in M_f$ to its ancestor at height $\hat{f}(x) + \varepsilon$, i.e., $\hat{f}(\sigma_f^\varepsilon(x)) = \hat{f}(x) + \varepsilon$. Given $\varepsilon \geq 0$ and merge trees M_f and M_g , two continuous maps $\alpha : M_f \rightarrow M_g$ and $\beta : M_g \rightarrow M_f$ are said to be ε -compatible if they satisfy the following conditions :

$$\begin{aligned} \hat{g}(\alpha(x)) &= \hat{f}(x) + \varepsilon, \forall x \in M_f; & \hat{f}(\beta(y)) &= \hat{g}(y) + \varepsilon, \forall y \in M_g; \\ \beta \circ \alpha &= \sigma_f^{2\varepsilon}; & \alpha \circ \beta &= \sigma_g^{2\varepsilon}. \end{aligned} \tag{1}$$

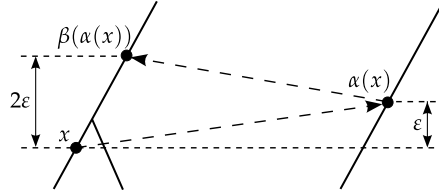


Figure 2. Part of trees M_f and M_g showing α and β .

See Figure 2 for an example. The *interleaving distance* [16] is then defined as

$$d_I(M_f, M_g) = \inf\{\varepsilon \geq 0 \mid \text{there exist } \varepsilon\text{-compatible maps } \alpha \text{ and } \beta\}.$$

Remark. We can relax the requirements on α and β from their normal definitions as follows.

(i) Instead of requiring *exact* value changes, we require

$$\hat{f}(x) \leq \hat{g}(\alpha(x)) \leq \hat{f}(x) + \varepsilon, \forall x \in M_f; \quad \hat{g}(y) \leq \hat{f}(\beta(y)) \leq \hat{g}(y) + \varepsilon, \forall y \in M_g.$$

(ii) If x_1 is an ancestor of x_2 in M_f , then $\alpha(x_1)$ is an ancestor of $\alpha(x_2)$ in M_g . A similar rule applies for β .

(iii) $\beta(\alpha(x))$ must go to an ancestor of x and $\alpha(\beta(y))$ must go to an ancestor of y .

Any pair of maps satisfying the original requirements also satisfies the relaxed requirements for the same value of ε . Conversely, for any pair of maps satisfying the relaxed requirements, we can *stretch up* the images for each map as necessary so that the new maps satisfy the original requirements, without changing the value of ε . Thus, both definitions of interleaving distance are equivalent. For convenience, when two

ε -compatible maps are given to us we assume that they satisfy (1), but we construct ε -compatible maps that satisfy the relaxed conditions mentioned, knowing that they can be “stretched” as just described to satisfy (1).

If we know $\alpha(x)$ for a point x at height h , then we can compute $\alpha(y)$ for any ancestor y of x at height $h' \geq h$ by simply putting $\alpha(y) = \sigma_f^{h'-h}(\alpha(x))$. A similar claim holds for β . Thus specifying the maps for the leaves of the trees suffices, because any point in the tree is the ancestor of at least one of the leaves. Hence, these maps have a representation that requires linear space in the size of the trees.

As shown in [16], the interleaving distance is a metric and has the desirable properties of being both stable to small function perturbations and more discriminative than the popular bottleneck distance between persistence diagrams [6].

3 Hardness of Approximation

We now show the hardness of approximating the GH distance by a reduction from the following decision problem called *balanced partition* (or BAL-PART for brevity): given a multiset of positive integers $X = \{a_1, \dots, a_n\}$, and an integer m such that $1 \leq m \leq n$, is it possible to partition X into m multisets $\{X_1, \dots, X_m\}$ such that all the elements in each multiset sum to the same quantity $\mu = (\sum_{i=1}^n a_i) / m$? We prove below that BAL-PART is *strongly* NP-complete, i.e., it remains NP-complete even if $a_i \leq n^c$ for some constant $c \geq 1$ for all $1 \leq i \leq n$.

Lemma 3.1. BAL-PART is *strongly* NP-complete.

Proof. We reduce 3-PARTITION, a strongly NP-complete problem [9] to BAL-PART. Given a multiset of positive integers $Y = \{a_1, \dots, a_n\}$ with $n = 3m$, 3-PARTITION asks to partition Y into m multisets $\{Y_1, \dots, Y_m\}$ of size 3 each so that the elements in each multiset sum to the same quantity. Given a 3-PARTITION instance, we construct an instance of BAL-PART as follows.

Basically, we add a sufficiently large number to each a_i so that if two multisets of the new numbers have the same sum, they have the same number of elements. In particular, let $\bar{a} = \sum_{i=1}^n a_i$. Then set $a'_i = a_i + \bar{a}$ and $X = \{a'_1, \dots, a'_n\}$. This reduction takes polynomial time, and the new numbers are polynomially larger than the original ones. We show that there exists an appropriate partition of Y iff there exists an appropriate partition of X .

Suppose there exists an appropriate partition $\{Y_1, \dots, Y_m\}$ of Y . Then setting $X_i = \{a'_j \mid a_j \in Y_i\}$ for $i = 1, \dots, m$ gives us the desired partition of X .

Conversely, suppose there exists an appropriate partition $\{X_1, \dots, X_m\}$ of X . Suppose $|X_i| = n_1 > |X_j| = n_2$ for some $i \neq j$. We thus have

$$\sum_{a'_k \in X_i} a_k + n_1 \bar{a} = \sum_{a'_k \in X_j} a_k + n_2 \bar{a} \Rightarrow (n_1 - n_2) \bar{a} = \sum_{a'_k \in X_j} a_k - \sum_{a'_k \in X_i} a_k \Rightarrow \sum_{a'_k \in X_j} a_k - \sum_{a'_k \in X_i} a_k \geq \bar{a}, \quad (2)$$

a contradiction since $\sum_{a'_k \in X_j} a_k < \bar{a}$. Thus, each partition X_i is of equal size. Since $n = 3m$, the size of each X_i is 3. \square

We now reduce an instance of BAL-PART, in which each $a_i \leq n^c$ for some constant $c \geq 1$, to GH-distance computation. Given an instance $X = \{a_1, \dots, a_n\}$ and $1 \leq m \leq n$ of BAL-PART, we construct two trees T_1 and T_2 as follows. Let $\lambda > 6$ and $\rho < \lambda - 6$ be two positive constants. Let $T_{l,k}$ denote a star graph having k edges, each of length l . T_1 consists of a node r_1 incident on an edge (r_1, r'_1) of length ρ and on n edges $\{(r_1, p_1), \dots, (r_1, p_n)\}$ of length 2, where p_i is the center of a copy of T_{λ, a_i} . T_2 consists of a node r_2 incident on an edge (r_2, r'_2) of length ρ and to m edges $\{(r_2, q_1), \dots, (r_2, q_m)\}$ of length 2, where each q_i is the center of a distinct copy of $T_{\lambda+1, \bar{a}}$, and $\bar{a} = (\sum_{i=1}^n a_i) / m$. See Figure 3 for an illustration. We refer to the edges of T_{λ, a_i} in T_1 and copies of $T_{\lambda+1, \bar{a}}$ in T_2 as *bottom* edges. Let \mathcal{T}_1 and \mathcal{T}_2 denote the metric trees associated with T_1 and T_2 respectively. Since λ, ρ are constants and $a_i \leq n^c$ for all $1 \leq i \leq n$, this construction can be done in polynomial time.

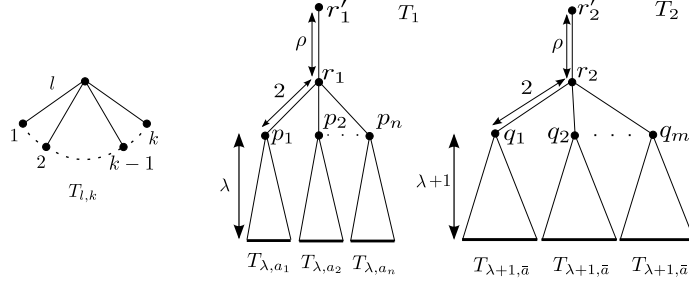


Figure 3. The trees $T_{l,k}$, T_1 and T_2 .

Lemma 3.2. *If (X, m) is a yes instance of BAL-PART, then $d_{GH}(\mathcal{T}_1, \mathcal{T}_2) \leq 1$. Otherwise, $d_{GH}(\mathcal{T}_1, \mathcal{T}_2) \geq 3$.*

Proof. Suppose X can be partitioned into m subsets X_1, \dots, X_m of equal weight $\bar{a} = (\sum_{i=1}^n a_i) / m$. We construct a correspondence \mathcal{C} between \mathcal{T}_1 and \mathcal{T}_2 with distortion at most 2, implying that $d_{GH}(\mathcal{T}_1, \mathcal{T}_2) \leq 1$. A linearly interpolated bijection between the points of edges (r_1, r'_1) and (r_2, r'_2) , with r_1 mapping to r_2 and r'_1 mapping to r'_2 , is added to \mathcal{C} . If $a_i \in X_j$, the linearly interpolated bijection between edges (r_1, p_i) and (r_2, q_j) is added to \mathcal{C} . Also, the leaves of T_{λ, a_i} are each mapped to a distinct leaf of $T_{\lambda+1, \bar{a}}$ attached to q_j such that there is a bijection between the leaves of T_1 and T_2 – this is possible since $T_{\lambda+1, \bar{a}}$ has \bar{a} leaves, and $\sum_{a \in X_j} a = \bar{a}$. The interior points of these leaf edges are mapped using linear interpolation. Overall, the distortion induced by \mathcal{C} is at most 2 – this stems from the fact that \mathcal{C} is piecewise linear, and the difference between the length of any path in one tree and its image under \mathcal{C} in the other tree is at most 2.

Suppose $d_{GH}(\mathcal{T}_1, \mathcal{T}_2) < 3$, and let \mathcal{C} be a correspondence between \mathcal{T}_1 and \mathcal{T}_2 with distortion < 6 . Consider two leaves $l, m \neq r'_2$ in \mathcal{T}_2 . Then $d(l, m) \geq 2\lambda + 2$. Let l', m' be their corresponding images in \mathcal{T}_1 under \mathcal{C} . We argue that l', m' lie on distinct bottom edges of \mathcal{T}_1 . Indeed, since $\text{Dist}(\mathcal{C}) < 6$, the distance between l' and m' is $d(l', m') > d(l, m) - 6 > 2\lambda - 4$. If l', m' lie on the same edge of \mathcal{T}_1 , then $d(l', m') \leq \lambda < 2\lambda - 4$, so they have to lie on distinct edges of \mathcal{T}_1 . If either of l', m' lies on an edge $r_1 p_i$, for some $i \leq n$, then by construction and the choice of ρ , $d(l', m') \leq \lambda + 2 < 2\lambda - 4$ (recall that $\lambda > 6$). Finally, if either of l', m' lies on (r_1, r'_1) then $d(l', m') \leq \rho + \lambda + 2 < 2\lambda - 4$. Thus, both l' and m' lie on distinct bottom edges of \mathcal{T}_1 . Hence, \mathcal{C} induces a bijection χ between the leaves of \mathcal{T}_1 and \mathcal{T}_2 , where $\chi(l) = l'$ for $l \in \mathcal{T}_2$ and $l' \in \mathcal{T}_1$ is the leaf whose incident edge contains the image(s) of l under \mathcal{C} . Note that if $l_i, l_j \in \mathcal{T}_2$ are incident to q_i, q_j with $q_i \neq q_j$, then $\chi(l_i)$ and $\chi(l_j)$ are incident to $p_{i'}, p_{j'}$ with $p_{i'} \neq p_{j'}$; otherwise $d(l_i, l_j) = 2\lambda + 6$ and $d(\chi(l_i), \chi(l_j)) \leq 2\lambda$, thereby incurring a distortion of at least 6. Hence, the bijection χ can be used to partition X into m subsets X_1, \dots, X_m of equal weight as follows : if $\chi(l) = l'$ for l incident to q_i and l' incident to p_j , then $a_j \in X_i$. Thus, (X, m) is a yes instance of BAL-PART. \square

We may also apply the reduction to metric trees with unit edge lengths by subdividing longer edges with an appropriate number of vertices. We thus have the following theorem.

Theorem 3.3. *Unless $P = NP$, there is no polynomial-time algorithm to approximate the Gromov-Hausdorff distance between two metric trees to a factor better than 3, even in the case of metric trees with unit edge lengths.*

4 Gromov-Hausdorff and Interleaving Distances

In this section we show that the GH distance between two tree metric spaces \mathcal{T}_1 and \mathcal{T}_2 , and the interleaving distance between two appropriately defined trees incuded from \mathcal{T}_i s, are within constant factors of each other.

Given a metric tree $\mathcal{T} = (T, d)$, let $V(T)$ denote the nodes of the tree. Given a point $s \in T$ (not necessarily a node), let $f_s : T \rightarrow \mathbb{R}$ be defined as $f_s(x) = -d(s, x)$. Equipped with this function, we obtain a merge tree T^s from \mathcal{T} . Intuitively, T^s has the structure of rooting T at s , and then adding an extra edge incident to s

with function value extending from 0 to $+\infty$. If s is an internal node of T or an interior point of an edge of T , s remains the root of T^s . But if s is a leaf of T , then s gets merged with the infinite edge and the node of T adjacent to s becomes the root of T^s .

Let $\mathcal{T}_1 = (T_1, d_1)$ and $\mathcal{T}_2 = (T_2, d_2)$ be two metric trees. Define

$$\Delta = \min_{u \in V(T_1), v \in V(T_2)} d_I(T_1^u, T_2^v). \quad (3)$$

We prove that Δ is within a constant factor of $d_{GH}(\mathcal{T}_1, \mathcal{T}_2)$. We first prove a lower bound on Δ .

Lemma 4.1. $\frac{1}{2}d_{GH}(\mathcal{T}_1, \mathcal{T}_2) \leq \Delta$.

Proof. Suppose $\Delta = d_I(T_1^s, T_2^t)$ for some $s \in V(T_1)$ and $t \in V(T_2)$. Set $f := f_s$ and $g := f_t$. Let $\alpha : T_1^s \rightarrow T_2^t, \beta : T_2^t \rightarrow T_1^s$ be Δ -compatible maps. We define the functions $\alpha^* : T_1^s \rightarrow T_2^t$ and $\beta^* : T_2^t \rightarrow T_1^s$ as follows :

$$\alpha^*(x) = \begin{cases} \alpha(x) & \text{if } g(\alpha(x)) \leq 0. \\ t & \text{otherwise.} \end{cases} \quad \beta^*(y) = \begin{cases} \beta(y) & \text{if } f(\beta(y)) \leq 0. \\ s & \text{otherwise.} \end{cases}$$

That is, if $\alpha(x)$ is an ancestor of t (resp. s) then x (resp. y) is mapped to the root t (resp. s). We note that

$$\begin{aligned} f(x) &\leq g(\alpha^*(x)) \leq f(x) + \Delta, \\ g(y) &\leq f(\beta^*(y)) \leq g(y) + \Delta. \end{aligned} \quad (4)$$

Indeed, if $\alpha^*(x) = \alpha(x)$ then $g(\alpha^*(x)) = f(x) + \Delta$. Otherwise $g(\alpha(x)) > 0$ and $g(\alpha^*(x)) = 0$. Since $f(x) \leq 0$, we obtain $g(\alpha^*(x)) < g(\alpha(x)) = f(x) + \Delta$. The same argument implies the second set of inequalities.

Consider the correspondence $\mathcal{C} \in T_1 \times T_2$ induced by α^* and β^* defined as:

$$\mathcal{C} := \{(x, \alpha^*(x)) \mid x \in T_1\} \cup \{(\beta^*(y), y) \mid y \in T_2\}.$$

We prove that $\text{Dist}(\mathcal{C}) \leq 4\Delta$.

Indeed, consider any two pairs $(x_1, y_1), (x_2, y_2) \in \mathcal{C}$. Let u be the common ancestor of x_1 and x_2 in T_1 , and w the common ancestor of y_1 and y_2 in T_2 . Note that since T_1 and T_2 are trees, there is a unique path $x_1 \rightsquigarrow u \rightsquigarrow x_2$ between x_1 and x_2 , such that $x_1 \rightsquigarrow u$ and $u \rightsquigarrow x_2$ are each monotone in function f values. This also implies that $d_1(x_1, u) = d_1(s, x_1) - d_1(s, u) = f(u) - f(x_1)$; similarly, $d_1(x_2, u) = f(u) - f(x_2)$. Symmetric statements hold for $y_1 \rightsquigarrow w \rightsquigarrow y_2$. Hence

$$\begin{aligned} d_1(x_1, x_2) &= d_1(x_1, u) + d_1(u, x_2) = 2f(u) - f(x_1) - f(x_2), \\ d_2(y_1, y_2) &= d_2(y_1, w) + d_2(w, y_2) = 2g(w) - g(y_1) - g(y_2). \end{aligned}$$

We then have,

$$\begin{aligned} |d_1(x_1, x_2) - d_2(y_1, y_2)| &= |2f(u) - f(x_1) - f(x_2) - 2g(w) + g(y_1) + g(y_2)| \\ &\leq 2|f(u) - g(w)| + |f(x_1) - g(y_1)| + |f(x_2) - g(y_2)| \\ &\leq 2|f(u) - g(w)| + 2\Delta \text{ (by (4))}. \end{aligned}$$

On the other hand, $\alpha^*(u)$ must be an ancestor of w , and similarly, $\beta^*(w)$ must be an ancestor of u . Thus, $f(u) - \Delta \leq g(w) \leq f(u) + \Delta \Rightarrow |f(u) - g(w)| \leq \Delta$. We thus have

$$|d_1(x_1, x_2) - d_2(y_1, y_2)| \leq 4\Delta.$$

It then follows that $\text{Dist}(\mathcal{C}) \leq 4\Delta$. Since $d_{GH}(\mathcal{T}_1, \mathcal{T}_2) \leq \frac{1}{2}\text{Dist}(\mathcal{C})$, the left inequality then follows. \square

Next, we prove an upper bound on Δ .

Lemma 4.2. $\Delta \leq 14d_{GH}(\mathcal{T}_1, \mathcal{T}_2)$.

Proof. Set $\delta = d_{GH}(\mathcal{T}_1, \mathcal{T}_2)$ and let $\mathcal{C}^* : T_1 \times T_2$ be an optimal correspondence that achieves $d_{GH}(\mathcal{T}_1, \mathcal{T}_2)$. Note that in general $d_{GH}(\mathcal{T}_1, \mathcal{T}_2)$ may only be achieved in the limit. In that case, our proof can be modified by considering a sequence of near-optimal correspondences (whose associated metric-distortion converges to δ), and taking a certain limit under it.

Let s be one of the endpoints of a longest simple path in T_1 (i.e, the length of this path realizes the diameter of \mathcal{T}_1); s is necessarily a leaf of T_1 . Let (s, t) be a pair in \mathcal{C}^* . Consider the merge trees T_1^s and T_2^t defined by the functions f_s and f_t , respectively. A result in [8] implies that

$$d_I(T_1^s, T_2^t) \leq 6\delta.$$

We prove below in Claim 4.3 that there is a vertex (in fact a leaf) $z \in V(T_2)$ such that $d_2(t, z) \leq 8\delta$.

It is easy to verify that

$$\|f_t - f_z\|_\infty \leq d_2(t, z) \leq 8\delta.$$

On the other hand, by the stability theorem of the interleaving distance (Theorem 2 of [16]),

$$d_I(T_2^t, T_2^z) \leq \|f_t - f_z\|_\infty \leq 8\delta.$$

By triangle inequality,

$$\begin{aligned} d_I(T_1^s, T_2^z) &\leq d_I(T_1^s, T_2^t) + d_I(T_2^t, T_2^z) \\ &\leq 6\delta + 8\delta \\ &\leq 14\delta. \end{aligned}$$

This completes the proof of the lemma. \square

Claim 4.3. *Let s be an endpoint of a longest simple path in T_1 , and let (s, t) be a pair in \mathcal{C}^* . Then there is a vertex $z \in V(T_2)$ such that $d_2(t, z) \leq 8\delta$.*

Proof. Assume that there is no tree node within 8δ distance to t . In this case, t must be in the interior of an edge $e \in E(T_2)$. Let u_1 and u_2 be the two points in e from opposite sides of t such that $d_2(t, u_1) = d_2(t, u_2) = 8\delta + \nu$, where $\nu > 0$ is an arbitrarily small value. Both u_1 and u_2 exist, as there is no tree node of T_2 within 8δ distance to t , and

$$d_2(u_1, u_2) = d_2(t, u_1) + d_2(t, u_2) = 16\delta + 2\nu.$$

Let $\tilde{u}_1, \tilde{u}_2 \in T_1$ be any corresponding points for u_1 and u_2 under \mathcal{C}^* , that is, $(\tilde{u}_1, u_1), (\tilde{u}_2, u_2) \in \mathcal{C}^*$. Since $\text{Dist}(\mathcal{C}^*) \leq 2\delta$, we have

$$d_1(\tilde{u}_1, \tilde{u}_2) \geq 14\delta + 2\nu. \tag{5}$$

On the other hand, since $d_2(t, u_1) = d_2(t, u_2) = 8\delta + \nu$, we have that

$$d_1(s, \tilde{u}_1), d_1(s, \tilde{u}_2) \in [6\delta + \nu, 10\delta + \nu]. \tag{6}$$

We now obtain an upper bound on $d_1(\tilde{u}_1, \tilde{u}_2)$. If \tilde{u}_1 and \tilde{u}_2 have ancestor/descendant relation in T_1^s , then $d_1(\tilde{u}_1, \tilde{u}_2) = |d_1(s, \tilde{u}_1) - d_1(s, \tilde{u}_2)|$ and by (6), we thus have that $d_1(\tilde{u}_1, \tilde{u}_2) \leq 4\delta$, which contradicts (5).

Now, let w be the nearest common ancestor of \tilde{u}_1 and \tilde{u}_2 in T_1^s (see Figure 4). Let $c_0 = d_1(s, w)$. For simplicity, set $a = d_1(s, \tilde{u}_1)$ and $b = d_1(s, \tilde{u}_2)$. It then follows that

$$d_1(\tilde{u}_1, \tilde{u}_2) = a + b - 2c_0. \text{ Note, } a \geq c_0, b \geq c_0. \tag{7}$$

Since s is an endpoint of the longest path in T_1 , it follows that $c_0 \geq \min\{a - c_0, b - c_0\}$ (if not, then without loss of generality, suppose the other point s' of the diameter pair is not in the subtree of T_1^s rooted at \tilde{u}_1 ; then $d_1(\tilde{u}_1, s') > d_1(s, s')$, a contradiction). By (6), $a, b \geq 6\delta + \nu$. Thus

$$c_0 \geq 6\delta + \nu - c_0 \Rightarrow c_0 \geq \frac{1}{2}[6\delta + \nu]. \tag{8}$$

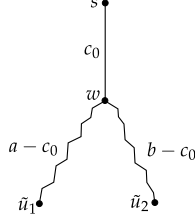


Figure 4. w is the nearest common ancestor of \tilde{u}_1 and \tilde{u}_2 in T_1^s .

Combining (7) and (8), we have

$$d_1(\tilde{u}_1, \tilde{u}_2) \leq a + b - 6\delta - \nu \leq 20\delta + 2\nu - 6\delta - \nu = 14\delta + \nu, \quad (9)$$

contradicting (5). Thus, there exists $z \in V(T_2)$ such that $d_2(t, z) \leq 8\delta$. \square

Remark. The proof of Claim 4.3 actually shows that t lies in the neighborhood of a leaf, as we never use the fact that u_1 and u_2 lie on the same edge of T_2 . The only fact we use is that u_1 and u_2 lie on opposite sides of t at distance $8\delta + \nu$ each.

From Lemmas 4.1 and 4.2, we get the following.

Theorem 4.4. Let $\Delta = \min_{u \in V(T_1), v \in V(T_2)} d_I(T_1^u, T_2^v)$. Then

$$\frac{1}{2}d_{GH}(\mathcal{T}_1, \mathcal{T}_2) \leq \Delta \leq 14d_{GH}(\mathcal{T}_1, \mathcal{T}_2).$$

Corollary 4.5. If there is a polynomial time, c -approximation algorithm for the interleaving distance between two merge trees, then there is a polynomial time, $28c$ -approximation algorithm for the Gromov-Hausdorff distance between two metric trees.

5 Computing the Interleaving Distance

Let M_f and M_g be merge trees of two functions f and g , respectively. For simplicity, we use f and g to denote the height functions on M_f and M_g as well. Let n be the total number of nodes in M_f and M_g , and let $r \geq 1$ be the ratio between the lengths of the longest and the shortest edges in M_f and M_g . We describe a $O(\min\{n, \sqrt{rn}\})$ -approximation algorithm for computing $d_I(M_f, M_g)$.

Candidate values and binary search. We first show that a candidate set Λ of $O(n^2)$ values can be computed in $O(n^2)$ time such that $d_I(M_f, M_g) \in \Lambda$. Given Λ , we perform a binary search on Λ . At each step, we use a c -approximate decision procedure, for $c = c_1 \min\{n, \sqrt{rn}\}$ for some constant c_1 , that given a value $\varepsilon > 0$ does the following : if $d_I(M_f, M_g) \leq \varepsilon$, it returns a pair of $c\varepsilon$ -compatible maps between M_f and M_g ; if $d_I(M_f, M_g) > \varepsilon$, it will either return a pair of $c\varepsilon$ -compatible maps between M_f and M_g or report that no such maps exist. The binary search terminates when one of the following two conditions meet :

- (i) We have two consecutive values $\varepsilon^-, \varepsilon^+ \in \Lambda$ with $\varepsilon^- < \varepsilon^+$ such that the decision procedure returned YES for ε^+ and NO for ε^- ; in this case we return ε^+ .
- (ii) We have two (not necessarily consecutive) values $\varepsilon^-, \varepsilon^+ \in \Lambda$ with $\varepsilon^- < \varepsilon^+$ such that the decision procedure returned NO for ε^+ and YES for ε^- (but with ε' -compatible maps for some $\varepsilon' > \varepsilon^+$), in which case we return ε^- .

It is clear that the procedure returns a value ε such that $d_I(M_f, M_g) \leq \varepsilon \leq cd_I(M_f, M_g)$. We now describe the candidate set Λ .

Let V_f (resp. V_g) be the set of nodes in M_f (resp. M_g). We define $\Lambda = \Lambda_{11} \cup \Lambda_{22} \cup \Lambda_{12}$, where

$$\begin{aligned}\Lambda_{11} &= \{\tfrac{1}{2}|f(u) - f(v)| \mid u, v \in V_f\}, \\ \Lambda_{22} &= \{\tfrac{1}{2}|g(u) - g(v)| \mid u, v \in V_g\}, \\ \Lambda_{12} &= \{|f(u) - g(v)| \mid u \in V_f, v \in V_g\}.\end{aligned}$$

Lemma 5.1. $d_I(M_f, M_g) \in \Lambda$.

Proof. Suppose to the contrary that $d_I(M_f, M_g) = \varepsilon \notin \Lambda$. Let $\alpha : M_f \rightarrow M_g$ and $\beta : M_g \rightarrow M_f$ be ε -compatible maps that realize $d_I(M_f, M_g) = \varepsilon$. We will obtain a contradiction by choosing $\varepsilon_0 > 0$ and constructing $(\varepsilon - \varepsilon_0)$ -compatible maps $\hat{\alpha}, \hat{\beta}$.

For any point $x \in M_f$, we define $\alpha_{\downarrow}(x) = \alpha(x)$ if $\alpha(x)$ is a node of M_g , otherwise $\alpha_{\downarrow}(x)$ is the lower endpoint of the edge of M_g containing $\alpha(x)$. Similarly we define the function $\beta_{\downarrow} : M_g \rightarrow M_f$.

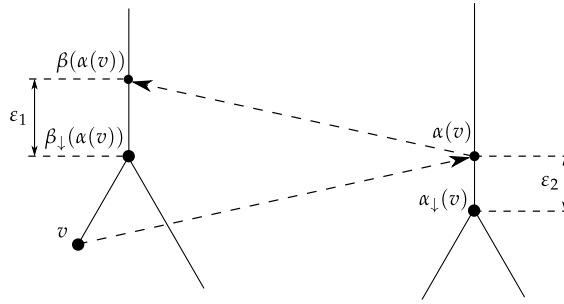


Figure 5. Trees M_f and M_g . Here $\delta_v = \min\{\frac{1}{2}\varepsilon_1, \varepsilon_2\}$.

For every node $v \in V_f$, $\alpha(v)$ (resp. $\beta(\alpha(v))$) lies in the interior of an edge of M_g (resp. M_f), because $\varepsilon \notin \Lambda \supseteq \Lambda_{12}$ (resp. Λ_{11}). We define

$$\delta_v = \min\{\tfrac{1}{2}(f(\beta(\alpha(v))) - f(\beta_{\downarrow}(\alpha(v))), g(\alpha(v)) - g(\alpha_{\downarrow}(v)))\}.$$

See Figure 5. Similarly we define δ_w for all $w \in V_g$. We set

$$\varepsilon_0 = \min\{\varepsilon, \min_{v \in V_f \cup V_g} \delta_v\}.$$

Since $\varepsilon \notin \Lambda$, we have $\varepsilon_0 > 0$. We now construct $(\varepsilon - \varepsilon_0)$ -compatible maps $\hat{\alpha} : M_f \rightarrow M_g$ and $\hat{\beta} : M_g \rightarrow M_f$. We describe the construction of $\hat{\alpha}$; $\hat{\beta}$ is constructed similarly. By construction, for any node $u \in V_f$, $g(\alpha(u)) - g(\alpha_{\downarrow}(u)) \geq \varepsilon_0$, so we set $\hat{\alpha}(u)$ to be the point w on the edge of M_g containing $\alpha(u)$ such that $g(w) = f(u) + \varepsilon - \varepsilon_0$. Once we have defined $\hat{\alpha}(u)$ and $\hat{\alpha}(v)$ for an edge $uv \in M_f$, with $f(u) < f(v)$, we set $\hat{\alpha}(x)$, for a point $x \in uv$ with $f(x) = f(u) + \gamma$, to be

$$\hat{\alpha}(x) = \sigma_g^\gamma(\hat{\alpha}(u)).$$

That is, we set $\hat{\alpha}(x)$ to be the ancestor of $\hat{\alpha}(u)$ at height $f(u) + \varepsilon - \varepsilon_0 + \gamma = f(x) + \varepsilon - \varepsilon_0$.

We claim that $\hat{\alpha}, \hat{\beta}$ are $(\varepsilon - \varepsilon_0)$ -compatible. Indeed, by construction, $g(\hat{\alpha}(x)) = f(x) + \varepsilon - \varepsilon_0$ for all $x \in M_f$, and $f(\hat{\beta}(y)) = g(y) + \varepsilon - \varepsilon_0$ for all $y \in M_g$. We now prove that

$$\hat{\beta} \circ \hat{\alpha} = \sigma_f^{2(\varepsilon - \varepsilon_0)}.$$

Suppose to the contrary there is a point $x \in M_f$ such that $y = \hat{\beta}(\hat{\alpha}(x)) \neq \sigma_f^{2(\varepsilon - \varepsilon_0)}(x)$. Since $f(y) =$

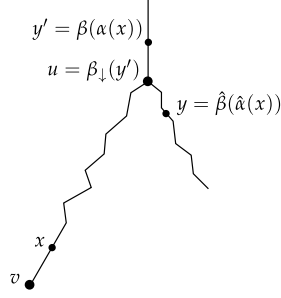


Figure 6. Figure showing u, v, x, y and y' .

$f(x) + 2(\varepsilon - \varepsilon_0)$, y must not be an ancestor of x . On the other hand, α, β are ε -compatible, so $y' = \beta(\alpha(x))$ is the ancestor of x at height $f(x) + 2\varepsilon$. By construction of $\hat{\alpha}$ and $\hat{\beta}$, y is a descendant of y' , in which case there is a node $u \in V_f$ that lies between y and y' . (If y and y' lie on the same edge of M_f , then y is also an ancestor of x .) Let $u = \beta_{\downarrow}(\alpha(x))$. Let v be the lower endpoint of the edge e containing x . See Figure 6. Since $\varepsilon \notin \Delta$, $f(u) \neq f(v) + 2\varepsilon$ (i.e., $u \neq \beta(\alpha(v))$). There are two cases to consider :

- (i) $f(u) > f(v) + 2\varepsilon$. Then let $u = \beta(\alpha(z))$ for the point z lying between x and v at height $f(z) = f(u) - 2\varepsilon$. Furthermore $f(x) \geq f(z) > f(x) - 2\varepsilon_0$ (if $f(x) - f(z) \geq 2\varepsilon_0$, then $f(y') - f(u) = f(x) - f(z) \geq 2\varepsilon_0$, contradicting the fact that $f(y') - f(y) = 2\varepsilon_0$). Therefore we can choose a point $w \neq v$ on e such that $f(z) > f(w) > f(x) - 2\varepsilon_0$. Now, it's not too hard to see that if x_1 is an ancestor of x_2 in M_f , then $\hat{\alpha}(x_1)$ is an ancestor of $\hat{\alpha}(x_2)$ (similarly for $\hat{\beta}$). Further, $\hat{\beta}(\hat{\alpha}(x_1))$ is a descendant of $\beta(\alpha(x_1))$ for all $x_1 \in M_f$ (a similar result holds for $\hat{\alpha} \circ \hat{\beta}$ and $\alpha \circ \beta$). Thus, $\hat{\beta}(\hat{\alpha}(w))$ is a descendant of $y = \hat{\beta}(\hat{\alpha}(x))$ (since w is a descendant of x). Moreover, $\beta(\alpha(w))$ is an ancestor of $\hat{\beta}(\hat{\alpha}(w))$. However, since $f(\beta(\alpha(w))) < f(u)$, $\beta(\alpha(w))$ lies between y and u . Thus, $\beta(\alpha(w))$ is not an ancestor of x (hence w), i.e., $\beta(\alpha(w)) \neq \sigma_f^{2\varepsilon}(w)$, contradicting the fact that α, β are ε -compatible.

- (ii) $f(u) < f(v) + 2\varepsilon$. In this case

$$f(\beta(\alpha(v))) - f(\beta_{\downarrow}(\alpha(v))) \leq f(\beta(\alpha(v))) - f(u) < f(y') - f(y) = 2\varepsilon_0 \leq 2\delta_v,$$

which contradicts the definition of δ_v .

Hence we conclude that y is an ancestor of x , i.e., $\hat{\beta} \circ \hat{\alpha} = \sigma_f^{2(\varepsilon - \varepsilon_0)}$. Similarly, we argue that $\hat{\alpha} \circ \hat{\beta} = \sigma_g^{2(\varepsilon - \varepsilon_0)}$, implying that $\hat{\alpha}, \hat{\beta}$ are $(\varepsilon - \varepsilon_0)$ -compatible maps as claimed.

Putting everything together, we conclude that $\varepsilon \in \Delta$. □

We now describe the decision procedure to answer the question “is $d_I(M_f, M_g) \leq \varepsilon$?” approximately. We define the *length* of any edge in a merge tree (other than the edge to infinity) to be the height difference between its two endpoints. Given a parameter $\varepsilon > 0$, an edge is called ε -long, or *long* for brevity, if its length is strictly greater than 2ε . We first describe an exact decision procedure for the case when all edges in both trees are long, and then describe an approximate decision procedure for the case when the two trees have short edges.

Trees with long edges. We remove all degree-two nodes in the beginning. A *subtree* rooted at a point x in a merge tree M , denoted M^x , includes all the points in the merge tree that are descendants of x and an edge from x that extends upwards to height ∞ . For a node $u \in V$, let $C(u)$ denote the children of u and let $p(u)$ denote its parent. Assume $d_I(M_f, M_g) \leq \varepsilon$, and let $\alpha : M_f \rightarrow M_g$ and $\beta : M_g \rightarrow M_f$ be a pair of ε -compatible maps. As in the proof of Lemma 5.1, we define the functions α_{\downarrow} and β_{\downarrow} but restricted only to the vertices of M_f and M_g . That is, for a node $v \in V_f$, we define $\alpha_{\downarrow}(v)$ to be the lower endpoint of the edge containing $\alpha(v)$ – if $\alpha(v)$ is a node, then $\alpha_{\downarrow}(v)$ is $\alpha(v)$ itself. Similarly we define $\beta_{\downarrow}(w)$, for a node $w \in V_g$.

The following two properties of α_{\downarrow} and β_{\downarrow} will be crucial for the decision procedure.

Lemma 5.2. (i) For a node $v \in V_f$, $|f(v) - g(\alpha_\downarrow(v))| \leq \varepsilon$, and (ii) for a node $w \in V_g$, $|g(w) - f(\beta_\downarrow(w))| \leq \varepsilon$.

Proof. We will prove part (i); part (ii) is similar. By definition, $g(\alpha_\downarrow(v)) \leq f(v) + \varepsilon$. Suppose $g(\alpha_\downarrow(v)) < f(v) - \varepsilon$. Let v' be a point in M_g lying on the edge containing $\alpha(v)$ and $\alpha_\downarrow(v)$ with height $f(v) - \varepsilon - \varepsilon_0$, for some sufficiently small ε_0 . Then $\beta(v')$ lies in one of the subtrees rooted at the children of v , say M_1 . Consider a descendant u of v at height $g(v') - \varepsilon$ lying in a different subtree M_2 rooted at v 's child. Since by definition and our choice of v' there does not exist any node in M_g between $\alpha(v)$ and v' , we have $\alpha(u) = v'$. But then $\beta(\alpha(u)) = \beta(v')$ lies in M_1 , and hence is not an ancestor of $u \in M_2$; in other words $\beta(\alpha(u)) \neq \sigma_f^{2\varepsilon}(u)$. This contradicts the fact that α and β are ε -compatible. Thus, $g(\alpha_\downarrow(v)) \geq f(v) - \varepsilon$, and the claim follows. \square

Lemma 5.3. If all edges in M_f and M_g are ε -long, then α_\downarrow and β_\downarrow are bijections with $\beta_\downarrow = \alpha_\downarrow^{-1}$ (and $\alpha_\downarrow = \beta_\downarrow^{-1}$).

Proof. We will first show that $\beta_\downarrow = \alpha_\downarrow^{-1}$. Suppose to the contrary there exists a vertex $v \in V_f$ such that $\beta_\downarrow(\alpha_\downarrow(v)) = w \neq v$. Let $\alpha_\downarrow(v) = u$, for $u \in V_g$. From Lemma 5.2 we have $|f(v) - f(w)| \leq 2\varepsilon$. Since all edges are longer than 2ε and $v \neq w$, v cannot be an ancestor/descendant of w in M_f . By definition of α_\downarrow , $\alpha(v)$ is an ancestor of $\alpha_\downarrow(v) = u$. Thus $\beta(\alpha(v))$ is an ancestor of $\beta(u)$. Further, $|f(v) - g(u)| \leq \varepsilon$ (Lemma 5.2) and $\beta(\alpha(v)) = \sigma_f^{2\varepsilon}(v)$ (since α, β are ε -compatible). Hence, $\beta(u)$ lies between v and $\beta(\alpha(v))$ on the edge e whose lower endpoint is v as e is ε -long. Thus, $\beta(u)$ is an ancestor of v . Also by definition of β_\downarrow , $\beta(u)$ is an ancestor of $\beta_\downarrow(u) = w$. Thus, w is a descendant of v , a contradiction since v cannot be an ancestor of w .

We thus have $\beta_\downarrow = \alpha_\downarrow^{-1}$. Similarly, we can show that $\alpha_\downarrow = \beta_\downarrow^{-1}$. This also implies that α_\downarrow and β_\downarrow are bijections. \square

We define an indicator function $\Phi : V_f \times V_g \rightarrow \{0, 1\}$ such that

$$\Phi(u, v) = \begin{cases} 1, & \text{if } d_I(M_f^u, M_g^v) \leq \varepsilon, \\ 0, & \text{otherwise.} \end{cases}$$

The following lemma gives a recursive definition of $\Phi(u, v)$.

Lemma 5.4. Suppose all the edges in M_f and M_g are ε -long. For a pair $(u, v) \in V_f \times V_g$, $\Phi(u, v) = 1$ if and only if the following conditions hold : (i) $|f(u) - g(v)| \leq \varepsilon$, (ii) $|C(u)| = |C(v)|$, and (iii) there exists a permutation π of $[1 : |C(u)|]$ such that $\Phi(u_i, v_{\pi(i)}) = 1$ for all $i \in [1 : |C(u)|]$.

Proof. Suppose $\Phi(u, v) = 1$, and let α, β be the corresponding ε -compatible maps. To see why (i) holds, for contradiction, suppose property (i) does not hold, and let $f(u) > g(v)$ without loss of generality. Thus, $\beta(v)$ maps to one of the multiple edges incident to u , and there exists at least one edge $e = (u, w)$ with $w \in C(u)$ such that none of e 's points (other than u) is in the image of β . However, $\beta(\alpha(u)) = \sigma_f^{2\varepsilon}(u)$ must lie in the interior of e (since e is ε -long), a contradiction. To prove that (ii) holds, note that by Lemma 5.3, there exist bijections $\alpha_\downarrow, \beta_\downarrow$ between V_f, V_g such that if $u_1 \in C(u_2)$ in V_f , then $\alpha_\downarrow(u_1) \in C(\alpha_\downarrow(u_2))$ (a symmetric statement holds for β_\downarrow and V_g). Thus, $\alpha_\downarrow, \beta_\downarrow$ induce bijections between $C(u)$ and $C(v)$, and hence $|C(u)| = |C(v)|$. Finally, for (iii), Let $\alpha_\downarrow(u') = v'$ for some $u' \in C(u), v' \in C(v)$. Then by definition of α_\downarrow and β_\downarrow , $\alpha(M_f^{u'}) \subseteq M_g^{v'}$ and $\beta(M_g^{v'}) \subseteq M_f^{u'}$. This means that the restriction of the pair of ε -compatible maps α and β to $M_f^{u'}$ and $M_g^{v'}$ respectively remain ε -compatible for $M_f^{u'}$ and $M_g^{v'}$. Thus, $\Phi(u', v') = 1$, and the permutation π is defined by $\alpha_\downarrow, \beta_\downarrow$.

We now prove the opposite direction. Suppose properties (i),(ii) and (iii) hold. Let (α_i, β_i) be the pair of ε -compatible maps between $M_f^{u_i}$ and $M_g^{v_{\pi(i)}}$. Then, a pair of ε -compatible maps (α, β) between M_f^u and M_g^v is obtained as follows : $\alpha(x) = \{\alpha^i(x) \mid x \in M_f^{u_i}\}$ (β is defined similarly). Note that points on the infinite edge from u (resp. v) upwards are *shared* among all $M_f^{u_i}$ (resp. $M_g^{v_i}$), whereas all other points in $M_f^{u_i}$ (resp. $M_g^{v_i}$) are present in only one $M_f^{u_i}$ (resp. $M_g^{v_i}$). However, since $|f(u) - g(v)| \leq \varepsilon$, *shared* points are mapped to *shared*

points and we have $|\alpha(x)| = 1$ (resp. $|\beta(y)| = 1$) for all $x \in M_f^u$ (resp. $y \in M_g^v$). Thus, α and β are functions and satisfy all the required properties. Hence, $\Phi(u, v) = 1$. \square

Decision procedure. We compute Φ for all pairs of nodes in $V_f \times V_g$ in a bottom-up manner and return $\Phi(r_f, r_g)$ where r_f (resp. r_g) is the root of M_f (resp. M_g). Let $(u, v) \in V_f \times V_g$.

Suppose we have computed $\Phi(u_i, v_j)$ for all $u_i \in C(u)$ and $v_j \in C(v)$. We compute $\Phi(u, v)$ as follows. If (i) or (ii) of Lemma 5.4 does not hold for u and v , then we return $\Phi(u, v) = 0$. Otherwise we construct the bipartite graph $G_{uv} = \{C(u) \cup C(v), E = \{(u_i, v_j) \mid \Phi(u_i, v_j) = 1\}\}$ and determine in $O(k^{5/2})$ time whether G_{uv} has a perfect matching, using the algorithm by Hopcroft and Karp [12]. Here, $k = |C(u)| = |C(v)|$. If G_{uv} has a perfect matching $M = \{(u_1, v_{\pi(1)}), \dots, (u_k, v_{\pi(k)})\}$, we set $\Phi(u, v) = 1$, else we set $\Phi(u, v) = 0$. If $\Phi(u, v) = 1$, we use the ε -compatible maps for $M_f^{u_i}, M_g^{v_{\pi(i)}}$, for $1 \leq i \leq k$, to compute a pair of ε -compatible maps between M_f^u and M_g^v , as discussed in the proof of Lemma 5.4.

For a node $u \in V_f \cup V_g$, let k_u be the number of its children. The total time taken for running Hopcroft and Karp [12] is :

$$\sum_{u \in V(T_1)} \sum_{v \in V(T_2)} O(k_u k_v \sqrt{k_v}) = \sum_{u \in V(T_1)} k_u \sum_{v \in V(T_2)} O(k_v \sqrt{k_v}) \leq O(n^{3/2}) \sum_{u \in V(T_1)} k_u \leq O(n^{5/2}).$$

Hence we obtain the following.

Lemma 5.5. *Given two merge trees M_f and M_g and a parameter $\varepsilon > 0$ such that all edges of M_f and M_g are ε -long, then whether $d_I(M_f, M_g) \leq \varepsilon$ can be determined in $O(n^{5/2})$ time. If the answer is yes, a pair of ε -compatible maps between M_f and M_g can be computed within the same time.*

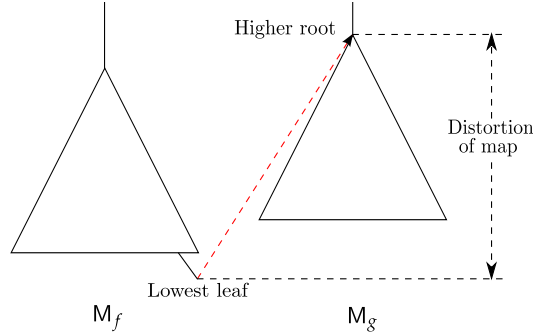


Figure 7. A naive map.

Trees with short edges. Given two merge trees, a naive map is to map the lowest among all the leaves in both the trees to a point at height equal to the height of the higher root (see Figure 7). Thus, all the points in one tree will be mapped to the infinitely long edge on the other tree. This map produces a distortion equal to the height of the trees, which can be arbitrarily larger than the optimum. Nevertheless, this simple idea leads to an approximation algorithm.

Here is an outline of the algorithm. After carefully *trimming* off short subtrees from the input trees, the algorithm decomposes the resulting trimmed trees into two kinds of regions – those with nodes and those without nodes. If the interleaving distance between the input trees is small, then there exists an isomorphism between trees induced by the regions without nodes. Using this isomorphism, the points in the nodeless regions are mapped without incurring additional distortion. Using a counting argument and the naive map described above, it is shown that the distortion incurred while mapping the regions with nodes and the trimmed regions is bounded.

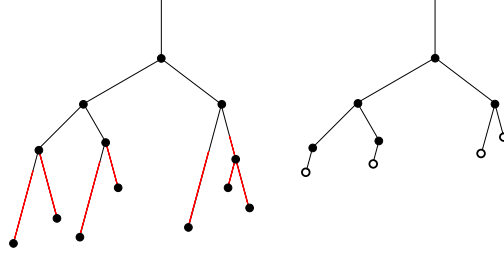


Figure 8. Trimming a tree : (left) original tree, red points have extent $< 2(\sqrt{2ns} + 1)\epsilon$; (right) trimmed tree, nodes added at the bottom (hollowed nodes).

More precisely, given M_f , M_g and $\epsilon > 0$, define the *extent* $e(x)$ of a point x (which is not necessarily a tree node) in M_f or M_g as the maximum height difference between x and any of its descendants. Suppose each edge is at most ϵ long. Let M'_f and M'_g be subsets of M_f and M_g consisting only of points with extent at least $2(\sqrt{2ns} + 1)\epsilon$, adding nodes to the new leaves of M'_f and M'_g as necessary. Note that M'_f and M'_g themselves are trees, however they might contain nodes of degree 2. See Figure 8 for an example.

Lemma 5.6. *If $d_I(M_f, M_g) \leq \epsilon$, then $d_I(M'_f, M'_g) \leq \epsilon$.*

Proof. Let $\alpha : M_f \rightarrow M_g$ and $\beta : M_g \rightarrow M_f$ be ϵ -compatible maps. Let α' and β' be restrictions of the functions' domains to M'_f and M'_g respectively. We argue that the ranges of α' and β' lie in M'_g and M'_f respectively. Suppose otherwise. Then without loss of generality, there is a point $x \in M'_f$ with $y = \alpha(x)$ not in M'_g . Because $x \in M'_f$, its extent in M_f is at least $2(\sqrt{2ns} + 1)\epsilon$. Therefore, there exists a descendant x' of x in M_f with $f(x') = f(x) - 2(\sqrt{2ns} + 1)\epsilon$. Because y is not in M'_g , the extent of y must be less than $2(\sqrt{2ns} + 1)\epsilon$ and there exists no descendant y' of y with $g(y') = g(y) - 2(\sqrt{2ns} + 1)\epsilon = f(x) - 2(\sqrt{2ns} + 1)\epsilon + \epsilon = f(x') + \epsilon$. Since $g(\alpha(x')) = f(x') + \epsilon$, $\alpha(x')$ is not a descendant of $\alpha(x)$, which contradicts the assumption that α, β are ϵ -compatible maps. \square

The above lemma can be easily generalized to say that removing points in both trees with extent less than or equal to any fixed value does not change the distance between them.

We now define *matching points* in M'_f and M'_g . A *branching node* is a node of degree greater than 2. A point x in M'_f is a matching point if there exists a branching node or a leaf x' in M'_f or y' in M'_g with function value $f(x)$ and there exist no branching nodes nor leaves in M'_f or M'_g with function value in the range $(f(x), f(x) + 2\epsilon]$. Matching points on M'_g are defined similarly. By this definition, no two matching points share a function value within 2ϵ of each other unless they share the exact same function value. Furthermore, if x is a matching point, then all points with the same function value as x on both M'_f and M'_g are matching points. There are $O(n^2)$ matching points.

Suppose $d_I(M'_f, M'_g) \leq \epsilon$, and let $\alpha' : M'_f \rightarrow M'_g$ and $\beta' : M'_g \rightarrow M'_f$ be a pair of ϵ -compatible maps. Call a matching point x in M'_f and a matching point y in M'_g with $f(x) = g(y)$ *matched* if $\alpha'(x)$ is an ancestor of y .

Lemma 5.7. *Let x be any matching point in M'_f . The matched relation between matching points in M'_f at height $f(x)$ and matching points in M'_g at height $f(x)$ is a bijective function.*

Proof. No two distinct matching points y_1 and y_2 on M'_g with $f(x) = g(y_1) = g(y_2)$ share the same ancestor with function value $f(x) + \epsilon$, because they have no branching node ancestors with low enough function value. Therefore, a matching point in M'_f can be matched to only one one matching point in M'_g .

Let x_1 and x_2 be two distinct matching points from M'_f with $f(x) = f(x_1) = f(x_2)$. If $\alpha'(x_1)$ and $\alpha'(x_2)$ are ancestors of a common matching point y , then $\alpha'(x_1) = \alpha'(x_2)$ and thus x_1 and x_2 must have a common

ancestor x' at height $f(x) + 2\varepsilon$. However, x_1 and x_2 have no branching node ancestor with low enough function value for x' to exist. Hence, the matching relation must be injective from matching points in M'_f to M'_g .

Finally, consider any matching point y on M'_g with $g(y) = f(x)$. Point $x'_1 = \beta'(y)$ is the ancestor of a matching point x_1 on M'_f . (Note that by the same argument as the beginning of this proof, only one such matching point x_1 can exist.) Point $y' = \alpha'(x'_1)$ is an ancestor of y with $g(y') \leq g(y) + 2\varepsilon$. Point y is the only descendant of y' with function value $f(x)$. Point $\alpha'(x_1)$ must be an ancestor of y , meaning x_1 and y are matched. Thus, the matching relation is surjective. \square

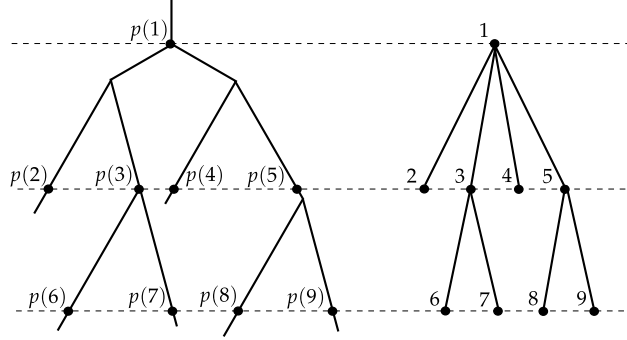


Figure 9. The left tree shows matching points on tree M'_f and the right tree shows \tilde{M}_f .

We now define a rooted tree \tilde{M}_f to be a rooted tree consisting of one node per matching point on M'_f . Let $p(v)$ be the matching point for node v . \tilde{M}_f has node v as an ancestor of node u if $p(v)$ is an ancestor of $p(u)$ (see Figure 9). Define \tilde{M}_g similarly. The size of \tilde{M}_f and \tilde{M}_g is $O(n^2)$. Intuitively, \tilde{M}_f and \tilde{M}_g represent the trees induced by matching points. By the definition of interleaving distance and Lemma 5.7, \tilde{M}_f and \tilde{M}_g are isomorphic if M'_f and M'_g satisfy that $d_I(M'_f, M'_g) \leq \varepsilon$.

Decision procedure. We are now ready to describe the decision procedure. We first construct the subtrees M'_f and M'_g of M_f and M_g , respectively, consisting of points with extent at least $2(\sqrt{2ns} + 1)\varepsilon$. Next, we compute matching points on M'_f and M'_g and construct the trees \tilde{M}_f and \tilde{M}_g on these matching points, as defined above.

Using the algorithm of [1, chap. 3, p. 85], we determine in time linear in the size of the trees whether \tilde{M}_f and \tilde{M}_g are isomorphic. If the answer is no, we return no. By Lemma 5.7, $d_I(M_f, M_g) > \varepsilon$ in this case. Otherwise we construct the following functions $\alpha : M_f \rightarrow M_g$ and $\beta : M_g \rightarrow M_f$ and return them. For each pair of matching points x and y matched by the isomorphism, the algorithm sets $\alpha(x) = y$ and $\beta(y) = x$. Now, let (ξ_1, ξ_2) be any maximal range of function values without any branching nodes or leaves in M'_f or M'_g with $\xi_2 - \xi_1 > 2\varepsilon$. Let x' be any point in M'_f with $f(x') \in (\xi_1, \xi_2)$. Point x' has a unique matching point descendant x at height ξ_1 , by the definition of matching points. The algorithm sets $\alpha(x')$ to the point y' in M'_g where y' is the ancestor of $\alpha(x)$ with $g(y') = f(x')$, and it sets $\beta(y') = x'$. For every remaining point x'' in M'_f , the algorithm sets $\alpha(x'')$ to $\alpha(x)$ where x is the lowest matching point ancestor of x'' . $\beta(y'')$ is defined similarly for remaining points y'' in M'_g . We call such points x'' and y'' *lazily assigned*. Finally, each point z in $M_f - M'_f$ has $\alpha(z)$ set to $\alpha(x)$ where x is the lowest ancestor of z in M'_f . Similar assignments are done for points in $M_g - M'_g$.

Lemma 5.8. (i) For each lazily assigned point $x'' \in M'_f$,

$$g(\alpha(x'')) \leq f(x'') + 2(\sqrt{2ns} + 1)\varepsilon.$$

(ii) For each lazily assigned point $y'' \in M'_g$,

$$f(\beta(y'')) \leq g(y'') + 2(\sqrt{2ns} + 1)\varepsilon.$$

Proof. We only prove (i); (ii) is symmetric. The higher of the two roots of M'_f and M'_g is a matching point, and so are all the points at that height. Thus, all lazily assigned points have a matching point ancestor. We show that the nearest such ancestor cannot be too higher up.

Let x be a matching point. We show that there exists a region (ξ_1, ξ_2) as defined above with

$$f(x) - 2(\sqrt{2ns} + 1)\varepsilon \leq \xi_2 \leq f(x).$$

Consider sweeping over the function values downward starting at $f(x)$ and let ξ_2 be the largest function value possible for a region as defined above. If the sweep line ever goes a distance greater than 2ε without encountering a branching node or leaf in M'_f or M'_g , then an ξ_2 is found. Therefore, there will be at least one branching node or leaf x' in M'_f or M'_g per descent of 2ε until ξ_2 is found. Suppose $\xi_2 < f(x) - 2(\sqrt{2ns} + 1)\varepsilon$. Let $l = \sqrt{2ns} + 1$, and $f' = f(x) - 2l\varepsilon$. Because each point in M'_f and M'_g has extent at least $2l\varepsilon$, each branching node or leaf $x' \in M'_f \cup M'_g$ with height $f(x') \in [f', f(x))$ introduces at least one descendant in M_f or M_g at height f' . Since each edge length is at most $s\varepsilon$, we can uniquely charge at least $(f(x') - f')/s\varepsilon$ nodes to each branching node x' at height $f(x') \in [f', f(x))$. For each leaf node x' , we can still charge $(f(x') - f')/s\varepsilon$ nodes to x' ; however these nodes can also be charged to at most one more branching node, namely the lowest branching node ancestor of x' with height in $[f', f(x))$. Thus, each node can be charged at most twice. Since there is at least one branching node or leaf per descent of 2ε , the total number of nodes charged is at least

$$\sum_{i=1}^l \frac{(l-i)2\varepsilon}{s\varepsilon} = \frac{l(l-1)}{s}.$$

Since each node is charged at most twice, we have

$$\frac{l(l-1)}{s} \leq 2n \Rightarrow l(l-1) \leq 2ns,$$

a contradiction since $l = \sqrt{2ns} + 1$. Therefore, either $\xi_2 \geq f(x) - 2(\sqrt{2ns} + 1)\varepsilon$, or the trees M'_f and M'_g do not extend below height $f(x) - 2(\sqrt{2ns} + 1)\varepsilon$. In either case, the lemma follows. \square

Lemma 5.9. *Let M_f and M_g be two merge trees and let $\varepsilon > 0$ be a parameter. There is an $O(n^2)$ time algorithm that returns a pair of $4(\sqrt{2ns} + 1)\varepsilon$ -compatible maps between M_f and M_g , if $d_I(M_f, M_g) \leq \varepsilon$ and the maximum length of a tree edge is $s\varepsilon$. If $d_I(M_f, M_g) > \varepsilon$, then the algorithm may return no or return a pair of $4(\sqrt{2ns} + 1)\varepsilon$ -compatible maps.*

Proof. Constructing the trees \tilde{M}_f and \tilde{M}_g , the corresponding isomorphism between them (if it exists), and the maps α and β between M_f and M_g (if they exist) takes time $O(n^2)$.

Except for the lazily assigned points, all the points in M'_f and M'_g are mapped by α and β resp. to points at the same function value. By Lemma 5.8, each point in M'_f and M'_g has its function value changed by at most $2(\sqrt{2ns} + 1)\varepsilon$. Points in $M_f - M'_f$ (resp. $M_g - M'_g$) have their nearest ancestors in M'_f (resp. M'_g) at function value at most $2(\sqrt{2ns} + 1)\varepsilon$ away. Since α and β map them to the images of their nearest ancestors, their function values change by at most $2 \cdot 2(\sqrt{2ns} + 1)\varepsilon$. \square

Remark. (i) Since the minimum edge length is $\leq 2\varepsilon$, the maximum edge length is $s\varepsilon$, and the ratio between the lengths of the longest and shortest edges is r ; we have $r \geq s/2$.

(ii) If $s = \Omega(n)$, we modify the above algorithm slightly – we skip the trimming step, but keep the rest same. It can be shown, as in Lemma 5.8, that the height of a point and its image differ by at most $2n\varepsilon$.

Putting it together. By Lemmas 5.5 and 5.9, the decision procedure takes $O(n^{5/2})$ time. If it returns no, then $d_I(M_f, M_g) > \varepsilon$. If it returns yes, then it also returns $O(\min\{n, \sqrt{rn}\}\varepsilon)$ -compatible maps between them. Hence, we conclude the following.

Theorem 5.10. *Given two merge trees M_f and M_g with a total of n vertices, there exists an $O(n^{5/2} \log n)$ time $O(\min\{n, \sqrt{rn}\})$ -approximation algorithm for computing the interleaving distance between them, where r is the ratio between the lengths of the longest and the shortest edge in both trees.*

Combining Theorem 5.10 with Corollary 4.5, we have:

Corollary 5.11. *Given two metric trees T_1 and T_2 with a total of n vertices, there exists an $O(n^{7/2} \log n)$ time $O(\min\{n, \sqrt{rn}\})$ -approximation algorithm for computing the Gromov-Hausdorff distance between them, where r is the ratio between the lengths of the longest and the shortest edge in both trees.*

6 Conclusion

We have presented the first hardness results for computing the Gromov-Hausdorff distance between metric trees. We have also given a polynomial time approximation algorithm for the problem. But the current gap between the lower and upper bounds on the approximation factor is polynomially large. It would be very interesting to close this gap. In general, we hope that our current investigation will stimulate more research on the theoretical and algorithmic aspects of embedding or matching under additive metric distortion.

References

- [1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
- [2] U. Bauer, X. Ge, and Y. Wang. Measuring distance between Reeb graphs. In *Proc. Symp. Comput. Geom.*, pages 464–473, 2014.
- [3] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Sci. Comp.*, 28(5):1812–1836, 2006.
- [4] D. Burago, Y. Burago, and S. Ivanov. *A Course in Metric Geometry*. American Mathematical Society, 2001.
- [5] G. Carlsson and F. Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *J. Mach. Learn. Res.*, 11:1425–1470, 2010.
- [6] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discr. Comput. Geom.*, 37(1):103–120, 2007.
- [7] V. De Silva, E. Munch, and A. Patel. Categorized reeb graphs. *Discr. Comput. Geom.*, 55(4):854–906, 2016.
- [8] T. K. Dey, D. Shi, and Y. Wang. Comparing graphs via persistence distortion. *CoRR*, abs/1503.07414, 2015.
- [9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [10] M. Gromov. *Metric Structures for Riemannian and Non-Riemannian Spaces*. Birkhäuser Basel, 2007.
- [11] A. Hall and C. Papadimitriou. Approximating the distortion. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 111–122. Springer, 2005.

- [12] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comp.*, 2(4):225–231, 1973.
- [13] C. Kenyon, Y. Rabani, and A. Sinclair. Low distortion maps between point sets. *SIAM J. Comp.*, 39(4):1617–1636, 2009.
- [14] F. Mémoli. On the use of gromov-hausdorff distances for shape comparison. In *Proc. Symp. Point Based Graphics*, pages 81–90, 2007.
- [15] F. Mémoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Found. Comput. Math.*, 5(3):313–347, 2005.
- [16] D. Morozov, K. Beketayev, and G. Weber. Interleaving distance between merge trees. *Discr. Comput. Geom.*, 49:22–45, 2013.
- [17] C. Papadimitriou and S. Safra. The complexity of low-distortion embeddings between point sets. In *Proc. ACM-SIAM Symp. Discr. Algo.*, pages 112–118, 2005.