# 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 distance is a natural way to measure distance between two metric spaces. We give the first proof of hardness and first non-trivial approximation algorithm for computing the Gromov-Hausdorff distance for geodesic metrics in trees. Specifically, we prove it is NP-hard to approximate the Gromov-Hausdorff distance better than a factor of 3. We complement this result by providing a polynomial time $O(\min\{n, \sqrt{rn}\})$-approximation algorithm 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 into another using a correspondence between their points. Multiple approaches have been proposed to estimate the Gromov-Hausdorff distance or provide alternatives to its computation [3, 14, 15].

Despite much effort, the problem of computing, either exactly or approximately, GH distance has remained elusive. On one hand, the problem is not known to be NP-hard, and on the other hand no polynomial-time approximation algorithm exists for graphic metrics [1] unless the graph isomorphism problem is in P. (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.

**Our results.** In this paper, we give the first non-trivial results on approximating the GH distance between metric trees. First, we prove (in Sect. 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 present here for arbitrary $r$ is a non-trivial task.

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

**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. See [11, 13, 17] for recent surveys.

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.

# 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 $\rho : X \times X \to \mathbb{R}_{\geq 0}$ 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) $\forall a \in A, \exists b \in B$ such that $(a, b) \in \mathcal{C}$; and (ii)$\forall b \in B, \exists a \in A$ 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:

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

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

[2] In fact, our hardness result can be easily extended to the GH distance between discrete tree metrics and the interleaving distance between merge trees.

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) embeddding 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 $l : E \to \mathbb{R}_{\geq 0}$, 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 $f : \mathbb{X} \to \mathbb{R}$ 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* $T_f$ 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 epi$f = \{(x, y) \in \mathbb{X} \times \mathbb{R} \mid y \geq f(x)\}$. Let $\bar{f} : \text{epi}f \to \mathbb{R}$ be such that $\bar{f}((x, y)) = y$. We may say $\bar{f}((x, y))$ 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 $\bar{f}^{-1}(y) (= \bar{f}^{-1}(y'))$. Then $\sim$ is an equivalence relation, and the merge tree $T_f$ is defined as the quotient space $(\mathbb{X} \times \mathbb{R})/\sim$.

Since two components 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 of $\infty$, 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 $T_f$ to be the node with the highest function value.

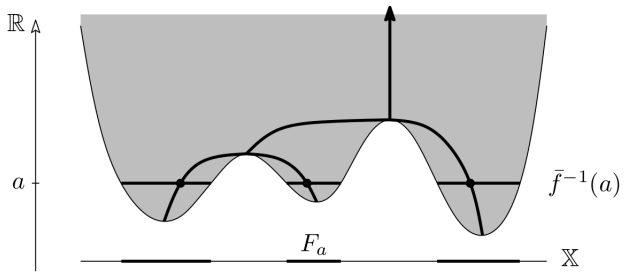


Figure 1: Merge tree for a function from $\mathbb{R} \to \mathbb{R}$ (image by Morozov et al. [16]).

Since each point $x \in T_f$ represents a component of a sublevel set at a certain height, we can associate a height value $\hat{f}(x)$ with $x$. Given a merge tree $T_f$ and $\epsilon \geq 0$, an $\epsilon$-shift map $i^\epsilon : T_f \to T_f$ is the map that maps a point in the tree to its ancestor at height $\epsilon$ higher. We thus have $\hat{f}(i^\epsilon(x)) = \hat{f}(x) + \epsilon$. Given $\epsilon \geq 0$ and merge trees $T_f$ and $T_g$ with the associated shift maps $i^\epsilon$ and $j^\epsilon$ respectively, two continuous maps $\alpha^\epsilon : T_f \to T_g$ and $\beta^\epsilon : T_g \to T_f$ are said to be $\epsilon$-compatible if they satisfy the following conditions

$$\hat{g}(\alpha^\epsilon(x)) = \hat{f}(x) + \epsilon, \forall x \in T_f \ ; \qquad \hat{f}(\beta^\epsilon(y)) = \hat{g}(y) + \epsilon, \forall y \in T_g \ ;$$
$$\beta^\epsilon \circ \alpha^\epsilon = i^{2\epsilon} \ ; \qquad \alpha^\epsilon \circ \beta^\epsilon = j^{2\epsilon} \ .$$

The *interleaving distance* [16] is then defined as

$$d_I(T_f, T_g) = \inf\{\epsilon \mid \text{there exist } \epsilon\text{-compatible maps } \alpha^\epsilon \text{ and } \beta^\epsilon\} \ .$$

**Remark.** We can relax the requirements on $\alpha^\epsilon$ and $\beta^\epsilon$ from their normal definitions. Instead of requiring *exact* value changes, we require $\hat{f}(x) \leq \hat{g}(\alpha^\epsilon(x)) \leq \hat{f}(x) + \epsilon$ and $\hat{g}(y) \leq \hat{f}(\beta^\epsilon(y)) \leq \hat{g}(y) + \epsilon$. In addition, as $x$ moves toward the root of $T_f$, $\alpha^\epsilon(x)$ must move toward the root of $T_g$ (although $\alpha^\epsilon(x)$ may remain constant for a range of $x$ values) and we do not need them to be continuous. A similar rule

applies for $\beta^\epsilon$. Finally, $\beta^\epsilon(\alpha^\epsilon(x))$ must go to an ancestor of $x$ and $\alpha^\epsilon(\beta^\epsilon(y))$ must go to an ancestor of $y$. Both definitions of interleaving distance are equivalent, and we may use either based on which is more convenient.

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].

In the remainder of the paper, we will frequently drop the superscript $\epsilon$ when it is clear from the context. Also, we may stop alluding to the underlying functions $f$ and $g$ of the merge trees $T_f$ and $T_g$ and simply refer to them as $T_1$ and $T_2$. We may also use $f$ and $g$ to sometimes denote the height of the points in the trees themselves.

# 3 Hardness of Approximation

We show a reduction from the following decision problem called UNRESTRICTED-PARTITION: given a multiset of positive integers $X = \{a_1, \ldots, a_n\}$ with $n = 3m$, is it possible to partition them into $m$ multisets $\{X_1, \ldots, X_m\}$ such that all the elements in each multiset sum to the same quantity $S = \left(\sum_{i=1}^{n} a_i\right)/m$. This problem can be proved to be strongly NP-complete, so we can assume that the size of the integers is polynomial in the input.

**Lemma 3.1.** UNRESTRICTED-PARTITION *is strongly NP-complete.*

*Proof.* We reduce from the 3-PARTITION problem which is known to be strongly NP-complete [9] – given a multiset of positive integers $Y = \{a_1, \ldots, a_n\}$ with $n = 3m$, partition it into $m$ multisets $\{Y_1, \ldots, Y_m\}$ 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 UNRESTRICTED-PARTITION 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 $M = \sum_{i=1}^{n} a_i$. Then set $a'_i = a_i + M$ and $X = \{a'_1, \ldots, a'_n\}$. 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 $\{Y_1, \ldots, Y_m\}$ of $Y$. Then setting $X_i = \{a'_j \mid a_j \in Y_i\}$ for $i = 1, \ldots, m$ gives us the desired partition of $X$.

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

$$\begin{aligned} &\sum_{a'_k \in X_i} a_k + n_1 M = \sum_{a'_k \in X_j} a_k + n_2 M \\ \Rightarrow &(n_1 - n_2) M = \sum_{a'_k \in X_i} a_k - \sum_{a'_k \in X_j} a_k \\ \Rightarrow &\sum_{a'_k \in X_i} a_k - \sum_{a'_k \in X_j} a_k \geq M \ , \end{aligned}$$

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

We construct two trees $T_1$ and $T_2$ as follows. Let $A$ and $B$ be two sufficiently large numbers. Let $T_{s,t}$ denote a star graph having $t$ edges of length $s$. $T_1$ consists of a node $r_1$ incident to an edge $(r_1, r'_1)$ of length $B$ and to $n$ edges $\{(r_1, p_1), \ldots, (r_1, p_n)\}$ of length 1, where $p_i$ is the center of a copy of $T_{A,a_i}$. $T_2$ consists of a node $r_2$ incident to an edge $(r_2, r'_2)$ of length $B$ and to $m$ edges $\{(r_2, q_1), \ldots, (r_2, q_m)\}$ of length 2, where each $q_i$ is the center of a distinct copy of $T_{A+1,S}$. See Figure 2 for an illustration. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ denote the metric trees associated with $T_1$ and $T_2$ respectively. Clearly, this construction can be done in polynomial time.

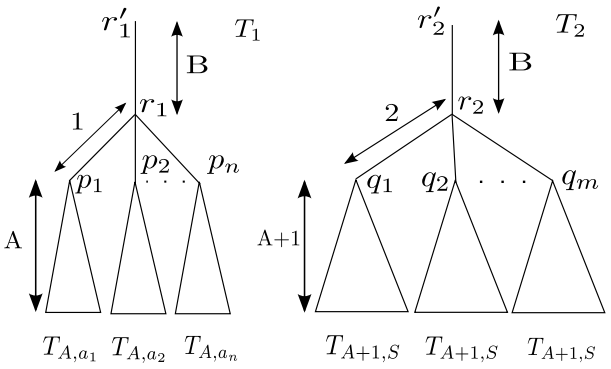


Figure 2: The trees $T_1$ and $T_2$.

**Lemma 3.2.** *If the given instance of* UNRESTRICTED-PARTITION *is a* yes *instance, then* $d_{GH}(\mathcal{T}_1, \mathcal{T}_2) \leq 1$. *Otherwise,* $d_{GH}(\mathcal{T}_1, \mathcal{T}_2) \geq 3$.

*Proof.* (*Yes* instance) We construct a correspondence $\mathcal{C}$ between $\mathcal{T}_1$ and $\mathcal{T}_2$ with distortion at most 2, hence distance at most 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 $a_i$ is assigned to $X_j$, the linearly interpolated bijection between edges $(r_1, p_i)$ and $(r_2, q_j)$ is added to $\mathcal{C}$. Also, the leaves of $T_{A,a_i}$ are each mapped to a distinct leaf of $T_{A+1,S}$ attached to $q_j$ such that there is a bijection between the leaves of $T_1$ and $T_2$ – this can be done since we have a *yes* instance. The interior points of the edges are mapped using linear interpolation. It can be easily verified that the distortion induced by $\mathcal{C}$ is at most 2.

(*No* instance) We show that any correspondence induces a distortion of at least 6, hence distance at least 3. Assume $A$ and $B$ are large enough so that for any correspondence with distortion $\leq 6$, we can construct a bijection between the leaf edges of $T_1$ and $T_2$ such that two leaf edges are related if the correspondence sends the leaf of one edge to a point on the other edge, with $(r_1, r_1')$ mapping to $(r_2, r_2')$. Since we have a *no* instance, either no such bijection exists or there exists an $i$ such that two leaves of $T_{A,a_i}$ map to points inside leaf edges in $T_{A+1,S}$ attached to $q_{j_1}$ and $q_{j_2}$, for some $j_1 \neq j_2$. Then the corresponding leaves attached to $q_{j_1}$ and $q_{j_2}$ (say $l_1$ and $l_2$ resp.) must map to points $l_1'$ and $l_2'$ inside $T_{A,a_i}$ in $T_1$. We then have $d_1(l_1', l_2') \leq 2A$ while $d_2(l_1, l_2) = 2A + 6$. The distortion is at least 6. □

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 Relating Gromov-Hausdorff and Interleaving Distances

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 $f_s : T \to \mathbb{R}$ be defined as $f_s(x) = -\, d(s, x)$. Equipped with this function, we obtain a merge tree $T_{f_s}$ from $\mathcal{T}$. Intuitively, $T_{f_s}$ has the structure of rooting $T$ at $s$, and then adding an extra edge incident to $s$ with function value extending to $+\infty$. The following theorem connects the GH distance and the interleaving distance.

**Theorem 4.1.** *Let* $\gamma = \min_{u \in V(T_1), v \in V(T_2)} d_I(T_{1f_u}, T_{2f_v})$. *Then*

$$\tfrac{1}{2} d_{GH}(\mathcal{T}_1, \mathcal{T}_2) \leq \gamma \leq 10 d_{GH}(\mathcal{T}_1, \mathcal{T}_2) \ .$$

*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 $\delta$), and taking a certain limit under it.

(1) *Proof of the right inequality.*

Consider an arbitrary pair $(s, t) \in \mathcal{C}^*$ (note that $s, t$ may not be tree nodes). Set $f = f_s$ which is the negation of the geodesic distance function to the base point $s \in T_1$. Similarly, set $g$ to be the negation of the geodesic distance function $g_t$ to the base point $t \in T_2$. We thus get the merge trees $T_{1f}$ and $T_{2g}$ rooted at $s$ and $t$ respectively. Using Lemma 5 of [8] and the equivalence between interleaving and functional distortion distance between merge trees [3], we have

$$d_I(T_{1f}, T_{2g}) \leq 6\delta \ . \tag{1}$$

Now let $v_1, v_2 \in V(T_1)$ be the two tree nodes (leaves) such that $d_1(v_1, v_2)$ is the diameter of the tree $T_1$. Set $s$ to be one of them; say $s = v_1$. Consider a pair $(s, t) \in \mathcal{C}^*$ from the optimal correspondence. We aim to prove that there is a tree node $\tilde{t} \in V(T_2)$ that is within $c\delta$ distance to $t$ for a constant $c = 4$.

Indeed, assume that there is no tree node within $c\delta$ distance to $t$. In this case, $t$ must be in the interior of a tree 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 $\|t - u_1\| = \|t - u_2\| = c\delta + \nu$, where $\nu$ is an arbitrarily small positive value – both $u_1$ and $u_2$ exist as

[3] This equivalence will be shown in the journal version of [2].

there is no tree node of $T_2$ within $c\delta$ distance to $t$. 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 $d_2(u_1, u_2) = \|t - u_1\| + \|t - u_2\| = 2c\delta + 2\nu$ and $\mathcal{C}^*$ is an optimal correspondence, we have that

$$d_1(\tilde{u}_1, \tilde{u}_2) \in [(2c-1)\delta + 2\nu, (2c+1)\delta + 2\nu] \ . \tag{2}$$

On the other hand, since $\|t - u_1\| = \|t - u_2\| = c\delta + \nu$, we have that

$$d_1(s, \tilde{u}_1) \in [(c-1)\delta + \nu, (c+1)\delta + \nu] \ \text{ and } \ d_1(s, \tilde{u}_2) \in [(c-1)\delta + \nu, (c+1)\delta + \nu] \ . \tag{3}$$

We now aim to bound $d_1(\tilde{u}_1, \tilde{u}_2)$ in tree $T_1$.

Consider the tree $T_1$ rooted at $s$. If $\tilde{u}_1$ and $\tilde{u}_2$ have ancestor / descendant relation, then $d_1(\tilde{u}_1, \tilde{u}_2) = |d_1(s, \tilde{u}_1) - d_1(s, \tilde{u}_2)|$ and by (3), we thus have that $d_1(\tilde{u}_1, \tilde{u}_2) \leq 2\delta$. However, this contradicts (2) as $2c - 1 > 2$.

Now let $w$ be the common ancestor of $\tilde{u}_1$ and $\tilde{u}_2$ in $T_1$. 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

$$d_1(\tilde{u}_1, \tilde{u}_2) = a + b - 2c_0 \ . \ \text{Note,} \ \ a \geq c_0, b \geq c_0 \ . \tag{4}$$

Since $s = v_1$ and $v_2$ span the diameter of tree $T_1$, it follows that $c_0 \geq \min\{a - c_0, b - c_0\}$. Otherwise, assume the other point $v_2$ from the diameter pair is *not* from the subtree rooted at $\tilde{u}_1$. Then $d_1(\tilde{u}_1, v_2) > d_1(s, v_2)$, which contradicts that $s, v_2$ spans the diameter of $T_1$. By (3), $a, b \geq (c-1)\delta + \nu$. Thus

$$c_0 \geq (c-1)\delta + \nu - c_0 \quad \Rightarrow c_0 \geq \frac{1}{2}[(c-1)\delta + \nu] \ . \tag{5}$$

Combining (4) and (5), we have:

$$d_1(\tilde{u}_1, \tilde{u}_2) \leq a + b - (c-1)\delta - \nu \leq 2(c+1)\delta + 2\nu - (c-1)\delta - \nu = (c+3)\delta + \nu \ . \tag{6}$$

However, note that for $c = 4$, $c + 3 = 2c - 1$ and thus $(c+3)\delta + \nu < (2c-1)\delta + 2\nu$. In other words, (2) and (6) cannot be satisfied simultaneously. Contradiction. It then follows that there must exist a tree node, denoting it by $\tilde{t} \in V(T_2)$, such that $d_2(t, \tilde{t}) \leq 4\delta$.

Now consider the negation of the geodesic function $g' = -g_{\tilde{t}}$. It is easy to see that $\|g - g'\|_\infty = \|g_t - g_{\tilde{t}}\| \leq d_2(t, \tilde{t}) \leq 4\delta$. On the other hand, by the stability theorem of the interleaving distance (Theorem 2 of [16]), we have that $d_I(T_{2g}, T_{2g'}) \leq \|g - g'\|_\infty \leq 4\delta$. Combining this with (1) and by triangle inequality, it then follows that

$$d_I(T_{1f}, T_{2g'}) \leq 10\delta \ .$$

Since $f$ and $g'$ are induced by a pair of tree nodes $s \in V(T_1)$ and $\tilde{t} \in V(T_2)$, we thus have that $\gamma \leq d_I(T_{1f}, T_{2g'}) \leq 10\delta$. The right inequality of the claim thus follows.

(2) *Proof of the left inequality.*

Assume that $\gamma$ is achieved by a pair of tree nodes $s \in V(T_1)$ and $t \in V(T_2)$. Set $f := -f_s$ and $g := -g_t$ to be the negation of the respective geodesic distance functions, and $T_{1f}$ and $T_{2g}$ their corresponding merge trees. Consider the pair of optimal continuous maps $\alpha^* : T_{1f} \to T_{2g}$ and $\beta^* : T_{2g} \to T_{1f}$ that achieves $\gamma = d_I(T_{1f}, T_{2g})$. Recall that $T_{1f}$ has the structure of rooting $T_1$ at tree node $s$, and then adding an extra infinite edge incident to $s$ with function value extending to $+\infty$. In what follows, for simplicity, we ignore this infinite edge and assume that $T_{1f}$ is isomorphic to $T_1$ in terms of the tree structure. We make a similar assumption on $T_{2g}$. The argument can be easily modified to handle the two infinite edges in $T_{1f}$ and $T_{2g}$.

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

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

We now aim to prove that $\text{Dist}(\mathcal{C}) \leq 2\gamma$.

Indeed, consider any two pairs $(x_1, y_1), (x_2, y_2) \in \mathcal{C}$. Root tree $T_1$ at $s$, and similarly, root tree $T_2$ at $t$. Let $u$ be the common ancester of $x_1$ and $x_2$ in $T_1$, and $w$ the common ancester 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. A symmetric statement holds for $y_1 \rightsquigarrow w \rightsquigarrow y_2$. Hence

$$d_1(x_1, x_2) = d_1(x_1, u) + d_1(u, x_2) = 2f(u) - f(x_1) - f(x_2) \text{ and}$$
$$d_2(y_1, y_2) = d_2(y_1, w) + d_2(w, y_2) = 2g(w) - g(y_1) - g(y_2) \ .$$

We will consider the case where $y_1 = \alpha^*(x_1)$ and $y_2 = \alpha^*(x_2)$. The other cases are similar. Since $\alpha^*$ and $\beta^*$ are $\gamma$-compatible, we have $g(y_1) = f(x_1) + \gamma$ and $g(y_2) = f(x_2) + \gamma$. Thus,

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

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

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

It then follows that $\text{Dist}(\mathcal{C}) \leq 2\gamma$. Since $d_{GH}(\mathcal{T}_1, \mathcal{T}_2) \leq \text{Dist}(\mathcal{C})$, the left inequality then follows. □

**Corollary 4.2.** *If there is a $c$-approximation algorithm for the interleaving distance between two merge trees, then there is a $20c$-approximation algorithm for the Gromov-Hausdorff distance between two metric trees.*

# 5 Computing the Interleaving Distance

We propose algorithms for the decision version of the interleaving distance problem, which is stated as follows : *Given two merge trees $T_1$ and $T_2$ and a value $\epsilon \geq 0$, compute an $\epsilon$-compatible map between them if such a map exists; otherwise report that no such map exists.*

Given two merge trees $T_1$ and $T_2$, a *$c$-approximate decision procedure* for any $c \geq 1$ does the following: if $d_I(T_1, T_2) \leq \epsilon$, it returns a pair of $c\epsilon$-compatible maps between $T_1$ and $T_2$; if $d_I(T_1, T_2) > \epsilon$ it will either return a pair of $c\epsilon$-compatible maps between $T_1$ and $T_2$ or report that no such maps exist. Using binary search, this gives us a $c$-approximation to $d_I(T_1, T_2)$.

If we know $\alpha^\epsilon(x)$ for a point $x$ at height $h$, then we can compute $\alpha^\epsilon(y)$ for any ancestor $y$ of $x$ at height $h' \geq h$ by simply putting $\alpha^\epsilon(y) = j^{h'-h} \circ \alpha^\epsilon(x)$. A similar claim holds for $\beta^\epsilon$. 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.

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 end points. Given a parameter $\epsilon > 0$, an edge is called *$\epsilon$-long*, or *long* for brevity, if its length is greater than $2\epsilon$. We first describe an exact decision procedure if all edges in both trees are long, and then describe an approximate decision procedure when the two trees have short edges. Finally, we combine the two procedures to handle arbitrary merge trees.

**Algorithm for trees with long edges.** A *subtree* rooted at a point $x$ in a merge tree $T$, denoted $T^x$, includes all the points in the merge tree that are descendants of $x$ and an edge from $x$ that extends upwards to height $\infty$. For every $x \in T$, the nearest descendant of $x$ (including $x$) that is in $V(T)$, say $\tau(x)$, is the only node such that $T^x = T^{\tau(x)}$. For a node $u \in V(T)$, let $C(u)$ denote the children of $u$.

Assume $d_I(T_1, T_2) \leq \epsilon$, and let $\alpha : T_1 \to T_2$ and $\beta : T_2 \to T_1$ be a pair of $\epsilon$-compatible maps. We define an indicator function $\Phi : T_1 \times T_2 \to \{0, 1\}$ such that $\Phi(u, v) = 1$ if $d_I(T_1^u, T_2^v) \leq \epsilon$ and 0 otherwise. We propose an algorithm to compute $\Phi(u, v)$ for all $u \in V(T_1), v \in V(T_2)$. If $\Phi(u, v) = 1$, the algorithm also computes a pair of $\epsilon$-compatible maps between $T_1^u$ and $T_2^v$. We are interested in $\Phi(r_1, r_2)$, where $r_1$ (resp. $r_2$) is the root of $T_1$ (resp. $T_2$).

**Lemma 5.1.** *If all the edges are long, the maps $\alpha$ and $\beta$ induce a bijection between the subtrees rooted at the nodes of $T_1$ and the nodes of $T_2$.*

*Proof.* We define $\Psi_1 : V(T_1) \to V(T_2)$ as follows. Let $u \in V(T_1)$, and let $u_p$ be its parent (for $u = r_1$ we set $u_p$ to be an artificial node at height $\infty$ above $r_1$). Let $u'$ be the ancestor of $u$ at height $f(u_p) - 2\epsilon - \epsilon_0$ where $\epsilon_0$ is such that all the children of $u_p$ have height less than $f(u')$ and $\alpha(u') \notin V(T_2)$. We may use the same $\epsilon_0$ for all $u \in V(T_1)$. Set $\Psi_1(u) = \tau(\alpha(u'))$. We prove that $|f(u) - g(\Psi_1(u))| \leq \epsilon$. This is

true because all the points in $T_1^u$ map to points in $T_2^{\Psi_1(u)}$ and vice versa, hence $d_I\left(T_1^u, T_2^{\Psi_1(u)}\right) \leq \epsilon$. If $|f(u) - g(\Psi_1(u))| > \epsilon$, the roots of $T_1^u$ and $T_2^{\Psi_1(u)}$ are more than $\epsilon$ apart and at least one edge $e$ incident to one of the roots will not be in the image of the corresponding $\epsilon$-compatible map. However, the composition map applied to the lower node incident to $e$ must map it to a point inside $e$ (since the edges are longer than $2\epsilon$), a contradiction. Define $\Psi_2$ similarly.

We now prove $\Psi_2(\Psi_1(u)) = u$ for all $u \in V(T_1)$. We know $\beta(\alpha(u'))$ lies on the edge $(u_p, u)$, because $f(u') < f(u_p) - 2\epsilon$. Therefore, $\beta(\Psi_1(u))$ is a descendant of $u_p$. Because $g(\Psi_1(u)) \geq f(u) - \epsilon$, we further conclude $\beta(\Psi_1(u))$ is an ancestor of $u$ and $\Psi_2(\Psi_1(u))$ is an ancestor of $u$ as well. Since $|f(u) - g(\Psi_1(u))| \leq \epsilon$ and $|g(v) - f(\Psi_2(v))| \leq \epsilon$ for all $u \in V(T_1)$ and $v \in V(T_2)$, we have $|f(u) - f(\Psi_2(\Psi_1(u))| \leq 2\epsilon$. All the edges are longer than $2\epsilon$, so $\Psi_2(\Psi_1(u)) = u$. We conclude $\Psi_1$ is a surjection, with $\Psi_2$ as its inverse. By symmetry, $\Psi_2$ must be surjective as well, making $\Psi_1$ a bijection. □

**Lemma 5.2.** *Suppose all the edges in $T_1$ and $T_2$ are long. For any pair of nodes $u \in V(T_1), v \in V(T_2)$, $\Phi(u, v) = 1$ iff all of the following hold : (i) $|f(u) - g(v)| \leq \epsilon$; (ii) $|C(u)| = |C(v)|$; (iii) Let $C(u) = \{u_1, \ldots u_k\}$ and $C(v) = \{v_1, \ldots, v_k\}$, then there exists a permutation $\pi$ of $[1 : k]$ such that $\Phi(u_i, v_{\pi(i)}) = 1$ for all $i \in [1 : k]$.*

*Proof.* Suppose $\Phi(u, v) = 1$. If property (i) does not hold, then in any pair of $\epsilon$-compatible maps $(\alpha, \beta)$ between $T_1^u$ and $T_2^v$ there exists a point immediately below either $u$ or $v$ that is not in the image of $\beta$ or $\alpha$ resp., a contradiction since such a point has a descendant at height more than $2\epsilon$ below and hence must lie in the image of $\alpha$ or $\beta$. By Lemma 5.1, there exists a bijection, $\Psi_1$, between the nodes of $T_1^u$ and $T_2^v$ that respects the ancestor-descendant relationship between nodes. Thus, $|C(u)| = |C(v)|$ and property (ii) holds. Also, if $\Psi_1(u') = v'$, then $\Phi(u', v') = 1$ and this proves property (iii).

Suppose properties (i),(ii) and (iii) hold. Let $(\alpha_i, \beta_i)$ be the pair of $\epsilon$-compatible maps between $T_1^{u_i}$ and $T_2^{v_{\pi(i)}}$. Then, a pair of $\epsilon$-compatible maps $(\alpha, \beta)$ between $T_1^u$ and $T_2^v$ is obtained as follows : $\alpha(x) = \{\alpha^i(x) \mid x \in T_1^{u_i}\}$ ($\beta$ is defined similarly). Note that points on the infinite edge from $u$ (resp. $v$) upwards are shared among all $T_1^{u_i}$ (resp. $T_2^{v_j}$), whereas all other points in $T_1^u$ (resp. $T_2^v$) are present in only one $T_1^{u_i}$ (resp. $T_2^{v_j}$). However, since $|f(u) - g(v)| \leq \epsilon$, *shared* points are mapped to *shared* points and we have $|\alpha(x)| = 1$ (resp. $|\beta(y)| = 1|$) for all $x \in T_1^u$ (resp. $y \in T_2^v$). Thus, $\alpha$ and $\beta$ are functions and satisfy all the required properties. □

Using Lemma 5.2, we compute $\Phi(u, v)$ in a bottom-up manner. 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 5.2 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]. If $G_{uv}$ has a perfect matching $M = \{(u_1, v_{\pi(1)}), \ldots, (u_k, v_{\pi(k)})\}$, we set $\Phi(u, v) = 1$, else we set $\Phi(u, v) = 0$. If $\Phi(u, v) = 1$, we use the $\epsilon$-compatible maps for $T_{u_i}, T_{v_{\pi(i)}}$, for $1 \leq i \leq k$, to compute a pair of $\epsilon$-compatible maps between $T_1^u$ and $T_2^v$, as discussed in the proof of Lemma 5.2.

**Theorem 5.3.** *Given two merge trees $T_1$ and $T_2$ and a parameter $\epsilon > 0$ such that all edges of $T_1$ and $T_2$ are $\epsilon$-long, then whether $d_I(T_1, T_2) \leq \epsilon$ can be determined in $O(n^{5/2})$ time. If the answer is yes, a pair of $\epsilon$-compatible maps between $T_1$ and $T_2$ can be computed within the same time.*

*Proof.* The only thing left to show is the running time. Let $k_u > 0$ be the number of children of non-leaf node $u$ in either tree $T_1$ or $T_2$. The total time taken running Hopcroft and Karp [12] can be upper bounded by a constant factor of the following expression:

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

□

**Algorithm for 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 3). 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.

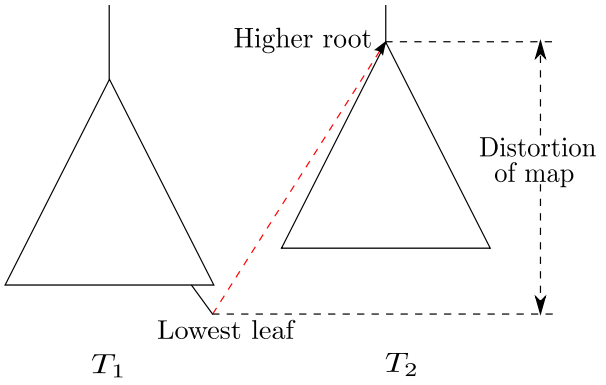


Figure 3: A naive map.

Here is an outline of the algorithm. After carefully *trimming* off short subtrees from the input trees, the algorithm decomposes them 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.

More precisely, given $T_1$, $T_2$ and $\epsilon > 0$, define the *extent* $e(x)$ of a *point* $x$ (which is not necessarily a tree node) in $T_1$ or $T_2$ as the maximum height difference between $x$ and any of its descendants. Suppose each edge is at most $s\epsilon$ long. Let $T_1'$ and $T_2'$ be subsets of $T_1$ and $T_2$ consisting only of points with extent at least $2\sqrt{ns}\epsilon$, adding nodes to the new leaves of $T_1'$ and $T_2'$ as necessary. Note that $T_1'$ and $T_2'$ themselves are trees. For example, in Figure 4 the red points in the left tree are those with extent less than a fixed value, and the right tree is obtained after trimming the red points.

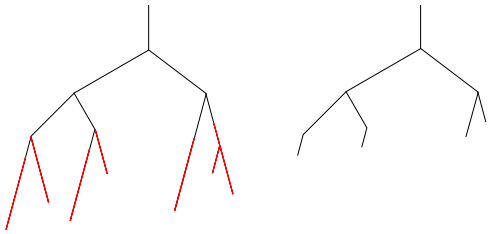

Figure 4: Tree after trimming red points.

**Lemma 5.4.** *If $d_I(T_1, T_2) \leq \epsilon$, then $d_I(T_1', T_2') \leq \epsilon$.*

*Proof.* Let $\alpha : T_1 \to T_2$ and $\beta : T_2 \to T_1$ be functions showing that the interleaving distance between $T_1$ and $T_2$ is at most $\epsilon$. Here, we assume that for any $x \in T_1$ we have $g(\alpha(x)) = f(x) + \epsilon$. Let $\alpha'$ and $\beta'$ be restrictions of the functions' domains to $T_1'$ and $T_2'$ respectively. We argue that the ranges of $\alpha'$ and $\beta'$ lie in $T_2'$ and $T_1'$ respectively. Suppose otherwise. Then without loss of generality, there is a point $x \in T_1'$ with $y = \alpha(x)$ not in $T_2'$. Because $x \in T_1'$, its extent in $T_1$ is at least $2\sqrt{ns}\epsilon$. Therefore, there exists a descendant $x'$ of $x$ in $T_1$ with $f(x') = f(x) - 2\sqrt{ns}\epsilon$. Because $y$ is not in $T_2'$, the extent of $y$ must be less than $2\sqrt{ns}\epsilon$ and there exists no descendant $y'$ of $y$ with $g(y') = g(y) - 2\sqrt{ns}\epsilon = f(x) - 2\sqrt{ns}\epsilon + \epsilon = f(x') + \epsilon$. We thus have a contradiction on $\alpha(x')$ going to a descendant of $\alpha(x)$. □

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 $T_1'$ and $T_2'$. A point $x$ in $T_1'$ is a matching point if there exists a branching node $x'$ in $T_1'$ or $y'$ in $T_2'$ with function value $f(x)$ and there exist no branching nodes nor leaves in $T_1'$ or $T_2'$ with function value in the range $(f(x), f(x) + 2\epsilon]$. Matching points on $T_2'$ are defined similarly. By this definition, no two matching points share a function value within $2\epsilon$ 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 $T_1'$ and $T_2'$ are matching points. There are at most $O(n^2)$ matching points.

Suppose $d_I(T_1', T_2') \leq \epsilon$, and let $\alpha' : T_1' \to T_2'$ and $\beta' : T_2' \to T_1'$ be a pair of $\epsilon$-compatible functions for $T_1'$ and $T_2'$. Call a matching point $x$ in $T_1'$ and a matching point $y$ in $T_2'$ with $f(x) = g(y)$ *matched* if $\alpha'(x)$ is an ancestor of $y$.

**Lemma 5.5.** *Let $x$ be any matching point in $T_1'$. The matched relation is a bijective function between matching points in $T_1'$ with function value $f(x)$ and matching points in $T_2'$ with function value $f(x)$.*

*Proof.* No two distinct matching points $y_1$ and $y_2$ on $T_2'$ with $f(x) = g(y_1) = g(y_2)$ share the same ancestor with function value $f(x)+\epsilon$, because they have no branching ancestors with low enough function value. Therefore, a matching point in $T_1'$ can be matched to at most one matching point in $T_2'$.

Let $x_1$ and $x_2$ be two distinct matching points from $T_1'$ 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 $f(x) + 2\epsilon$. However, $x_1$ and $x_2$ have no branching ancestor with low enough function value for $x'$ to exist. Hence, the matching relation must be injective.

Finally, consider any matching point $y$ on $T_2'$ with $g(y) = f(x)$. Point $x_1' = \beta'(y)$ is the ancestor of a matching point $x_1$ on $T_1'$. (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 $g(y') \leq g(y) + 2\epsilon$. 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. □

Let $T_1^m$ be a rooted tree consisting of one node per matching point on $T_1'$. Let $p(v)$ be the matching point for node $v$. Tree $T_1^m$ has node $v$ as an ancestor of node $u$ if $p(v)$ is an ancestor of $p(u)$ (see Figure 5). Define $T_2^m$ similarly. The size of $T_1^m$ and $T_2^m$ is $O(n^2)$.

Intuitively, $T_1^m$ and $T_2^m$ represent the trees induced by matching points. By the definition of interleaving distance and Lemma 5.5, $T_1^m$ and $T_2^m$ are isomorphic if $T_1'$ and $T_2'$ have interleaving distance at most $\epsilon$.

Our algorithm finds an isomorphism between $T_1^m$ and $T_2^m$ in linear time [1]. If one does not exist, then the interleaving distance between $T_1'$ and $T_2'$ must be greater than $\epsilon$; by Lemma 5.4, it thus reports that $T_1$ and $T_2$ have interleaving distance greater than $\epsilon$.

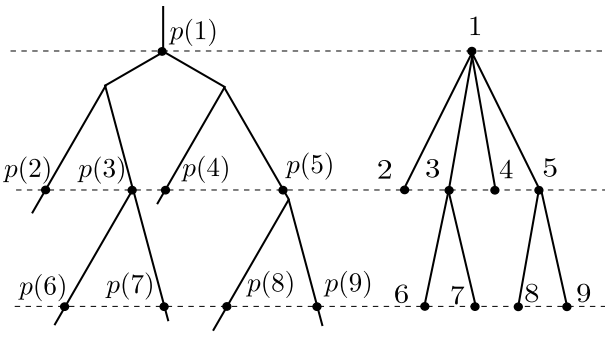


Figure 5: The left tree shows matching points on tree $T_1'$ and the right tree shows $T_1^m$.

If an isomorphism between $T_1^m$ and $T_2^m$ does exist, then the following functions $\alpha : T_1 \to T_2$ and $\beta : T_2 \to T_1$ are returned. For each matched pair of matching points $x$ and $y$, the algorithm sets $\alpha(x) = y$ and $\beta(y) = x$. Now, let $(f_1, f_2)$ be any maximal range of function values without any branching points in $T_1'$ or $T_2'$ where $f_2 - f_1 > 2\epsilon$. Let $x'$ be any point in $T_1'$ with $f(x') \in (f_1, f_2)$. Point $x'$ has a unique matching point descendant $x$. The algorithm sets $\alpha(x')$ to the point $y'$ in $T_2'$ 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 $T_1'$, the algorithm sets $\alpha(x'')$ to $\alpha(x)$ where $x$ is the lowest matching point *ancestor* of $x''$. Assignment $\beta(y'')$ is defined similarly for remaining points $y''$ in $T_2'$. We call such points $x''$ and $y''$ *lazily assigned*. Finally, each point $x'''$ in $T_1 - T_1'$ has $\alpha(x''')$ set to $\alpha(x)$ where $x$ is the lowest ancestor of $x'''$ on $T_1'$. Similar assignments are done for points in $T_2 - T_2'$.

One can verify that $\alpha$ and $\beta$ meet all their desired properties except for how much a point's function value can change going from one tree to the other.

**Lemma 5.6.** *For each lazily assigned point $x''$ in $T_1'$, we have $g(\alpha(x'')) \leq f(x'') + 2\sqrt{ns}\epsilon$.*

*Proof.* Let $x$ be a matching point. We show that there exists a region $(f_1, f_2)$ as defined above with $f(x) - 2\sqrt{ns}\epsilon \leq f_2 \leq f(x)$. Consider sweeping over the function values downward starting at $f(x)$ and let $f_2$ be the largest function value possible for a region as defined above. If the sweep line ever goes a distance greater than $2\epsilon$ without encountering a branching node in $T_1'$ or $T_2'$, then an $f_2$ is found. Therefore, there will be at least one branching point $x'$ in $T_1'$ or $T_2'$ per change of $2\epsilon$ until $f_2$ is found. Fix an $f' \geq f_2$ with $f(x) - 2\sqrt{ns}\epsilon \leq f' \leq f(x)$. Because the extent of each point in $T_1'$ and $T_2'$ is at least $2\sqrt{ns}\epsilon$ in $T_1$ or $T_2$, each branching point $x'$ introduces at least one more component of $T_1$ or $T_2$ that exists at function value $f'$. Since each edge length is at most $s\epsilon$, we can uniquely charge at least $(f(x') - f')/s\epsilon$ nodes between $T_1$ and $T_2$ to each branching point $x'$ . The total number of nodes with function value between $f(x)$ and $f'$ is at least

$$\sum_{i=0}^{\frac{f(x)-f'}{2\epsilon}} \frac{f(x) - 2i\epsilon - f'}{s\epsilon} = \frac{2}{s} \sum_{i'=0}^{\frac{f(x)-f'}{2\epsilon}} i' > \frac{1}{s}\left(\frac{f(x) - f'}{2\epsilon}\right)^2 .$$

However, there are at most $n$ nodes total in $T_1$ and $T_2$. We see

$$\frac{1}{s}\left(\frac{f(x)-f'}{2\epsilon}\right)^2 < n$$
$$\Rightarrow f(x) - f' < 2\sqrt{ns}\cdot\epsilon \ .$$

We see $f'$ cannot be too far below $f(x)$. Therefore, $f_2 \geq f(x) - 2\sqrt{ns}\epsilon$. □

**Theorem 5.7.** *Let $T_1$ and $T_2$ be two merge trees and $\epsilon > 0$ a parameter. There is an $O(n^2)$ time algorithm that returns a pair of $4\sqrt{ns}\epsilon$-compatible maps between $T_1$ and $T_2$, if $d_I(T_1, T_2) \leq \epsilon$ and the maximum length of a tree edge is $s\epsilon$. If $d_I(T_1, T_2) > \epsilon$, then the algorithm may return no or return a pair of $4\sqrt{ns}\epsilon$-compatible maps.*

*Proof.* By Lemma 5.6 and the symmetric lemma for $T_2'$, each point in $T_1'$ and $T_2'$ has its function value changed by at most $2\sqrt{ns}\epsilon$. Points outside $T_1'$ and $T_2'$ have their function value changed by at most $2 \cdot 2\sqrt{ns}\epsilon$. □

*Remark.* 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 5.6, that the height of a point and its image differ by at most $2n\epsilon$.

**Overall Algorithm.** Given trees $T_1$ and $T_2$, let $r$ denote the ratio between the lengths of the longest and the shortest edge in both trees. Our decision procedure works as follows. There are two cases –

**Case 1.** The shortest edge is longer than $2\epsilon$. We invoke the procedure for long edges and use Theorem 5.3.

**Case 2.** The shortest edge is at most $2\epsilon$. We invoke the procedure for short edges with $s = 2r$. Using Theorem 5.7 and the remark following it, we get a $\min(2n, 4\sqrt{2rn})$-approximate decision procedure.

Finally, by plugging this decision into a binary search over all possible candidate values for $\epsilon$, we obtain an approximation algorithm for the interleaving distance. The following lemma states that the number of candidate values for $\epsilon$ is only $O(n^2)$.

**Lemma 5.8.** *Let $T_1$ and $T_2$ be two merge trees with internal nodes $I_1$ and $I_2$ resp. and leaves $L_1$ and $L_2$ resp. Then the value of $d_I(T_1, T_2)$ is either*
*(i) $|f(u) - g(v)|$ for some pair $(u, v) \in I_1 \times I_2 \cup L_1 \times L_2$, or*
*(ii) $\frac{1}{2}|f(u) - f(u')|$ for some $u \in L_1$, where $u'$ is an ancestor node of $u$, or*
*(iii) $\frac{1}{2}|f(v) - f(v')|$ for some $v \in L_2$, where $v'$ is an ancestor node of $v$.*

*Proof.* Given a pair of $\epsilon$-compatible maps $(\alpha, \beta)$ between $T_1$ and $T_2$, we will try to construct a pair of $(\epsilon - \epsilon_0)$-compatible maps $(\alpha', \beta')$, for an infinitesimally small $\epsilon_0$, by shifting $\alpha(x)$ (resp. $\beta(x)$) down by a height of $\epsilon_0$ for all $x \in T_1$ (resp. $y \in T_2$). The cases where it is impossible (or merely difficult) to do so will specify the necessary conditions for the optimality of $(\alpha, \beta)$. In the remainder of the proof, we will deal with a single point $x \in T_1$ and try to come up with the maps $\alpha'(x)$ and $\beta'(\alpha'(x))$ (the proof is symmetric for $y \in T_2$).

First, if any two of $x$, $\alpha(x)$, or $\beta(\alpha(x))$ are branching nodes or leaves, then one of the necessary conditions for the lemma holds. We assume otherwise for the remainder of the proof.

If none of $x$, $\alpha(x)$, or $\beta(\alpha(x))$ is a node, or if only $x$ is a node, we set $\alpha'(x) = j^{-\epsilon_0}(\alpha(x))$ and $\beta'(\alpha'(x)) = i^{-2\epsilon_0}(\beta(\alpha(x)))$ (note that $j^{-\epsilon_0}$ and $i^{-2\epsilon_0}$ are well defined since the points that they operate upon are internal points and have a unique descendant at height $2\epsilon_0$ below).

If $\alpha(x)$ is a branching node, we set $\alpha'(x) = \alpha(j^{-\epsilon_0}(x))$ and $\beta'(\alpha'(x)) = i^{-2\epsilon_0}(\beta(\alpha(x)))$. If $\alpha(x)$ is a leaf, then we have a contradiction, because the descendants of $x$ cannot map to descendants of $\alpha(x)$.

Finally, if $\beta(\alpha(x))$ is a branching node, then we set $\alpha'(x) = \alpha(j^{-\epsilon_0}(x))$ and $\beta'(\alpha'(x)) = i^{2(\epsilon-\epsilon_0))}(x)$. □

Thus, binary search takes $O(\log n)$ time. Hence, we have the following.

**Theorem 5.9.** *Given two merge trees $T_1$ and $T_2$ with a total of $n$ vertices, there exists an $O(n^{5/2}\log n)$ time $O\left(\min\{n, \sqrt{rn}\}\right)$-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 5.9 with Corollary 4.2, we have:

**Corollary 5.10.** *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 $O\left(\min\{n, \sqrt{rn}\}\right)$-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 *30th Annual Sympos. on Comput. Geom.*, page 464, 2014.

[3] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Efficient computation of isometry-invariant distances between surfaces. *SIAM J. on Sci. Comput.*, 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. of Mach. Learn. Res.*, 11:1425–1470, 2010.

[6] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Disc. Comput. Geom.*, 37(1):103–120, 2007.

[7] V. de Silva, E. Munch, and A. Patel. Categorification of reeb graphs. Preprint, 2014.

[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*, volume 3624 of *Lecture Notes in Computer Science*, pages 111–122. Springer Berlin Heidelberg, 2005.

[12] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. on Comp.*, 2(4):225–231, 1973.

[13] C. Kenyon, Y. Rabani, and A. Sinclair. Low distortion maps between point sets. *SIAM J. on Comp.*, 39(4):1617–1636, 2009.

[14] F. Memoli. On the use of Gromov-Hausdorff distances for shape comparison. In *Eurographics Symposium on Point-Based Graphics*, 2007.

[15] F. Mémoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Found. of Comput. Math.*, 5(3):313–347, 2005.

[16] D. Morozov, K. Beketayev, and G. H. Weber. Interleaving distance between merge trees. In *Workshop on Topological Methods in Data Analysis and Visualization: Theory, Algorithms and Applications*, 2013.

[17] C. Papadimitriou and S. Safra. The complexity of low-distortion embeddings between point sets. In *16th Annual ACM-SIAM Symp. on Discrete Algo.*, pages 112–118, 2005.