The Gromov-Hausdorff distance is one of the most natural distance measures between metric spaces, and has been used, for example, for matching deformable shapes [4, 16], and for analyzing hierarchical clustering trees [6]. Informally, the Gromov-Hausdorff distance measures the additive distortion suffered when mapping one metric space to another using a correspondence between their points. Even when working with spaces of equal size, correspondences such as those computed for GH distance are often preferable to, say, bijections, because the distance is less sensitive to slight stretching or the exact choice of points sampled from a larger space (see Figure 1). Multiple approaches have been proposed to estimate the Gromov-Hausdorff distance [4, 15, 16].

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 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 [2].
**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. [14] 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 [18] 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 [12] discuss the additive distortion problem – given two equal-sized point sets $S, T \subset \mathbb{R}^d$, find the smallest $\Delta$ such that there exists a bijection $f : S \to T$ 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. In particular, an optimal correspondence between arbitrary spaces of equal size may not be a bijection (see Figure 1), so these hardness results do not apply to GH distance. Thus, their approach cannot be easily extended to our setting.

![Fig. 1. Solid and hollow points sampled from distinct metric spaces; each metric space contains two distinct clusters with similar intra- and inter-cluster distances, but the points are sampled differently for each. Using correspondences between the two sets will lead to a small distortion. However, using bijections will map points from the same cluster to different clusters in the other metric space, leading to a much larger distortion.](image)

The interleaving distance between merge trees [17] 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 [3, 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 efficient 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 $O(\min\{n, \sqrt{n}\})$ 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, and we hope this first non-trivial approximation result will stimulate further research on this and similar problems.

Our algorithm uses a reduction, described in Section 4, to the similar problem of computing the interleaving distance [17] between two merge trees. Given a function $f : X \to \mathbb{R}$ over a topological space $X$, 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. [17] proposed the interleaving distance as a way to compare merge trees and their associated functions. For us, interleaving distance provides a convenient way to measure distance after rooting the input trees so that correspondences are more stable to small perturbations in a function. Distances for the more general Reeb graphs are given in [3, 7].

---

3In fact, our hardness result can be easily extended to the GH distance between graphic metrics for trees and the interleaving distance between merge trees.
nicely structured. We describe, in Section 5, an $O(\min(n, \sqrt{n}^3))$-approximation algorithm for interleaving distance between merge trees, and our reduction provides a similar approximation for computing the GH distance between two metric trees.

Note on independent work. After writing preliminary versions of the current report, we became aware of recent work by Schmiedl [19] in which he proves a lower bound of 3 for the approximation ratio of any polynomial time approximation for GH distance in metric trees. As in our proof, his ultimately comes down to a reduction from 3-PARTITION. He also gives an approximation algorithm for arbitrary metrics, but the running time is not guaranteed to be polynomial in the size of the input.

2 PRELIMINARIES

Metric Spaces and the Gromov-Hausdorff Distance. A metric space $X = (X, \rho)$ 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 $C \subseteq A \times B$ such that: (i) for all $a \in A$, there exists $b \in B$ such that $(a, b) \in C$; and (ii) for all $b \in B$, there exists $a \in A$ such that $(a, b) \in C$. We use $\Pi(A, B)$ to denote the set of all correspondences between $A$ and $B$.

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

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

The Gromov-Hausdorff distance [15], $d_{GH}$, between $X_1$ and $X_2$ is defined as:

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

Intuitively, $d_{GH}$ measures how close can we can 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 [5] and Remark 1 of [15].

A tree $T = (V, E)$ consists of a set of nodes $V$ and edges $E$ connecting pairs of nodes such that some geometric realization of $T$ is simply connected. Note our definition allows for nodes of degree 2. The leaves of $T$ are the nodes of degree 1, and its branching nodes are the nodes of degree 3 or higher.

Given a tree $T = (V, E)$ and a length function $l : E \to \mathbb{R}_{\geq 0}$, we associate a metric space $T = (T, d)$ with $T$ as follows. $T$ is a geometric realization of $T$. For $x, y \in T$, define $d(x, y)$ to be the length of the unique shortest path $\pi(x, y) \in T$. It is clear that $d$ is a metric. The metric tree thus obtained is a metric tree. We often do not distinguish between $T$ and $T$ and write $T = (T, d)$.

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

The sublevel sets $f_a$ of $f$ merge as function value $a$ increases, so we get a rooted tree where branching nodes represent the moments where two components merge and leaves represent the birth of a new component at a local minimum. Figure 2 shows an example of a merge tree for a 1-dimensional function. Note that the merge tree extends to a height of $\infty$. Our assumption that $X$ is connected implies we have only one component in $f_{<N}$ for some sufficiently large $N > 0$, and $M_f$ is, in fact, a single tree.
We define the root of merge tree $M_f$ to be the node with the highest function value. The ancestors of a point $x \in M_f$ are the points lying on the unique path from $x$ to the root of $M_f$ as well as all points with greater height than the root. 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. Ancestors of points in $M_g$ are defined similarly.

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 $\varepsilon$-shift map $\sigma^f_\varepsilon : M_f \to M_f$ 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 $\alpha : M_f \to M_g$ and $\beta : M_g \to M_f$ are said to be $\varepsilon$-compatible if they satisfy the following conditions:

$$\hat{g}(\alpha(x)) = \hat{f}(x) + \varepsilon, \forall x \in M_f; \quad \hat{f}(\beta(y)) = \hat{g}(y) + \varepsilon, \forall y \in M_g; \quad \beta \circ \alpha = \sigma^2_\varepsilon; \quad \alpha \circ \beta = \sigma^2_\varepsilon. \quad (1)$$

See Figure 3 for an example. The interleaving distance [17] is then defined as

$$d_I(M_f, M_g) = \inf \{ \varepsilon \geq 0 \mid \text{there exist } \varepsilon\text{-compatible maps } \alpha \text{ and } \beta \}. \quad (1)$$

Remark. We can relax the requirements on $\alpha$ and $\beta$ from their normal definitions as follows.

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

$$\hat{f}(x) \leq \hat{g}(\alpha(x)) \leq \hat{f}(x) + \varepsilon, \forall x \in M_f; \quad \hat{g}(y) \leq \hat{f}(\beta(y)) \leq \hat{g}(y) + \varepsilon, \forall y \in M_g. \quad (1)$$

(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 $\beta$.

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

Any pair of maps satisfying the original requirements also satisfies the relaxed requirements for the same value of $\varepsilon$. Conversely, for any pair of maps satisfying the relaxed requirements, we can stretch up the images for each map by instead mapping to the ancestors of the appropriate height of the original images, without changing the value of $\varepsilon$. Thus, both definitions of interleaving distance are equivalent. For convenience, when two $\varepsilon$-compatible maps are given to us we assume that they satisfy (1), but we construct $\varepsilon$-compatible maps that satisfy the relaxed conditions mentioned, knowing that they can be “stretched” as just described to satisfy (1).
If we know $a(x)$ for a point $x$ at height $h$, then we can compute $a(y)$ for any ancestor $y$ of $x$ at height $h' \geq h$ by simply putting $a(y) = c_y^{h' - h}a(x)$. A similar claim holds for $\beta$. 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.

Morozov et al. [17] prove several facts about interleaving distance that will prove useful in this work. In particular, interleaving distance is a metric [17, Lemma 1]. Further, it is a stable distance measure with respect to small function perturbations; specifically, given two functions $f : X \to \mathbb{R}$ and $g : X \to \mathbb{R}$, we have $d_I(M_f, M_g) \leq \|f - g\|_\infty$ [17, Theorem 2].

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 $X = \{a_1, \ldots, a_n\}$, and an integer $m$ such that $1 \leq m \leq n$, is it possible to partition $X$ into $m$ multisets $\{X_1, \ldots, X_m\}$ such that all 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 $X = \{a'_1, \ldots, a'_n\}$. 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 $X$ if and only if there exists an appropriate partition of $X'$.

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

Conversely, 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

$$d_{i'} \in X_i \quad a_k + n_1 \bar{a} = a_k + n_2 \bar{a} \Rightarrow (n_1 - n_2)\bar{a} = a_k - a_k \Rightarrow a_k \geq \bar{a},$$

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

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. Consider an instance $X = \{a_1, \ldots, a_n\}$ and $1 \leq m \leq n$ of BAL-PART. If $m = 1$, then a balanced partition into one
multiset trivially exists. If $m > 1$, we construct two trees $T_1$ and $T_2$ as follows. Let $\lambda > 8$ and $\rho < \lambda - 6$ be two positive constants. Let $T_{1,k}$ denote a star graph having $k$ edges, each of length $l$. $T_1$ has a node $r_1$ incident to an edge $(r_1, r_1')$ of length $\rho$ and to $n$ edges $\{ (r_1, p_1), \ldots, (r_1, p_n) \}$ of length 1. Each node $p_i$ is incident to another edge $(p_i, p_i')$ of length 1, and each $p_i'$ is the center of a copy of $T_{\lambda, \lambda}$. $T_2$ consists of a node $r_2$ incident to an edge $(r_2, r_2')$ of length $\rho$ 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 $T_{\lambda + 1, \lambda}$, and $a = \binom{n}{i} a_i / m$. See Figure 4 for an illustration. We refer to the edges of each $T_{\lambda, \lambda}$ in $T_1$ and the edges of copies of $T_{\lambda + 1, \lambda}$ in $T_2$ as bottom edges. Let $T_1$ and $T_2$ denote the metric trees associated with $T_1$ and $T_2$ respectively. Since $\lambda, \rho$ are constants and $a_i \leq n^c$ for all $1 \leq i \leq n$, this construction can be done in polynomial time.

**Lemma 3.2.** If $(X, m)$ is a yes instance of BAL-PART, then $d_{GH}(T_1, T_2) \leq 1$. Otherwise, $d_{GH}(T_1, T_2) \geq 3$.

**Proof.** Suppose $X$ can be partitioned into $m$ subsets $X_1, \ldots, X_m$ of equal weight $a = \binom{n}{2} a_i / m$. We construct a correspondence $C$ between $T_1$ and $T_2$ with distortion at most 2, implying that $d_{GH}(T_1, T_2) \leq 1$. 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 $C$. If $a_i \in X_1$, the linearly interpolated bijection between edges $(r_1, p_i)$ and $(r_2, q_i)$ is added to $C$, with $p_i$ mapping to $q_i$. Also, using linear interpolation, we map the path from $p_i$ to each leaf of $T_{\lambda, \lambda}$ to distinct edges going from $q_i$ to leaves of its copy of $T_{\lambda + 1, \lambda}$; this is possible since $T_{\lambda + 1, \lambda}$ has $\bar{a}$ leaves, and $a_{\in X_1, a = \bar{a}}$. Note that each edge $(p_i, p_i')$ may be mapped to a contiguous part of multiple edges in $T_2$. Overall, the distortion induced by $C$ is at most 2; this stems from the fact that $C$ is piecewise linear, and the difference between the length of any node-to-node path in one tree and its image under $C$ in the other tree is at most 2.

Suppose $d_{GH}(T_1, T_2) < 3$, and let $C$ be a correspondence between $T_1$ and $T_2$ with distortion $< 6$. Consider two leaves $l_1, l_2 \neq r_2'$ in $T_2$. Then $d(l_1, l_2) \geq 2\lambda + 2$. Let $l_1', l_2'$ be their corresponding images in $T_1$ under $C$. We argue that $l_1', l_2'$ lie on different bottom edges of $T_1$. Indeed, since $\text{Dist}(C) < 6$, the distance between $l_1'$ and $l_2'$ is $d(l_1', l_2') > d(l_1, l_2) - 6 \geq 2\lambda - 4$. Since $l_1', l_2'$ lie on the same edge of $T_1$, then $d(l_1', l_2') \leq \lambda < 2\lambda - 4$, so they have to lie on distinct edges of $T_1$. If either $l_1', l_2'$ lies on an edge $r_1p_i$, for some $1 \leq i \leq n$, then by construction, $d(l_1', l_2') \leq \lambda + 4 < 2\lambda - 4$ (recall that $\lambda > 8$). Finally, if either $l_1', l_2'$ lies on $(r_1, r_1')$ then by the choice of $\rho$, $d(l_1', l_2') \leq \rho + 2 < 2\lambda - 4$. Thus, both $l_1'$ and $l_2'$ lie on distinct bottom edges of $T_1$. In particular, the corresponding image $l_i'$ in $T_1$ of any one leaf $l_i$ in $T_2$ must lie on a bottom edge of $T_1$, because there exists some other leaf $l_j$ in $T_2$ by the assumption that $m > 1$. Hence, $C$ induces a bijection $\chi$ between the leaves of $T_1$ and $T_2$, where $\chi(l) = l'$ for $l \in T_2$ and $l' \in T_1$ is the leaf whose incident edge contains the image(s) of $l$ under $C$. Note that if $l_i, l_j \in T_2$ are incident to $q_i, q_j$ with $q_i \neq q_j$, then $\chi(l_i)$ and $\chi(l_j)$ are distinct if $p_j \neq p_i$; 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 $\chi$ can be used to partition $X$ into $m$ subsets $X_1, \ldots, X_m$ 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. □

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 $\text{GH}$ distance between two tree metric spaces $T_1$ and $T_2$, and the interleaving distance between two appropriately defined trees induced by the $T_i$'s are within constant factors of each other.

Given a metric tree $T = (V(T), d)$, 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 $T'$ from $T$. Intuitively, $T'$ 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 $T'$. 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 $T'$. 6
Let $\mathcal{T}_1 = (T_1, d_1)$ and $\mathcal{T}_2 = (T_2, d_2)$ be two metric trees. Define

$$\Delta = \min_{u \in V(\mathcal{T}_1), v \in V(\mathcal{T}_2)} d_1(T^u_1, T^v_2).$$

(3)

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

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

**Proof.** Suppose $\Delta = d_1(T^s_1, T^t_2)$ for some $s \in V(\mathcal{T}_1)$ and $t \in V(\mathcal{T}_2)$. Set $f := f_s$ and $g := f_t$. Let $\alpha : T^s_1 \rightarrow T^t_2 \beta : T^t_2 \rightarrow T^s_1$ be $\Delta$-compatible maps. We define the functions $\alpha^* : T^s_1 \rightarrow T^t_2$ and $\beta^* : T^t_2 \rightarrow T^s_1$ as follows:

$$\alpha^*(x) = \begin{cases} a(x) & \text{if } g(a(x)) \leq 0, \\ t & \text{otherwise.} \end{cases}$$

$$\beta^*(y) = \begin{cases} b(y) & \text{if } f(b(y)) \leq 0, \\ s & \text{otherwise.} \end{cases}$$

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

$$f(x) \leq g(\alpha^*(x)) \leq f(x) + \Delta,$$

$$g(y) \leq f(\beta^*(y)) \leq g(y) + \Delta.$$  

(4)

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

Consider the correspondence $\mathcal{C} \in T_1 \times T_2$ induced by $a^*$ and $\beta^*$ 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 least common ancestor of $x_1$ and $x_2$ in $T_1$ (if $T_1$ is rooted at $s$), and $w$ the least common ancestor of $y_1$ and $y_2$ in $T_2$ (if $T_2$ is rooted at $t$). Note that since $T_1$ and $T_2$ are trees, there is a unique path $x_1 \sim u \sim x_2$ between $x_1$ and $x_2$, such that $x_1 \sim u$ and $u \sim x_2$ 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 $y_1 \sim w \sim y_2$. Hence

$$d_1(x_1, x_2) = d_1(x_1, u) + d_1(u, x_2) = 2f(u) - f(x_1) - f(x_2),$$

$$d_2(y_1, y_2) = d_2(y_1, w) + d_2(w, y_2) = 2g(w) - g(y_1) - g(y_2).$$

We then have,

$$|d_1(x_1, x_2) - d_2(y_1, y_2)| = |2f(u) - f(x_1) - f(x_2) - 2g(w) + g(y_1) + g(y_2)|$$

$$\leq 2|f(u) - g(w)| + |f(x_1) - g(y_1)| + |f(x_2) - g(y_2)|$$

$$\leq 2|f(u) - g(w)| + 2\Delta \text{ (by (4))}.$$  

We now wish to bound $|f(u) - g(w)|$. Consider the case where $y_1 = \alpha^*(x_1), y_2 = \alpha^*(x_2)$. As we traverse the path $x_1 \sim u \sim x_2$, the image of this path under $\alpha^*$ is a path $y_1 \sim \alpha^*(u) \sim y_2$, where $y_1 \sim \alpha^*(u)$ and $\alpha^*(u) \sim y_2$ are each monotone in function $g$ values. Hence, $\alpha^*(u)$ is an ancestor of $y_1$ and $y_2$. Since $w$ is the least common ancestor of $y_1$ and $y_2$, we have that $\alpha^*(u)$ must be an ancestor of $w$. The same claim can be similarly proved for the remaining cases. Further, it can be similarly shown that $\beta^*(w)$ 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 lemma follows. 

Next, we prove an upper bound on $\Delta$.

**Lemma 4.2.** $\Delta \leq 14d_{GH}(\mathcal{T}_1, \mathcal{T}_2)$.
We prove below in Claim 4.3 that there is a vertex (in fact a leaf) \( z \) such that \( d(z, w) \geq \nu \) respectively. A result by Dey, Shi, and Wang [8, Corollary 6] (see also [9]) implies that \( T_1 \) and \( T_2 \) defined by the functions \( f_s \) and \( f_t \), respectively. A result by Dey, Shi, and Wang [8, Corollary 6] (see also [9]) implies that

\[
T \overset{\text{ROOF}}{=} 4.3 \end{equation}

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,

\[
d_1(T_1^2, T_2^2) \leq \|f_t - f_z\|_\infty \leq 8\delta.
\]

By the triangle inequality,

\[
d_1(T_1^2, T_2^2) \leq d_1(T_1^2, T_2^2) + d_1(T_1^2, T_2^2)
\]

\[
\leq 6\delta + 8\delta
\]

\[
\leq 14\delta.
\]

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 \( C^* \). 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 \( 8\delta \) 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 \( 8\delta \) 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 \( C^* \), that is, \( (\tilde{u}_1, u_1), (\tilde{u}_2, u_2) \in C^* \). Since \( \text{Dist}(C^*) \leq 2\delta \), 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 \( T_1 \), 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). Otherwise, if \( \tilde{u}_1 \) and \( \tilde{u}_2 \) do not have ancestor/descendant relation, let \( w \) be the nearest common ancestor of \( \tilde{u}_1 \) and \( \tilde{u}_2 \) in \( T_1 \) (see Figure 5). Let \( c_0 = d_1(s, w) \). For simplicity, set \( a = d_1(s, \tilde{u}_1) \) and
We continue the search using only higher values of \( \varepsilon \). Note, \( a \geq c_0, b \geq c_0 \).

(7)

Since \( s \) is an endpoint of the longest path in \( T_1 \), it follows that \( c_0 \geq \min\{a - c_0, b - c_0\} \). Indeed, if this were not the case, then without loss of generality, suppose the other point \( s' \) of the diameter pair is not in the subtree of \( T_1 \) rooted at \( \bar{u}_1 \). We then have

\[
\begin{align*}
d_1(\bar{u}_1, s') &= (a - c_0) + (b - c_0) > c_0 + (b - c_0) = d_1(s, s'),
\end{align*}
\]

a contradiction. By (6), \( a, b \geq 6\delta + v \). Thus

\[
c_0 \geq \min\{a - c_0, b - c_0\} \geq 6\delta + v - c_0 \Rightarrow c_0 \geq \frac{1}{3}(6\delta + v).
\]

(9)

Combining (7) and (9), we have

\[
d_1(\bar{u}_1, u_2) \leq a + b - 6\delta - v \leq 20\delta + 2v - 6\delta - v = 14\delta + v,
\]

(10)

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 within distance \( 8\delta \) of at least one 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 \( 8\delta + v \) each.

From Lemmas 4.1 and 4.2, we get the following.

**THEOREM 4.4.** Let \( \Delta = \min_{u \in V(T_1), v \in V(T_2)} d_1(T_1^u, T_2^v) \). Then

\[
\frac{1}{2}d_{GH}(T_1, T_2) \leq \Delta \leq 14d_{GH}(T_1, T_2).
\]

In order to approximate \( d_{GH}(T_1, T_2) \), we merely need to approximate the interleaving distance for each of the \( O(n^2) \) pairs of merge trees obtained by rooting \( T_1 \) and \( T_2 \) at each of their vertices.

**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 that runs the algorithm for interleaving distance \( O(n^2) \) times.

## 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 \( O(\min\{n, \sqrt{n}\}) \)-approximation algorithm for computing \( d_I(M_f, M_g) \).

**Candidate values and binary search.** We first show that a candidate set \( \Lambda \) of \( O(n^2) \) values can be computed in \( O(n^2) \) time such that \( d_I(M_f, M_g) \in \Lambda \). Given \( \Lambda \), we perform a binary search on \( \Lambda \). At each step, we use a \( c \)-approximate decision procedure, for \( c = c_1 \min\{n, \sqrt{n}\} \) 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 \( cc \)-compatible maps between \( M_f \) and \( M_g \); if \( d_I(M_f, M_g) > \varepsilon \), it will either return a pair of \( cc \)-compatible maps between \( M_f \) and \( M_g \) or report that no pair of \( c \)-compatible maps exist.

We perform the binary search using the \( c \)-approximate decision procedure in the following way: if the procedure returns a pair of \( cc \)-compatible maps for a value \( \varepsilon \in \Lambda \), we continue the search using only lesser values of \( \varepsilon \). Otherwise, we continue the search using only higher values of \( \varepsilon \). When there are no more candidate values to search, we return the maps for the minimum value of \( \varepsilon \) that yielded a pair of maps.

The above procedure returns a pair of \( cc \)-compatible maps where \( d_I(M_f, M_g) \leq cc \leq cd_I(M_f, M_g) \). Indeed, the procedure does not run out of candidate values \( \varepsilon < d_I(M_f, M_g) \) until it has tried some \( \varepsilon ' \leq d_I(M_f, M_g) \) for which the approximate decision procedure returned a pair of \( cc \)-compatible maps. We now describe the candidate set \( \Lambda \).
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

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

**Lemma 5.1.** $d_1(M_f, M_g) \in \Lambda$.

**Proof.** Suppose to the contrary that $d_1(M_f, M_g) = \varepsilon \notin \Lambda$. Let $\alpha : M_f \to M_g$ and $\beta : M_g \to M_f$ be $\varepsilon$-compatible maps that realize $d_1(M_f, M_g) = \varepsilon$. 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 $\alpha_{\downarrow}(x) = a(x)$ if $a(x)$ is a node of $M_g$, otherwise $\alpha_{\downarrow}(x)$ is the lower endpoint of the edge of $M_g$ containing $a(x)$. Similarly we define the function $\beta_{\downarrow} : M_g \to M_f$.

![Fig. 6. Trees $M_f$ and $M_g$. Here $\delta_v = \min(\frac{1}{2}\varepsilon_1, \varepsilon_2)$.](image)

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

$\delta_v = \min\{ \frac{1}{2}(f(\beta(a(v))) - f(\beta_{\downarrow}(a(v)))) - \varepsilon \}

$\delta_v \leq 0$ \text{ if } v \in V_f \text{ and } a(v) \notin \Lambda_g.

We now construct $(\varepsilon - \varepsilon_0)$-compatible maps $\hat{\alpha} : M_f \to M_g$ and $\hat{\beta} : M_g \to M_f$ (note $\varepsilon_0 \leq \varepsilon$, so $\varepsilon - \varepsilon_0$ is non-negative). We describe the construction of $\hat{\alpha}$; $\hat{\beta}$ is constructed similarly. By construction, for any node $u \in V_f$, $g(a(u)) - g(\alpha_{\downarrow}(u)) \geq \varepsilon_0$, so we set $\hat{\alpha}(u)$ to be the point on the edge of $M_g$ containing $a(u)$ such that $g(w) = f(u) + \varepsilon - \varepsilon_0$. Once we have defined $\hat{\alpha}(u)$ and $\hat{\beta}(v)$ 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

$\hat{\alpha}(x) = \sigma_f^\varepsilon(\hat{\alpha}(u))$.

That is, we set $\hat{\alpha}(x)$ to be the ancestor of $\hat{\alpha}(u)$ at height $f(u) + \varepsilon - \varepsilon_0 + \gamma = f(x) + \varepsilon - \varepsilon_0$. 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 $\beta(a(x_1))$ for all $x_1 \in M_f$ (a similar result holds for $\hat{\alpha} \circ \hat{\beta}$ and $\alpha \circ \beta$).

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

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

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 $f(y) = f(x) + 2(\varepsilon - \varepsilon_0)$, $y$ must not be an ancestor of $x$. On the other hand, $\alpha, \beta$ are $\varepsilon$-compatible, so $y = \beta(a(x))$ 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
\[ y' = \beta(\alpha(x)) \]
\[ u = \beta_\gamma(\alpha(x)) \]
\[ y = \hat{\beta}(\hat{\alpha}(x)) \]

Fig. 7. Figure showing \( u, v, x, y \) and \( y' \).

\( y' \). (If \( y \) and \( y' \) lie on the same edge of \( M_\gamma \), then \( y \) is also an ancestor of \( x \).) Let \( u = \beta_\gamma(\alpha(x)) \). Let \( v \) be the lower endpoint of the edge \( e \) containing \( x \). See Figure 7. Since \( \epsilon \notin \Lambda \), \( f(u) \neq f(v) + 2\epsilon \) (i.e., \( u \neq \beta(\alpha(v)) \)). There are two cases to consider:

(i) \( f(u) > f(v) + 2\epsilon \). Then let \( u = \beta(\alpha(z)) \) for the point \( z \) lying between \( x \) and \( v \) at height \( f(z) = f(u) - 2\epsilon \).

Furthermore \( f(x) \geq f(z) > f(x) - 2\epsilon_0 \) (if \( f(x) - f(z) \geq 2\epsilon_0 \), then \( f(y') - f(u) = f(x) - f(z) \geq 2\epsilon_0 \), contradicting the fact that \( f(y') - f(y) = 2\epsilon_0 \)). Therefore we can choose a point \( w \neq v \) on \( e \) such that \( f(z) > f(w) > f(x) - 2\epsilon_0 \). We have \( \hat{\beta}(\hat{\alpha}(w)) \) 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., \( \beta(\alpha(w)) \neq \alpha_2^f(w) \), contradicting the fact that \( \alpha, \beta \) are \( \epsilon \)-compatible.

(ii) \( f(u) < f(v) + 2\epsilon \). In this case

\[ f(\beta(\alpha(v))) - f(\beta_\gamma(\alpha(v))) \leq f(\beta(\alpha(v))) - f(u) < f(y') - f(y) = 2\epsilon_0 \leq 2\epsilon_\nu, \]

which contradicts the definition of \( \delta_\nu \).

Hence we conclude that \( y \) is an ancestor of \( x \), i.e., \( \hat{\beta} \circ \hat{\alpha} = \alpha_2^f(\epsilon - \epsilon_0) \). Similarly, one can argue that \( \hat{\alpha} \circ \hat{\beta} = \alpha_2^f(\epsilon - \epsilon_0) \), implying that \( \hat{\alpha}, \hat{\beta} \) are \( (\epsilon - \epsilon_0) \)-compatible maps as claimed.

Putting everything together, we conclude that \( \epsilon \in \Lambda \).

We now describe the decision procedure to answer the question “is \( d_I(M_f, M_\gamma) \leq \epsilon ? \)” approximately. Given a parameter \( \epsilon > 0 \), an edge is called \( \epsilon \)-long, or long for brevity, if its length is strictly greater than \( 2\epsilon \). 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 \( \infty \). For a node \( u \in V \), let \( C(u) \) denote the children of \( u \) and let \( p(u) \) denote its parent. Assume \( d_I(M_f, M_\gamma) \leq \epsilon \), and let \( \alpha : M_f \rightarrow M_\gamma \) and \( \beta : M_\gamma \rightarrow M_f \) be a pair of \( \epsilon \)-compatible maps. As in the proof of Lemma 5.1, we define the functions \( \alpha_\gamma \) and \( \beta_\gamma \) but restricted only to the vertices of \( M_f \) and \( M_\gamma \). That is, for a node \( v \in V_f \), we define \( \alpha_\gamma(v) \) to be the lower endpoint of the edge containing \( \alpha(v) \) if \( \alpha(v) \) is a node, then \( \alpha_\gamma(v) \) is \( \alpha(v) \) itself. Similarly we define \( \beta_\gamma(w) \), for a node \( w \in V_\gamma \).

The following two properties of \( \alpha_\gamma \) and \( \beta_\gamma \) will be crucial for the decision procedure.

**Lemma 5.2.** If all edges in \( M_f \) and \( M_\gamma \) are \( \epsilon \)-long, then the following hold: (i) For a node \( v \in V_f \), \( |f(v) - g(\alpha_\gamma(v))| \leq \epsilon \), and (ii) for a node \( w \in V_\gamma \), \( |g(w) - f(\beta_\gamma(w))| \leq \epsilon \).

**Proof.** We will prove part (i); part (ii) is similar. By definition, \( g(\alpha_\gamma(v)) \leq f(v) + \epsilon \). Suppose \( g(\alpha_\gamma(v)) < f(v) - \epsilon \). Let \( v' \) be a point in \( M_\gamma \) lying on the edge containing \( \alpha(v) \) and \( \alpha_\gamma(v) \) with height \( f(v) - \epsilon - \epsilon_0 \), for some sufficiently small \( \epsilon_0 \). 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
The following lemma gives a recursive definition of \( \Phi \) maps between \( M \in \mu M \) that if \( u \in M, \alpha \) hold, and let \( v \) be a descendant of \( u \). Thus, \( \beta(\alpha(u)) \) is an ancestor of \( u \). Since all edges are longer than \( 2 \epsilon \), the claim follows. □

**Lemma 5.3.** If all edges in \( M_f \) and \( M_g \) are \( \epsilon \)-long, then \( \alpha_j \) and \( \beta_j \) are bijections with \( \alpha_j = \alpha_j^{-1} \) and \( \beta_j = \beta_j^{-1} \).

**Proof.** We will first show that \( \beta_j = \alpha_j^{-1} \). Suppose to the contrary there exists a vertex \( v \in V_f \) such that \( \beta_j(\alpha_j(v)) = w \neq v \). Let \( \alpha_j(v) = u \), for \( u \in V_g \). From Lemma 5.2 we have \( |f(v) - f(w)| \leq 2 \epsilon \). Since all edges are longer than \( 2 \epsilon \), \( v \neq w \), \( v \) cannot be an ancestor/descendant of \( w \) in \( M_f \). By definition of \( \alpha_j \), \( \alpha(v) \) is an ancestor of \( \alpha_j(v) = u \). Thus \( \beta(\alpha(v)) \) is an ancestor of \( \beta(u) \). Further, \( |f(v) - g(u)| \leq \epsilon \) (Lemma 5.2) and \( \beta(\alpha(v)) = \sigma^2_f(u) \) (since \( \alpha, \beta \) are \( \epsilon \)-compatible). Hence, \( \beta(u) \) lies between \( v \) and \( \beta(\alpha(v)) \) on the edge \( e \) whose lower endpoint is \( v \) as \( \epsilon \) is \( \epsilon \)-long. Thus, \( \beta(u) \) is an ancestor of \( v \). See Figure 8. Also by definition of \( \beta_j \), \( \beta(u) \) is an ancestor of \( \beta_j(u) = w \). Thus, \( w \) is a descendant of \( v \), a contradiction since \( v \) cannot be an ancestor of \( w \).

We thus have \( \beta_j = \alpha_j^{-1} \). Similarly, we can show that \( \alpha_j = \beta_j^{-1} \). This also implies that \( \alpha_j \) and \( \beta_j \) are bijections. □

\[ \begin{align*}
\Phi(u, v) &= \begin{cases} 
1, & \text{if } d_1(M^u_f, M^v_g) \leq \epsilon, \\
0, & \text{otherwise.}
\end{cases}
\end{align*} \]

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

For two integers, \( k \) and \( \ell \) with \( k \leq \ell \), let \( [k, \ell] = \{k, k+1, \ldots, \ell\} \).

**Lemma 5.4.** Suppose all the edges in \( M_f \) and \( M_g \) are \( \epsilon \)-long. Let \( (u, v) \in V_f \times V_g \), and let \( u_i \) and \( v_i \) denote the \( i \)-th and \( j \)-th child, respectively, of \( u \) and \( v \). We have \( \Phi(u, v) = 1 \) if and only if the following conditions hold: (i) \( |f(u) - g(v)| \leq \epsilon \), (ii) \( |C(u)| = |C(v)| \), and (iii) there exists a permutation \( \pi \) 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 \( \alpha, \beta \) be the corresponding \( \epsilon \)-compatible maps. Suppose property (i) does not hold, and let \( f(u) > g(v) + \epsilon \) 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 \( \beta \). However, \( \beta(\alpha(w)) = \sigma^2_f(w) \) must lie in the interior of \( e \) (since \( e \) is \( \epsilon \)-long), a contradiction. To prove that (ii) holds, note that by Lemma 5.3 and monotonicity of \( \alpha_j \) and \( \beta_j \), there exist bijections \( \alpha_j \) and \( \beta_j \) between the vertices of \( M^u_f \) and \( M^v_g \) such that if \( u_1 \in C(u_2) \) for a vertex \( u_2 \in M^u_f \), then \( \alpha_j(u_1) \in C(\alpha_j(u_2)) \) (a symmetric statement holds for \( \beta_j \) and vertices in \( M^v_g \)). Thus, \( \alpha_j, \beta_j \) induce bijections between \( C(u) \) and \( C(v) \), and hence \( |C(u)| = |C(v)| \). Finally, for (iii), let \( \alpha_j(u') = v' \) for some \( u' \in C(u), v' \in C(v) \). Then by definition of \( \alpha_j \) and \( \beta_j \), \( \alpha_j(M^u_f) \subseteq M^v_g \) and \( \beta_j(M^v_g) \subseteq M^u_f \). This means that the restriction of the pair of \( \epsilon \)-compatible maps \( \alpha \) and \( \beta \) to \( M^u_f \) and \( M^v_g \) respectively remain \( \epsilon \)-compatible for \( M^u_f \) and \( M^v_g \). Thus, \( \Phi(u', v') = 1 \), and the permutation \( \pi \) is defined by \( \alpha_j, \beta_j \).

We now prove the opposite direction. Suppose properties (i),(ii) and (iii) hold. Let \( \phi(u, v) \) be the pair of \( \epsilon \)-compatible maps between \( M^u_f \) and \( M^v_g \). Then, a pair of \( \epsilon \)-compatible maps \( (\alpha, \beta) \) between \( M^u_f \) and \( M^v_g \) is obtained as follows:
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 of the two roots of $M_f$ and $M_g$ (see Figure 9). 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.
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 $\epsilon$ 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 10 for an example.

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

**PROOF.** Let $\alpha : M_f \rightarrow M_g$ and $\beta : M_g \rightarrow M_f$ be $\epsilon$-compatible maps. Let $\alpha'$ and $\beta'$ be restrictions of the functions’ domains to $M'_f$ and $M'_g$ respectively. We argue that the ranges of $\alpha'$ and $\beta'$ 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 $\alpha, \beta$ are $\epsilon$-compatible maps.

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 increase the distance between them.

We now define *matching points* in $M'_f$ and $M'_g$. Let $H$ be the set of function values for leaves and branching nodes in $M'_f$ or $M'_g$, and let $H' \subseteq H$ be the subset of $H$ consisting of function values $h$ for which $(h, h + 2\epsilon)$ does not intersect $H$. A point $x$ in $M'_f$ is a matching point if $f(x) \in H'$. Similarly, a point $y$ in $M'_g$ is a matching point if $g(y) \in H'$. 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 $M'_f$ and $M'_g$ are matching points. There are $O(n^2)$ matching points.

Suppose $d_l(M'_f, M'_g) \leq \epsilon$, and let $\alpha' : M'_f \rightarrow M'_g$ and $\beta' : M'_g \rightarrow M'_f$ be a pair of $\epsilon$-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 matching point in $M'_g$.

Let $x_1$ and $x_2$ be two distinct matching points from $M'_f$ with $f(x_1) = f(x_2)$. 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 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 $M'_f$ with $f(x_1) \geq g(y)$. (Note that by the same argument as the beginning of this proof, only one such matching
point $x_1$ can exist. Point $y' = a'(x'_1)$ 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 $a'(x_1)$ must be an ancestor of $y$, meaning $x_1$ and $y$ are matched. Thus, the matching relation is surjective.

\[ \square \]

![Fig. 11. The left tree shows matching points on tree $M_f$ and the right tree shows $\hat{M}_f$.](image)

We now define a rooted tree $\hat{M}_f$ to be a rooted tree consisting of one node per matching point on $M_f$. Let $p(v)$ be the matching point for node $v$. $\hat{M}_f$ has node $v$ as an ancestor of node $u$ if $p(v)$ is an ancestor of $p(u)$ (see Figure 11). Define $\hat{M}_g$ similarly. The size of $\hat{M}_f$ and $\hat{M}_g$ is $O(n^2)$. Intuitively, $\hat{M}_f$ and $\hat{M}_g$ represent the trees induced by matching points. By the definition of interleaving distance and Lemma 5.7, $\hat{M}_f$ and $\hat{M}_g$ are isomorphic if $M'_f$ and $M'_g$ satisfy that $d_I(M'_f, M'_g) \leq \epsilon$.

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

Using the algorithm of [2, chap. 3, p. 85], we determine in time linear in the size of the trees whether $\hat{M}_f$ and $\hat{M}_g$ are isomorphic. If the answer is no, we return no. By Lemma 5.7, $d_I(M_f, M_g) > \epsilon$ in this case. Otherwise we construct the following functions $\alpha : M_f \to M_g$ and $\beta : M_g \to M_f$ and return them. Recall, it suffices to perform assignments where the function value increases by at most $ce$. 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 $(\xi_1, \xi_2)$ be any maximal range of function values without any branching nodes or leaves in $M'_f$ or $M'_g$ with $\xi_2 - \xi_1 > 2\epsilon$. 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 $\xi_1$, 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 $M'_g$ that were not paired with some $x'$. We call such points $x''$ and $y''$ lazily assigned. See Figure 12. 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 $M_g - M'_g$.

![Fig. 12. Figure showing points $x, y, x', y', x'', y''$.](image)
LEMMA 5.8. (i) For each lazily assigned point \( x'' \in M'_f \),
\[
g(a(x'')) \leq f(x'') + 2(\sqrt{2ns} + 1)\varepsilon.
\]
(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 much higher up.

Let \( x \) be a matching point. We show that there exists a region \((\xi_1, \xi_2)\) 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 \( \xi_2 \) be the largest function value possible for a region as defined above. If the sweep line ever goes a distance greater than \( 2\varepsilon \) without encountering a branching node or leaf in \( M'_f \) or \( M'_g \), then an \( \xi_2 \) is found. Therefore, there will be at least one branching node or leaf \( x' \) in \( M'_f \) or \( M'_g \) per descent of \( 2\varepsilon \) until \( \xi_2 \) is found. Suppose \( \xi_2 < f(x) - 2(\sqrt{2ns} + 1)\varepsilon \). Let \( l = \sqrt{2ns} + 1 \), and \( f' = f(x) - 2l\varepsilon \).

Let \( \{P_1, P_2, \ldots\} \) be a set of paths from each branching node of \( M'_f \) or \( M'_g \) encountered during the sweep to leaves of \( M'_f \) and \( M'_g \) such that for each pair of paths, both paths are disjoint except possibly at the higher endpoint of one of the twopaths. Such a set of paths can be found by greedily selecting an arbitrary path for each branching node as it is encountered. In addition, let \( \{Q_1, Q_2, \ldots\} \) be a set of pairwise-disjoint paths from leaves of \( M'_f \) and \( M'_g \) to leaves of \( M'_f \) and \( M'_g \). Because each point in \( M'_f \) and \( M'_g \) has extent at least \( 2\varepsilon \), the lower endpoint of each of these paths lies below height \( f' \). Since edge lengths are at most \( s\varepsilon \), there are at least \( (f(x') - f')/s\varepsilon \) nodes in \( M'_f \) or \( M'_g \) on the path \( P_i \) or \( Q_j \) from each branching node or leaf \( x' \) at height \( f(x') \in [f', f(x)) \), not counting \( x' \) itself. In total, these paths contain at least
\[
l \geq \frac{(l - i)2\varepsilon}{s\varepsilon} = \frac{l(l - 1)}{s}
\]

nodes, not counting their higher endpoints. Each node counted above appears on at most one path \( P_i \) and at most one path \( Q_j \), for at most two paths total, so
\[
l(l - 1) \leq 2n \Rightarrow l(l - 1) \leq 2ns,
\]
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 \( d_1(M_f, M_g) \leq \varepsilon \) and the maximum length of a tree edge is \( s\varepsilon \). If \( d_1(M_f, M_g) > \varepsilon \), then the algorithm may return no or return a pair of \( 4(\sqrt{2ns} + 1)\varepsilon \)-compatible maps.

PROOF. Constructing the trees \( M_f \) and \( M_g \), the corresponding isomorphism between them (if it exists), and the maps \( \alpha \) and \( \beta \) 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 \( \alpha \) and \( \beta \) 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 \( \alpha \) and \( \beta \) 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 \( 2ns \). In particular, the proof no longer requires as complicated a counting argument, because any path contains at most \( n \) nodes.
We have presented the first hardness results for computing the Gromov-Hausdorff distance between metric trees. We very difficult. On the algorithmic side of things in particular, trying for anything less than an $O(\sqrt{n})$-approximation appears to prevent our use of algorithms for graph isomorphism, the strongest algorithmic tool used in the above algorithm. We hope that our current investigation will stimulate more research on the theoretical and algorithmic aspects of embedding or matching under additive metric distortion.

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. While we would like to reduce this gap, doing so seems very difficult. On the algorithmic side of things in particular, trying for anything less than an $O(\sqrt{n})$-approximation appears to prevent our use of algorithms for graph isomorphism, the strongest algorithmic tool used in the above algorithm. 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