Ricci-Ollivier Curvature of the Rooted Phylogenetic Subtree-Prune-Regraft Graph*

Chris Whidden† Frederick A Matsen IV†

Abstract

Statistical phylogenetic inference methods use tree rearrangement operations such as subtree-prune-regraft (SPR) to perform Markov chain Monte Carlo (MCMC) across tree topologies. These methods are known to mix quickly when sampling from the simple uniform distribution of trees but may become stuck in the local optima of multi-modal posterior distributions for real data induced by non-uniform likelihoods. The structure of the graph induced by tree rearrangement operations is an important determinant of the mixing properties of MCMC, motivating study of the underlying *rSPR graph* in greater detail.

In this paper, we investigate the rSPR graph in a new way: by calculating Ricci-Ollivier curvature with respect to uniform and Metropolis-Hastings random walks. We confirm using simulation that mean access time distributions depend on distance, degree, and curvature, showing the relevance of these curvature results to stochastic tree search. These calculations require fast new algorithms for constructing and sampling these graphs, reducing the time required to compute an rSPR graph from $O(m^2n)$-time to $O(mn^3)$, where m is the (often large) number of trees in the graph and n their number of leaves, and reducing the time required to select an SPR neighbor of a tree uniformly at random to $O(n)$ time. We then develop a closed form solution to characterize how the number of SPR neighbors of a tree changes after an SPR operation is applied to that tree. This gives bounds on the curvature, as well as a flatness-in-the-limit theorem indicating that paths of small topology changes are easy to traverse. However, we find that large topology changes (i.e. moving a large subtree) gives pairs of trees with negative curvature. Although these pairs of trees with negative curvature do not impede mixing in this simple well-connected space, they may manifest as bottlenecks in the much smaller credible sets induced by phylogenetic posteriors with a likelihood function. This work extends our knowledge of the rSPR graph, in particular properties that are relevant for investigation of sampling the rSPR graph.

1 Introduction

Molecular phylogenetic methods reconstruct evolutionary trees from DNA or RNA data and are of fundamental importance to modern biology. Statistical phylogenetics is the currently most popular means of reconstructing phylogenetic trees, in which the tree is viewed as an unknown parameter in a likelihood-based statistical inference problem. The likelihood function in this setting is the likelihood of generating the observed sequences via a continuous time Markov chain (CTMC) evolving down the tree starting from a sequence assumed to be sampled from the stationary distribution [7]. The lengths of the branches of the phylogenetic tree give the "time" parameter in the CTMC, where the generated sequence accrues mutations, typically in an IID manner across sites. It is now common for researchers to approximate the posterior distribution of trees and their associated parameters in a Bayesian setting using Markov chain Monte Carlo (MCMC).

In order to estimate these distributions accurately, MCMC samplers must sufficiently explore the set of trees. Phylogenetic search algorithms typically attempt to do so through a combination of modifications to the continuous parameters and tree topology. Topology changes have been identified as the main limiting factor of Bayesian MCMC algorithms [13, 16], as other parameters cannot be accurately estimated if the topology distribution is not accurately sampled. Commonly used phylogenetics software packages such as MrBayes [29] and BEAST [4] rearrange subtrees via subtree-prune-regraft (SPR) moves (Figure 1(d)) or the subset of SPR moves called nearest neighbor interchanges (NNI) [27]. Thus, phylogenetic searches can be viewed as traversing the *SPR graph*: the graph with phylogenetic trees as vertices and SPR adjacencies as edges.

It has become increasingly clear that the structure of the SPR graph plays an important role in determining the accuracy of tree searches. Researchers have pre-

*This work was funded by National Science Foundation award 1223057. Chris Whidden is a Simons Foundation Fellow of the Life Sciences Research Foundation.

†Program in Computational Biology, Fred Hutchinson Cancer Research Center, Seattle, WA, USA 98109. {cwhidden,matsen}@fredhutch.org

viously identified slow mixing in MCMC with pathological data [22, 23, 28]. On the other hand, fast mixing has been identified with exceptionally well-behaved data [38] or with a uniform distribution [34]. Studies on real data [2, 16], however, have identified posteriors which are difficult to sample using MCMC. Previously, the lack of sufficient computational tools for examining phylogenetic posteriors in terms of SPR operations made it difficult to determine the cause of these difficulties. By developing the first such tools, we recently showed that graph structure has a significant effect on MCMC mixing with MrBayes applied to real data [44], and that multimodal posteriors are common and separated by "bottlenecks" of specific classes of SPR moves.

Although the SPR graph is thus very important in determining the success of phylogenetic inference procedures, still little is known about the rooted or unrooted versions of the SPR graph itself. [32] developed a recursive procedure on a tree to find the degree of the corresponding vertex in the rooted SPR (rSPR) graph, and corresponding bounds on degree. [5] showed that the diameter Δ_{rSPR} of the rSPR graph is $n - \Theta(\sqrt{n})$, and for the unrooted case they show

$$(1)\quad n-2\lceil\sqrt{n}\rceil+1 \leq \Delta_{\text{uSPR}}(n) \leq n-3-\left\lfloor \frac{\sqrt{n-2}-1}{2} \right\rfloor.$$

We are not aware of any further work investigating properties of the SPR graph, which may be due to its complexity. Indeed, even computing the distance between topologies in terms of SPR operations (rooted and unrooted) is NP-hard [3,12]. Fortunately, it is fixed-parameter tractable with respect to the distance in the rooted case [3] and efficient fixed-parameter algorithms have recently been developed [43,44] and begun to allow such investigation.

Ollivier and colleagues recently pioneered a new approach to calculating Ricci curvature on a general type of metric space, including graphs [15, 25]. In this framework, local information about the metric space is given by a random walk (rather than a Riemann tensor) such that their notion of curvature formalizes the notion of to what extent random walking brings points together. Applying the framework to Brownian motion on a manifold returns the classical definition of Ricci curvature. Curvature is determined by the ratio of the earth mover's distance [30] between neighborhoods of a pair of vertices given by a random walk and the distance between the vertices. Here the term *random walk* on a space X simply denotes a family of probability measures parameterized by points of X satisfying reasonable assumptions, which includes biased walks such as MCMC. This approach has been useful for determining properties of a wide variety of graphs including the internet topology [24] and cancer networks [31].

In this paper, we investigate curvature of the rSPR graph with respect to two random walks and compare those results to access times (i.e. hitting times) for those random walks. Our explicit focus here is to investigate random walks defined only in terms of the graph itself: the uniform random walk and MCMC sampling from the uniform prior on trees. In future work, we will extend these methods to study more complicated distributions with non-uniform topology probabilities.

We required several new computational tools. We present a fast new algorithm for computing rSPR graphs from a set of trees, reducing the time to do so from $\mathrm{O}(m^2n)$ to $\mathrm{O}(mn^3)$ for a set of m trees with n leaves. As the full rSPR graph on trees with n leaves contains $(2n-3)!! = 3 \cdot 5 \cdot \ldots \cdot (2n-3)$ trees, this is a significant improvement in practice for exploring large subsets of the graph (or, as we do here, the full graph for small numbers of leaves). By exploiting symmetries in the rSPR graph, we were able to calculate all of the curvatures for pairs of trees with up to seven leaves. By carefully examining the overlap in rSPR moves, we present a new method for computing the degree of a tree in the rSPR graph that allows one to select an rSPR neighbor uniformly at random in linear-time without explicitly generating the graph. This stands in contrast to the sampling methods used in current software such as MrBayes, which do not propose SPR moves uniformly.

Using our methods to simulate these random walks, we found that the distribution of access times between pairs of trees can be described by distance between the trees, the degrees of the trees, and the curvature. Moreover, we found that rSPR graphs for trees with 7 or more leaves have tree pairs with negative curvature, corresponding to direct paths that are difficult to traverse stochastically. By getting a more fine-tuned understanding of the rSPR neighborhood of pairs of vertices, we are able to give bounds on the earth mover's distance in this context and thus curvatures under these random walks. In particular, we present a full characterization of the change in rSPR degree that occurs from a given rSPR move and find that even though they each count as one move, rSPR moves which modify large subtrees are less likely to be explored during these random walks. Pairs of trees separated by such moves correspond to the pairs with negative curvature identified in our simulation results. These pairs occur infrequently in these well-connected graphs, however, they may be more problematic in real posterior distributions where the majority of probability is spread over a relatively

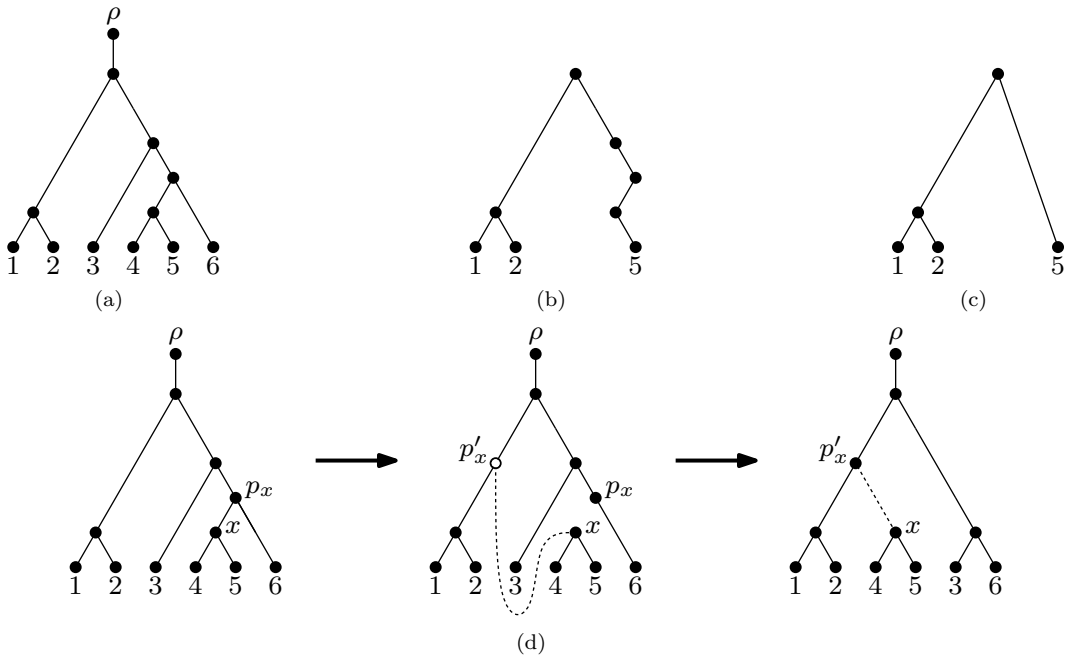


Figure 1: (a) An X-tree T. (b) $T(V)$, where $V = \{1, 2, 5\}$. (c) $T|V$. (d) An rSPR operation transforms T into a new tree T' by *pruning* a *subtree* and *regrafting* it in another location.

small number of trees [44]. In summary, we extend knowledge about an important graph for phylogenetics, specifically in a way that models phylogenetic MCMC search.

The automated computational analysis code can be found at `https://github.com/matsengrp/curvature`. Selected proofs of our theorems and lemmas can be found in the appendix. Proofs omitted for space can be found in [45].

2 Preliminaries

We follow the definitions and notation from [3, 43, 44]. A (rooted binary phylogenetic) X-tree is a rooted tree T whose nodes have zero or two children such that the leaves of T are bijectively labelled with the members of a label set X. As in [3,43,44], the tree is augmented with a labelled root node ρ and ρ is considered a member of X (Fig. 1(a)). We generally use n to refer to the number of leaves in an X-tree. For a subset V of X, $T(V)$ is the smallest subtree of T that connects all nodes in V (Fig. 1(b)). The V-tree induced by T is the smallest tree $T|V$ that can be obtained from $T(V)$ by suppressing unlabelled nodes with fewer than two children (Fig. 1(c)). For the rest of the paper, **we will assume that all phylogenetic trees are binary and rooted**, and that tree inclusion is rooted tree inclusion.

A *parent (sub)tree* of a subtree U is the smallest subtree strictly containing U. A *parent edge* of a subtree U is the edge connecting U to the rest of the tree. The *internal edges* of a tree are the edges that do not contact a leaf or ρ. A *ladder tree* (also known as a *caterpillar tree*) is a tree such that every internal node has a leaf as a direct descendant. A *balanced tree* is a tree such that the sum of the depths of internal nodes is minimum over all trees with the same number of leaves. The *least common ancestor* (LCA) of a set R of two or more nodes is the unique node that is an ancestor of each node $r \in R$ and at maximum depth. Similarly, the LCA of two or more subtrees is the LCA of their parent nodes.

A (rooted) *subtree-prune-regraft* (rSPR) operation on an X-tree T cuts an edge $e = (x, p_x)$ where p_x denotes the parent of node x. T is divided into two subtrees T_x and T_{p_x} containing x and p_x, respectively. Then the operation adds a new node p'_x to T_{p_x} by subdividing an edge of T_{p_x} and adding a new edge (x, p'_x), making x a child of p'_x. Finally, p_x is suppressed, joining the two edges on either side of that node. See Figure 1(d) for an example. The inclusion of ρ allows for rSPR moves which move subtrees to the root of the tree.

rSPR operations give rise to a distance measure between X-trees: $d_{\mathrm{SPR}}(T_1, T_2)$ is the minimum number of rSPR operations required to transform an X-tree T_1 into T_2. For example, the trees in Figure 2 are separated by two rSPR operations. Moreover, rSPR operations

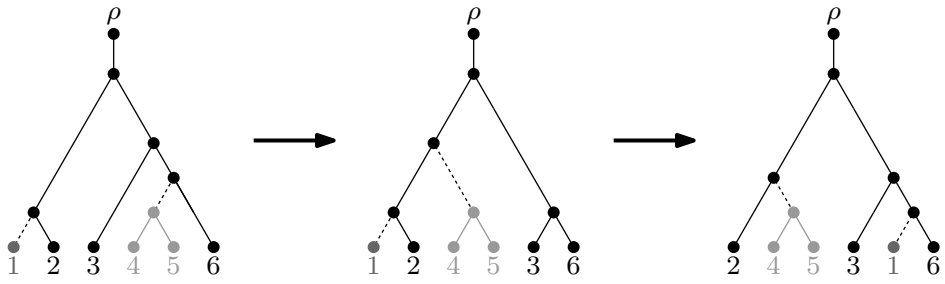


Figure 2: Two rSPR operations, each of which moves one grey subtree. The leftmost and rightmost trees are rSPR distance two apart.

naturally give rise to a graph on the set of X-trees for which this distance is simply the shortest-path graph distance. Let $\mathcal{T}_n$ be the set of trees with n leaves and label set $X = \{1, 2, \ldots n, \rho\}$. Then the rSPR graph G of $\mathcal{T}_n$ is the graph with vertex set $V(G) = \mathcal{T}_n$ and edge set $E(G) = \{(T, S) \mid d_{\mathrm{SPR}}(T, S) = 1, T \in V, S \in V\}$.

To avoid confusion between the two types of graph structures considered here, we refer to vertices of the rSPR graph as *vertices* and vertices of individual trees (i.e. leaves and internal nodes) as *nodes*. Let $N(T)$ be the set of rSPR neighbors of a tree T (this does not include T). For example, the tree T with 4 leaves in Figure 3 has 10 neighbors. We say that the degree of T is $|N(T)|$, that is, the number of trees which can be obtained from T by a single rSPR operation. We assume that all trees are bifurcating, and thus use degree to refer only to the degree of rSPR graph vertices.

Ricci-Ollivier curvature provides a rigorous yet intuitive formalization of the shape of a metric space with respect to a random walk. For the purposes of this paper, we will specialize to that space being a graph equipped with the shortest-path distance. For a more rigorous presentation in the more general setting of a Polish metric space, see [25] or the survey [26].

Let m_x and m_y be probability densities of the position of a specified random walk after one step of the random walk, starting at points x and y of a graph $G = (V, E)$, respectively. The transportation distance [37] (equivalently Wasserstein distance, or "earth movers distance" [30]) between m_x and m_y is the minimum amount of "work" required to move m_x to m_y along edges of the graph, that is

$$W_1(m_x, m_y) := \min_{\xi \in \Pi(m_x, m_y)} \sum_{\{z,w\} \subset V} d(z, w)\xi(z, w), \tag{2}$$

where $d(z, w)$ is the graph shortest-path distance ($d_{\mathrm{SPR}}(z, w)$ in our case) and $\Pi(m_x, m_y)$ is the set of densities on $V \times V$ that are m_x after projecting on the first component and m_y after projecting on the second.

The *coarse Ricci-Ollivier curvature* of x and y is then defined as:

$$\kappa(m; x, y) := 1 - \frac{W_1(m_x, m_y)}{d(x, y)}. \tag{3}$$

For the purposes of this paper, "curvature" without further specification will refer to (3). We will use $\kappa(x, y)$ to denote the curvature of the simple (uniform choice of neighbor) random walk, and use $\kappa(\mathrm{MH}; x, y)$ to indicate curvature with respect to the Metropolis-Hastings random walk sampling the uniform distribution (described in detail in Section 3.2). Positive curvature implies that the neighborhoods m_x and m_y are closer in transportation distance than point masses at x and y, zero curvature implies that they are neither closer nor farther, and negative curvature implies that m_x and m_y are more distant than point masses at x and y. Curvature thus provides an intuitive measure of the difficulty of moving between regions of the graph with a random walk.

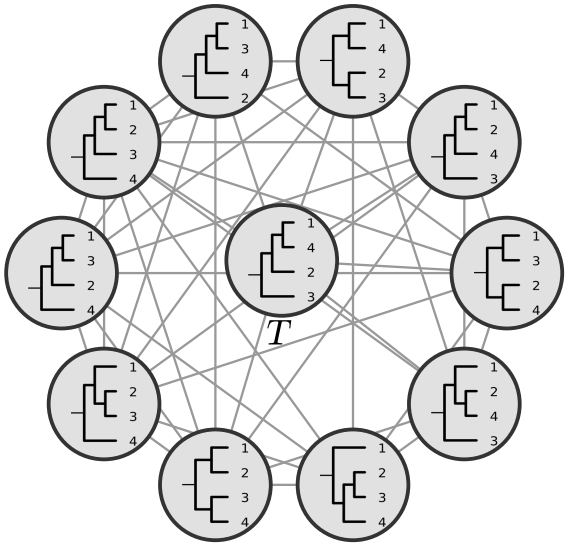


Figure 3: The neighborhood of an X-tree T with 4 leaves, showing connections with T and between neighbors.

Lin et al. [17] defined a variant definition of curvature in terms of lazy random walks which Loisel and Romon [18] dubbed the *asymptotic Ricci-Olivier curvature*. The lazy random walk only travels according to m_x with probability p and otherwise stays put. Thus the lazy mass assignment m_x^p is the sum of $p\,m_x$ and a point mass of $1-p$ on x. We denote the coarse curvature of the p-lazy random walk between two vertices x and y with respect to a random walk m by $\kappa_p(m;x,y)$. For example, $\kappa_{1/4}(m;x,y)$ describes the curvature of the lazy random walk that follows the given random walk m with probability $1/4$ and remains stationary with probability $3/4$. The asymptotic Ricci-Ollivier curvature of x and y is then:

$$\mathrm{ric}(m;x,y) := \lim_{p\to 0} \frac{\kappa_p(m;x,y)}{p}. \tag{4}$$

As above for κ, we use $\mathrm{ric}(x,y)$ as shorthand for $\mathrm{ric}(m;x,y)$ when m is the uniform lazy random walk, and $\mathrm{ric}(\mathrm{MH};x,y)$ when m is the Metropolis-Hastings random walk sampling the uniform distribution (Section 3.2). This definition of curvature is invariant of p for small p [18] and can be used to avoid parity problems on graphs where the uniform random walk is periodic without choosing a specific laziness parameter (e.g. Ollivier often considered $\kappa_{\frac{1}{2}}(x,y)$ for this purpose). As we prove in Lemma 6.7, the notions of coarse and asymptotic curvature differ only by a small factor bounded by $\frac{2}{\max(|N(x)|,|N(y)|)}$ between adjacent vertices and are equal for nonadjacent vertices.

3 Efficient algorithms for computing and sampling rSPR graphs

3.1 Computing the rSPR graph of m trees with n leaves in $\mathrm{O}(mn^3)$-time. It is necessary to have an efficient method of constructing the full rSPR graph for a fixed number of leaves in order to study it. The previous best algorithm for this problem requires $\mathrm{O}(m^2n)$ time, where m is the number of trees in the graph and n the number of leaves [44]. Here we reduce that time to $\mathrm{O}(mn^3)$. Note that for the full rSPR graph, m is the rapidly growing function $(2n-3)!!$, that is, $3\cdot 5\cdot\ldots\cdot(2n-3)$, and this is therefore a significant improvement in practice, as we demonstrate below.

In previous work [44], we constructed (unrooted) SPR graphs from subsets of m high probability trees sampled from phylogenetic posteriors to compare mixing and identify local maxima. Although the SPR distance (rooted and unrooted) is NP-hard to compute [3,12], it is fixed-parameter tractable with respect to the distance in the rooted case [3]. In particular, one can determine in $\mathrm{O}(n)$-time whether two rooted phylogenetic trees are adjacent in the rSPR graph ($\mathrm{O}(n^2)$-time for unrooted trees) using the algorithms of Whidden et al. [42–44, 46]. We applied this method comparing each of the m trees pairwise to identify adjacencies, requiring a total of $\mathrm{O}(m^2n)$-time ($\mathrm{O}(m^2n^2)$-time in the unrooted case). However, this method is impractical when applied to construct graphs with 7 or more leaves, due to the rapidly growing $\mathrm{O}(m^2)$ factor.

The key to our efficient algorithm for quickly computing dense rSPR graphs (those containing a significant portion of the full rSPR graph) lies in avoiding the pairwise comparison of non-adjacent trees and thereby shaving off an $\mathrm{O}(m)$ factor. The input to our algorithm is a set $\mathcal{T}$ of phylogenetic trees in the $\mathrm{O}(n)$-length Newick [47] representation of each tree as a string. These representations are made unique by ordering each tree so that leftmost subtrees contain the smallest alphanumeric label of descendants. We construct a mapping from each tree T_i to its order index in this list i. Begin with an empty graph G. For each tree T_i, we first add a vertex i to the graph and then use Corollary 3.4 below to enumerate the $\mathrm{O}(n^2)$ neighbors of T_i in the rSPR graph in $\mathrm{O}(n^3)$-time. This efficient enumeration procedure is the key step required to achieve our desired running time of $\mathrm{O}(mn^3)$. We use the tree to index mappings to determine whether these trees are already vertices of the graph and, if so, add an edge in the graph from T_i to each such neighbor T_j. The high-level steps are as follows, and we show in Theorem 3.1 that this algorithm is correct and can be implemented to run in the stated time.

Construct-rSPR-Graph($\mathcal{T}$)

1. Let G be an empty graph.
2. Let M be a mapping from trees to integers.
3. Let $i = 0$.
4. For each of the m trees:
   (a) Add a vertex i to G representing the current tree T_i.
   (b) Add $T_i \to i$ to M.
   (c) For each of the $\mathrm{O}(n^2)$ neighbors of T_i, enumerated using Enumerate-rSPR-Neighbors(T_i):
      i. If the current neighbor T_j is in M then add an edge $(i, M[T_j])$ to G.
   (d) $i = i + 1$.

Theorem 3.1. *The subgraph of the rSPR graph induced by a set $\mathcal{T}$ of m trees with n leaves can be constructed in $\mathrm{O}(mn^3)$-time.*

We implemented this procedure in the C++ program `dense_spr_graph` of the software package

`spr_neighbors` [40], which outputs an edge list format graph suitable for input to other software. The construction procedure reduced the time required to compute the 10,395-vertex 7-taxon rSPR graph from 2,104.68 seconds to 12.71 seconds on an Intel Core 2 Duo E7500 desktop running Ubuntu 14.04. Moreover, although we do not study the 135,135-vertex 8-taxon rSPR graph in this paper, our algorithm required only 303.45 seconds to construct it on the same hardware. Constructing the 8-taxon rSPR graph using the previous method required 377,395 seconds (more than 4 days), and thus that method is infeasible for constructing larger rSPR tree graphs. Thus, we believe our fast graph construction procedure will itself be useful for further studies of rSPR graph subsets similar to [44], as the algorithm can quickly construct rSPR graphs for any given subset of trees.

3.2 Simulating random walks on the rSPR graph. The uniform random walk moves from one vertex to one of its neighbors uniformly at random, which makes this walk more likely to sample higher degree vertices. In contrast, the Metropolis Hastings (MH) random walk with constant likelihood function proposes a move from a tree T to a neighbor tree S uniformly at random and then accepts the move according to the Hastings ratio, $\min\left(1, \frac{|N(T)|}{|N(S)|}\right)$. The MH random walk is guaranteed to sample each tree uniformly at random and is therefore representative of a phylogenetic MCMC program sampling trees under a uniform prior.

To efficiently simulate the MH random walk, we developed a linear-time algorithm for proposing rSPR moves that does not require the rSPR graph to be explicitly built and stored in memory. A naïve approach would require $O(n^3)$ time: $O(n)$ time to generate each of the $O(n^2)$ neighbors of a given tree so that one could be picked uniformly at random. To eliminate an $O(n^2)$ factor, we developed a deterministic ordering of rSPR moves with a one-to-one correspondence to rSPR neighbors, as described in the next paragraph. Given such an order, a uniform neighbor can be selected by its index in $O(n)$ time. We note that the recursive formula of Song [32] for the degree of a tree does not group rSPR moves that move a particular subtree, and thus would still require $O(n^2)$ time to select a specific rSPR neighbor by index.

We consider the distribution of rSPR moves in terms of the number of nodes contained within a subtree. Recall that a tree with n leaves has $2n-1$ total nodes (ignoring the artificial ρ node). Given a subtree R with x nodes, observe that there are $2n-1-x$ possible locations to regraft R. However, some of these moves will result in the same neighboring tree as other rSPR moves. In particular, where we call the edge connecting the subtree rooted at that node to the rest of the tree the "node's edge", we have:

i. Moving R to its sibling edge results in the same tree, not a neighboring tree,
ii. Moving R to its parent edge results in the same tree,
iii. Moving R to its grandparent edge is the same as moving its aunt to its sibling edge, and
iv. Moving R to its aunt edge is the same as moving its aunt to R's edge.

We prove in Lemma 3.2 that this list is exhaustive, that is each other pair of R and destination edge e results in a unique rSPR neighbor. We assign $(2n-1-x)-2$ moves to children of the original non-ρ root (lacking both an aunt and a grandparent), and $(2n-1-x)-4$ moves to each other non-root node. Let $N(T,u)$ denote the neighbors of T assigned to node u, obtained by moving the subtree R rooted at u. We thus achieve a new method for computing the neighborhood size:

LEMMA 3.2. *For a tree T with n leaves,*

$$|N(T)| = \sum_{u \in T} |N(T,u)|,$$

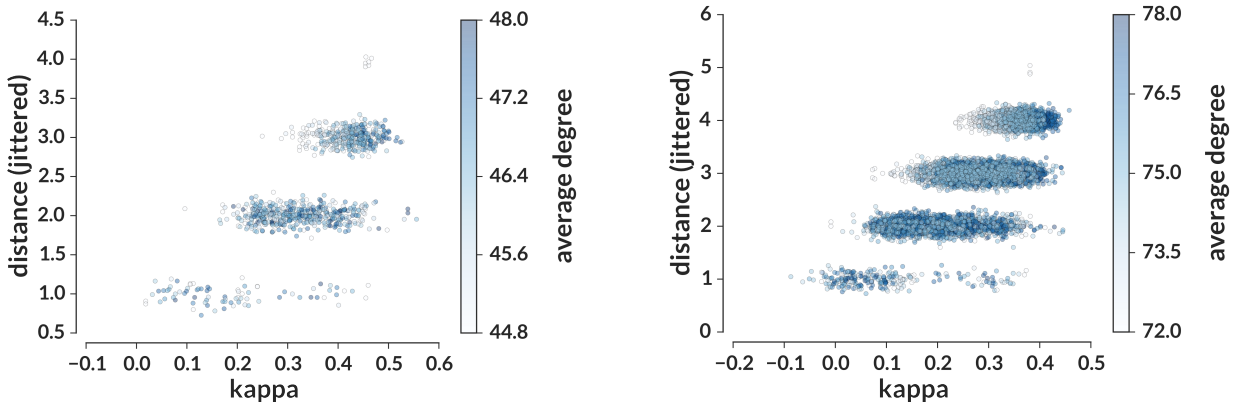
for nodes u of T, where $N(T,u)$ is as defined above, and:

$$|N(T,u)| = \begin{cases} 2n-x-5 & \text{if } depth(u) > 1, \\ 2n-x-3 & \text{if } depth(u) = 1 \\ 0 & \text{if } depth(u) \leq 0 \end{cases}.$$

In particular, this formulation implies a total ordering of rSPR moves such that every move moving the same subtree R forms a contiguous subsequence. We can thus apply the following algorithm to select a neighbor uniformly at random for a tree T:

SELECT-RSPR-NEIGHBOR(T)

1. Compute the degree of T, $|N(T)|$ using Lemma 3.2.
2. Pick a random integer r in the range $[1, |N(T)|]$.
3. Label each node u of T by its preorder number and compute the number of nodes in the subtree rooted at each u.
4. For each tree node u and while $r > 0$:
   (a) Decrease r by $|N(T,u)|$.
   (b) If $r < 0$, let S be the $|r|$ member of $N(T,u)$ and terminate the for loop.
5. Return the neighbor S.



(a) 6 taxa

(b) 7 taxa

Figure 4: Scatter plot of $\kappa(\mathrm{MH}; T_1, T_2)$ values versus $d_{\mathrm{SPR}}(T_1, T_2)$ for the rSPR graph. Color displays the average degree of T_1 and T_2. Distance values randomly perturbed ("jittered") a small amount to avoid superimposed points.

LEMMA 3.3. *An rSPR neighbor of a tree T can be chosen uniformly at random in $\mathrm{O}(n)$-time using $\mathrm{O}(n)$ space.*

Observe that this procedure can be easily adapted to explore the full neighborhood of a tree in $\mathrm{O}(n^3)$ time, which we use for Theorem 3.1. We call the resulting procedure ENUMERATE-RSPR-NEIGHBORS(T). We thus have the following corollary:

COROLLARY 3.4. *The rSPR neighbors of a tree T can be enumerated in $\mathrm{O}(n^3)$-time.*

We implemented this procedure in the C++ package `random_spr_walk` [39]. We sampled a 200,000-iteration random walk on the 4-leaf rSPR graph and a 50,000-iteration random walk on the 5-leaf rSPR graph.

4 Access times of random walks on the rSPR graph can be understood using distance, degree, and curvature

4.1 Computing curvature values. To compute curvature values, we first used `dense_spr_graph` to compute the rSPR graph for four to seven leaves, as discussed in Section 3.1. We then computed curvatures for given pairs of trees directly, by using linear programming [18] to compute the minimal mass transport W_1 using the SAGE [35] front-end to the GLPK [1] solver; code can be found in [20] which grew from the code described in [18].

This would have required an enormous amount of computation to directly compute curvatures for the $((2n-3)!!)^2$ pairs of trees with n leaves, even for the small values of n we consider here. We instead exploited the fact that pairs of trees which are equivalent modulo label renumbering are symmetric in the rSPR graph and therefore guaranteed to have the same curvature. For example, the pairs $\{(((1,2),3),4),((1,2),(3,4))\}$ and $\{(((1,4),2),3),((1,4),(2,3))\}$ are the same after relabeling, so their curvatures are the same. We thus directly computed curvature values for one representative pair from each such equivalence class, or *tanglegram* [36]; the group-theoretic enumeration methods are described in a manuscript in preparation, and the SAGE [35] and GAP4 [9] code is at [21].

We find a wide variation in curvature among tanglegrams (Figure 4). Curvature values tended to increase with increasing rSPR distance, and their variance decreased with increasing distance. Neighboring trees achieved minimum curvature values for a given number of leaves, and we found maximum curvature values between trees at maximum distance or one rSPR move closer than the maximum. This suggests that the increased difficulty of moving between trees with a random walk due to distance may be tempered somewhat by larger curvature in the highly connected rSPR graph.

Larger rSPR graphs tended to have smaller curvature values. Indeed, the 7-leaf rSPR graph contained adjacent pairs of trees with negative curvature. Such pairs indicate difficult paths for phylogenetic searches, which may be exacerbated by likelihood or branch length constraints.

4.2 Access time simulation. The access time for a pair of vertices in a graph is the (random) number of iterations required to go from one of the vertices to the other in a random walk [19]; we were interested in the

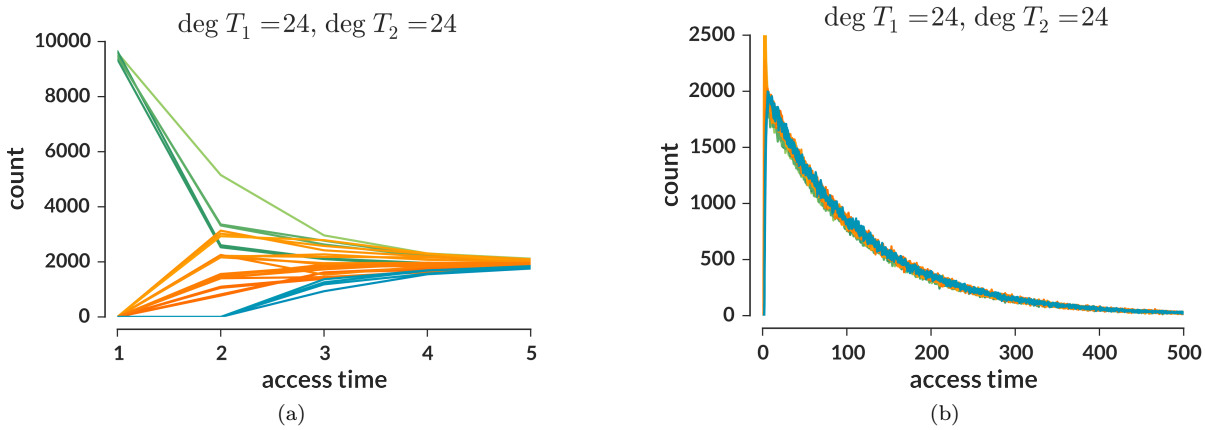


Figure 5: Distribution of rSPR MH access times for those pairs of 5-taxon trees with degree 24 that are not simple inclusions of 4-taxon pairs of trees. Color signifies rSPR distance between the trees, with green, orange, and blue signifying distances of 1, 2, and 3, respectively; the saturation of the color shows coarse curvature $\kappa(\text{MH};\cdot,\cdot)$, such that increased saturation (i.e. darker color) indicates a smaller κ.

Table 1: p-values for ordinary least squares linear multiple regression of rSPR mean access time against degree and distance (two-tailed t-test of regression coefficient). The p-values for 7 taxa are smaller than the machine precision used to calculate them.

variable	5 taxa	6 taxa	7 taxa
T_1 degree	2.425e-07	2.726e-55	0
T_2 degree	0.04367	4.302e-21	0
d_{rSPR}	5.026e-09	1.104e-44	0

Table 2: p-values for ordinary least squares linear multiple regression of rSPR δ_1 against degree, distance, and κ (two-tailed t-test of regression coefficient).

variable	5 taxa	6 taxa	7 taxa
T_1 degree	9.376e-05	2.944e-07	5.51e-09
T_2 degree	0.2366	0.1432	0.1687
d_{rSPR}	5.151e-06	0.0007557	3.276e-23
κ(MH)	4.462e-06	1.436e-22	1.459e-46

connection between curvature and access time. In previous work, we computed mean access times (MAT) between pairs of trees in MCMC random walks: the mean number of iterations required to move from one tree to the other. We applied this work to demonstrate the influence of SPR graph structure on real MCMC posteriors sampled with MrBayes [44] using `sprspace` [41].

Here, to gain more insight, we used simulation to approximate the entire access time distribution. Again we use the insight that the access time for a pair of trees with a simple random walk does not depend on the actual labeling of those trees, but rather only on their relative labeling. Thus rather than enumerate access times between trees, which would have required a tremendous amount of memory and computational power to obtain accurate estimates, we enumerate times between pairs of trees in a tanglegram. To calculate the empirical distributions of access times we aggregate all access times for the same tanglegram using our group-theoretic methods [21].

We find that the mean access time between trees T_1 and T_2 is determined by $|N(T_1)|$ and $|N(T_2)|$ (Table 1). Furthermore, plotting the distribution of access times between pairs of trees with respect to their distance and curvature hints that smaller κ slightly shifts the distribution of access times towards larger access times (Fig. 5(a)). We quantify this effect by defining δ_1 to be the difference between the first pair of access time counts such that the second entry in the pair is nonzero. For example, δ_1 for distance 1 pairs (green lines in Fig. 5) is the count for time 1 minus the count for time 2, while δ_1 for distance 3 pairs (blue lines in Fig. 5) is the count for time 2 minus the count for time 3. Regression finds a clear influence of κ on δ_1 (Table 2). This confirms the intuitive interpretation of $\kappa(T_1, T_2)$ as quantifying the propensity of a random walk to go from T_1 to T_2 relatively directly, certainly before the random walk achieves stationarity. On the other hand, if the random walk starting from T_1 does not quickly arrive at T_2 and instead achieves stationarity, the original position of the random walk is forgotten, and the access time is then a standard exponentially distributed waiting time for an

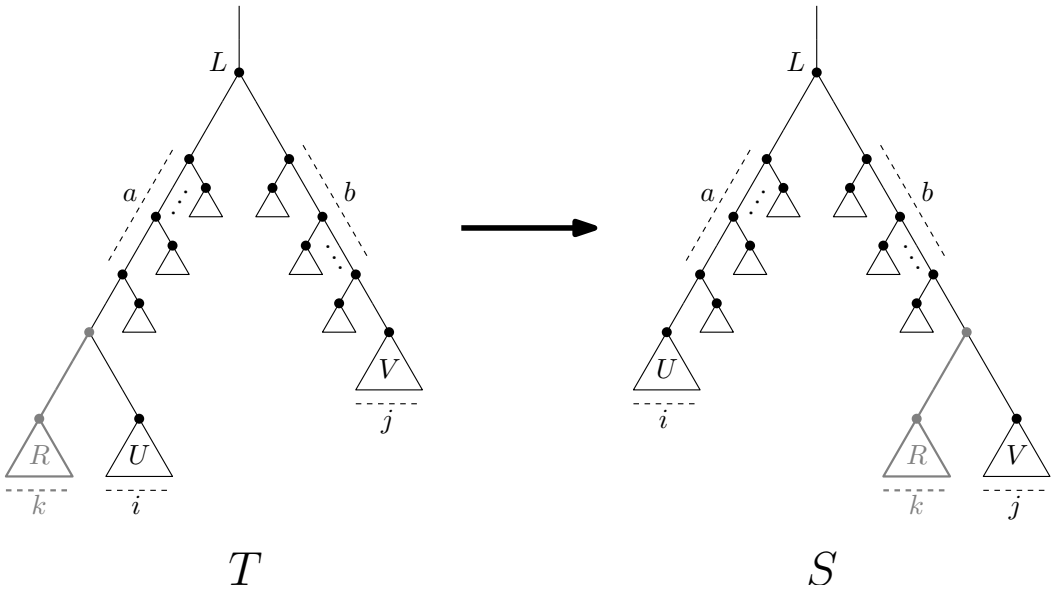


Figure 6: An rSPR move labelled as in Lemma 5.3. Moving the grey subtree R from its position adjacent to U in tree T to its position adjacent to V in tree S changes the rSPR degree by $2\,(k(a-b)+i-j)$.

event in a Poisson process (Fig. 5(b)).

The analysis can be reproduced by invoking the SCons (`http://scons.org/`) build tool and running the cells in an IPython notebook; instructions are in the repository README file.

5 Rooted SPR Neighborhoods

Having made the connection between curvature values and access times on rSPR graphs, we now consider curvature theoretically. We begin by bounding differences between degrees, and then continue by considering features relevant to the earth mover's distance that we call "squares" and "triangles" in the rSPR graph. Many of our results in this section follow from a characterization of the change in degree and distribution of permissible rSPR moves after an rSPR move is applied.

Lemma 5.1. (Song [32]) *For a tree T with n leaves:*

i. $|N(T)| = 3n^2 - 13n + 14$, *if T is a ladder tree,*
ii. $|N(T)| = 4(n-2)^2 - 2\sum_{m=1}^{n-2} \lfloor \log_2(m+1) \rfloor$, *if T is a balanced tree, and*
iii. $3n^2 - 13n + 14 \leq |N(T)| \leq 4(n-2)^2 - 2\sum_{m=1}^{n-2} \lfloor \log_2(m+1) \rfloor$, *otherwise.*

We now bound the ratio and difference of rSPR degree between two trees with n leaves.

Lemma 5.2. *Let T,S be trees with $n \geq 3$ leaves, and assume w.l.o.g. that $|N(T)| \leq |N(S)|$. Then:*

i. $\frac{|N(T)|}{|N(S)|} \geq 3/4$, *and*
ii. $|N(S)| - |N(T)| \leq n^2 - 5n + 6$.

We can improve these bounds in the case of adjacent trees. To do so, we require the following lemma that characterizes how the degree of a tree changes after an rSPR operation. See Figure 6 for an illustration.

Lemma 5.3. *Let T and S be trees such that S can be obtained from T by moving a subtree R with k leaves from its position adjacent to subtree U to a location adjacent to subtree V. Let L be the $LCA(U,V)$ in T. Let a be the number of intermediate nodes on the path from the parent of R to L in T, excluding endpoints. Similarly, let b be the number of intermediate nodes on the path from V to L in T, excluding endpoints. Let i be the number of leaves in U and j be the number of leaves in V, excluding any leaves of R. Then the degrees of T and S differ by:*

$$2\,(k(a-b)+i-j)\,.$$

Moreover, we can use these ideas to determine the number of rSPR moves that are, in some respects, independent of a given rSPR move. That is, for two trees S and T differing by a single rSPR move, we wish to know the number of rSPR moves that are applicable to both trees rather than unique to one of the trees. To formalize this concept, consider pairs of trees $T' \in N(T)$ and $S' \in S(T)$ such that $d_{\mathrm{SPR}}(T', S') = 1$. The number of such "squares" involving two adjacent trees will play a key role in our curvature bounds, as they push the curvature of those trees towards 0.

COROLLARY 5.4. *Continuing with the setting and notation in Lemma 5.3, at least*

$$\gamma := \deg(T) - 2kb - 2(j-1) = \deg(S) - 2ka - 2(i-1)$$

trees in the neighborhood of T can be paired with o trees in the neighborhood of S such that the pairings are disjoint and $d_{\mathrm{SPR}}(T', S') = 1$ for each (T', S') pair.

We can now use Lemma 5.3 to improve the bounds in Lemma 5.2 for two adjacent trees.

LEMMA 5.5. *Let T, S be trees with $n \geq 3$ leaves, s.t. $|N(T)| \leq |N(S)|$ and $d_{\mathrm{SPR}}(T, S) = 1$. Then:*

i. $|N(S)| - |N(T)| \leq 2\lfloor \frac{n-2}{2} \rfloor \lceil \frac{n-2}{2} \rceil \leq \frac{1}{2}(n-2)^2$,
ii. $\frac{|N(T)|}{|N(S)|} \geq \frac{5}{6}$, $\forall n \geq 4$, *and*
iii. $\lim_{n\to\infty} \frac{|N(T)|}{|N(S)|} = \frac{6}{7}$.

Next, we bound the number of neighbors shared by two adjacent trees. The number of such "triangles" involving two adjacent trees has a key role in determining whether their curvature is positive or negative.

LEMMA 5.6. *Let T and S be trees such that $d_{\mathrm{SPR}}(T, S) = 1$. Then $|N(T) \cap N(S)| \leq 6n - 17$.*

6 Curvature

We now consider properties of the uniform (a.k.a. isotropic) random walk on the n-leaf rSPR graph. Recall that the uniform random walk begins at a tree T and moves to a tree uniformly at random from $N(T)$. Recall that the coarse uniform random walk curvature between two trees T and S is $\kappa(T, S) := 1 - \frac{W_1(m_T, m_S)}{d(T,S)}$, where $W_{1,n}$ is the mass transport term (3). For the uniform random walk, m_T is the probability measure assigning a mass of $\frac{1}{|N(T)|}$ to each of T's neighbors. Our results follow from the lemmas of Section 5.

THEOREM 6.1. *Fix a positive integer k and let R be a tree with k leaves. Let $\{T_n \mid n > k\}$ be a sequence of trees all containing R, and let $\{S_n \mid n > k\}$ be the same sequence T_n but with R cut off and attached at a different location. Then $\lim_{n\to\infty} \kappa(T_n, S_n) = 0$ for the uniform random walk on the rSPR graph.*

Next we note a simple and rough bound on the curvature of two trees with respect to their distance, then obtain a tighter bound on the maximum curvature of two adjacent trees.

LEMMA 6.2. *Let T and S be two trees. Then:*

$$\frac{-2}{d_{\mathrm{SPR}}(T, S)} \leq \kappa(T, S) \leq \frac{2}{d_{\mathrm{SPR}}(T, S)}.$$

LEMMA 6.3. *The maximum curvature between two adjacent trees with n leaves is $\frac{6n-17}{3n^2-13n+14}$.*

This bound is tight and has been verified computationally for $n \leq 7$.

It is more difficult to obtain a closer bound on the maximum curvature of nonadjacent trees. Lemma 6.2 suggests that more distant pairs of trees should have smaller curvatures than close trees as neighborhood effects decrease with respect to the increasing distance. However, our experiments with $n \leq 7$ suggest that maximum curvature tends to increase with distance (with respect to a fixed n), as a far greater fraction of the neighbors approach each other as the distance increases. Indeed, for $5 \leq n \leq 7$ the maximum curvature is obtained by pairs of trees at one less than the maximum distance. Moreover, nearly all of the neighbors of these pairs approach each either. We thus conjecture the following:

CONJECTURE 6.4. *Let k_n be the maximum curvature between two trees with n-leaves. Then:*

i. $k_n \leq \frac{2}{\Delta_{rSPR}(n)-1}$, *and*
ii. $k_n \sim \frac{2}{\Delta_{rSPR}(n)-1}$.

Proving or disproving this conjecture would go a long way toward understanding the effects of relative distance on curvature. However, we suspect that this will require a greater understanding of the distribution of tree neighborhoods with respect to one another than is currently known. Next, we bound the minimum curvature of two adjacent trees.

LEMMA 6.5. *The curvature between adjacent trees with n leaves is at least*

$$\frac{-n^2 + 2n}{3.5n^2 - 15n + 16}.$$

We further observe that the limit of our curvature lower bound is $-\frac{2}{7}$. Complete enumeration with $n \leq 7$ show that no pair of trees have curvature less than $-\frac{2}{5}$ and our bound meets or exceeds this value for $n > 7$. Moreover, the rSPR distance is a metric, so this bounds the curvature for arbitrary pairs of trees (Proposition 19 of [25]). This directly leads to the following Corollary:

COROLLARY 6.6. *The curvature between two trees is at least $-\frac{2}{5}$.*

Note that this bound is not tight (at least for small n) as it is rarely necessary to transport mass the maximum distance between unpaired trees. We also note that the lower bounds in this section do not follow from the more general setting described in

[14]. However, the pair of trees used in the proof of Lemma 6.5 will always have negative curvature, for all $n \geq 7$.

We next bound the difference between the coarse and asymptotic curvatures. Recall that $\kappa_p(T,S)$ is the coarse Ricci-Ollivier curvature between trees T and S with respect to the lazy walk that remains at a given tree with probability $1-p$ and moves with probability p. For the lazy uniform random walk, m_T is now $T \cup N(T)$, with each neighbor assigned mass $\frac{p}{|N(T)|}$ and T assigned the remaining $1-p$ mass. The asymptotic Ricci-Ollivier curvature $\mathrm{ric}(T,S)$ is $\lim_{p\to 0} \kappa_p(T,S)/p$. As we now prove, these two notions of curvature differ only by a small factor inversely proportional to the maximum degree of T and S.

LEMMA 6.7. *Let T and S be trees with n leaves. Then:*

i. $\mathrm{ric}(T,S) = \kappa(T,S)$, *if* $d_{\mathrm{SPR}}(T,S) > 1$,
ii. $\kappa(T,S) \leq \mathrm{ric}(T,S) \leq \kappa(T,S) + \frac{2}{\max(|N(T)|,|N(S)|)}$, *if* $d_{\mathrm{SPR}}(T,S) = 1$.

Finally, we bound the difference between the curvature of the uniform random walk $\kappa(T,S)$ and that of the Metropolis-Hastings (MH) random walk $\kappa(\mathrm{MH};T,S)$. Recall that this random walk proposes a move from a tree T to a neighbor tree S uniformly at random and then accepts the move according to the Hastings ratio, which in this case is $\min\left(1, \frac{|N(T)|}{|N(S)|}\right)$. The mass distribution for the MH random walk thus leaves a portion of mass at the origin tree, proportional to the relative degree difference of its higher degree neighbors. Note that the same statement and proof of Lemma 6.7 holds with $\kappa(T,S)$ and $\mathrm{ric}(T,S)$ replaced by the MH curvatures $\kappa(\mathrm{MH};T,S)$ and $\mathrm{ric}(\mathrm{MH};T,S)$, respectively.

LEMMA 6.8. *Let T and S be trees with n leaves. Then:*

$$\kappa(T,S) - \frac{1}{3d_{\mathrm{SPR}}(T,S)} \leq \kappa(\mathrm{MH};T,S)$$
$$\kappa(\mathrm{MH};T,S) \leq \kappa(T,S) + \frac{1}{3d_{\mathrm{SPR}}(T,S)}, \text{ and}$$
$$\kappa(T,S) - 1/6 \leq \kappa(\mathrm{MH};T,S) \leq \kappa(T,S) + 1/6.$$

7 Conclusion and future work

In summary, we have gone beyond graph diameter and vertex degree to substantially advance understanding of the phylogenetic rSPR graph. We did so by developing the first theoretical and computational frameworks to bound and compute Ricci-Ollivier curvature of the rSPR graph. We found that curvature, along with degree and distance, determine the early dynamics of hitting times for random walks. Moreover, we proved that rSPR graph degree changes depend quadratically on the product of the size of the regrafted subtree with its change in depth, as well as that the rSPR graph tends toward flatness with respect to rSPR moves that move asymptotically small subtrees. Finally, we proved that the coarse and asymptotic definitions of Ricci-Ollivier curvature are closely related with respect to uniform and Metropolis-Hastings walks on the rSPR graph.

In this data-free setting the stationary distribution is, unlike with real data, quite evenly spread over all trees. Correspondingly, we found that the influence of curvature is small in this case (Fig. 5(a)) and that the probability of the target node in the stationary distribution predominantly determines access times for pairs of trees (Fig. 5(b)). However, it is well known that MCMC takes a long time to approximate real phylogenetic posterior distributions even when the Bayesian credible set is small, and in fact our previous work showed significant SPR graph influence on the mixing time for phylogenetic MCMC for credible sets that had tens, hundreds or thousands of trees [44]. Thus, our next step will be to investigate curvature of MCMC with nontrivial likelihood functions, which will reduce the posterior distribution to a more realistic effective size, and in certain cases will lead to significant "bottlenecks" like those we have observed in real data. In those cases the curvature between two trees at either end of a bottleneck will describe how difficult it is to traverse the bottleneck.

Now that we have established the foundations of using curvature to understand graphs relevant for phylogenetic inference, many graph structures remain to be explored including NNI graphs, unrooted SPR graphs, graphs of ranked trees [33], graphs of BEAST [6] rooted "time-trees," and random walks on other discrete structures such as partitions [11] that can be expressed as trees.

8 Acknowledgements

The authors would like to thank Alex Gavruskin, Vladimir Minin, and Bianca Viray for helpful discussions. They are also grateful to the authors of the SAGE and GAP4 software, especially Alexander Hulpke.

References

[1] *GNU linear programming kit.* `http://www.gnu.org/software/glpk/glpk.html`.
[2] R. G. BEIKO, J. M. KEITH, T. J. HARLOW, AND M. A. RAGAN, *Searching for convergence in phylogenetic markov chain monte carlo*, Syst. Biol., 55 (2006), pp. 553–565.
[3] M. BORDEWICH AND C. SEMPLE, *On the computational complexity of the rooted subtree prune and regraft distance*, Ann. Comb., 8 (2005), pp. 409–423.

[4] R. Bouckaert, J. Heled, D. Kühnert, T. Vaughan, C.-H. Wu, D. Xie, M. A. Suchard, A. Rambaut, and A. J. Drummond, *Beast 2: a software platform for bayesian evolutionary analysis*, PLoS computational biology, 10 (2014), p. e1003537.

[5] Y. Ding, S. Grünewald, and P. J. Humphries, *On agreement forests*, J. Combin. Theory Ser. A, 118 (2011), pp. 2059–2065.

[6] A. J. Drummond, M. A. Suchard, D. Xie, and A. Rambaut, *Bayesian phylogenetics with BEAUti and the BEAST 1.7*, Mol. Biol. Evol., 29 (2012), pp. 1969–1973.

[7] J. Felsenstein, *Evolutionary trees from DNA sequences: a maximum likelihood approach*, Journal of molecular evolution, 17 (1981), pp. 368–376.

[8] E. Fredkin, *Trie memory*, Communications of the ACM, 3 (1960), pp. 490–499.

[9] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.7.7*, 2015. http://www.gap-system.org.

[10] L. J. Guibas and R. Sedgewick, *A dichromatic framework for balanced trees*, in Proceedings of the 19th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1978, pp. 8–21.

[11] D. Gusfield, *Partition-distance: A problem and class of perfect graphs arising in clustering*, Inf. Process. Lett., 82 (2002), pp. 159–164.

[12] G. Hickey, F. Dehne, A. Rau-Chaplin, and C. Blouin, *SPR distance computation for unrooted trees*, Evolutionary Bioinformatics, 4 (2008), pp. 17–27.

[13] S. Höhna and A. J. Drummond, *Guided tree topology proposals for bayesian phylogenetic inference*, Systematic Biology, 61 (2012), pp. 1–11.

[14] J. Jost and S. Liu, *Ollivier's Ricci curvature, local clustering and Curvature-Dimension inequalities on graphs*, Discrete Comput. Geom., 51 (2013), pp. 300–322.

[15] A. Joulin and Y. Ollivier, *Curvature, concentration and error estimates for Markov chain Monte Carlo*, Ann. Probab., 38 (2010), pp. 2418–2442.

[16] C. Lakner, P. Van Der Mark, J. P. Huelsenbeck, B. Larget, and F. Ronquist, *Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics*, Syst. Biol., 57 (2008), pp. 86–103.

[17] Y. Lin, L. Lu, and S.-T. Yau, *Ricci curvature of graphs*, Tohoku Mathematical Journal, 63 (2011), pp. 605–627.

[18] B. Loisel and P. Romon, *Ricci curvature on polyhedral surfaces via optimal transportation*, arXiv preprint, (2014).

[19] L. Lovász, *Random walks on graphs: a survey*, Combinatorics, Paul Erdös is Eighty, 2 (1993), pp. 1–46.

[20] F. A. Matsen IV, *gricci*. https://github.com/matsengrp/gricci, 2015. http://dx.doi.org/10.5281/zenodo.16428.

[21] ———, *tangle*. https://github.com/matsengrp/tangle, 2015. http://dx.doi.org/10.5281/zenodo.16427.

[22] E. Mossel and E. Vigoda, *Phylogenetic MCMC algorithms are misleading on mixtures of trees*, Science, 309 (2005), pp. 2207–2209.

[23] ———, *Limitations of Markov chain Monte Carlo algorithms for bayesian inference of phylogeny*, Ann. Appl. Probab., 16 (2006), pp. 2215–2234.

[24] C.-C. Ni, Y.-Y. Lin, J. Gao, and D. Gu, *Ricci curvature of the internet toplogy*, in Proceedings of the IEEE Conference on Computer Communications INFOCOM 2015, IEEE Computer Society, 2015.

[25] Y. Ollivier, *Ricci curvature of Markov chains on metric spaces*, J. Funct. Anal., 256 (2009), pp. 810–864.

[26] ———, *A survey of Ricci curvature for metric spaces and Markov chains*, Probabilistic approach to geometry, 57 (2010), pp. 343–381.

[27] D. F. Robinson, *Comparison of labeled trees with valency three*, Journal of Combinatorial Theory, Series B, 11 (1971), pp. 105–119.

[28] F. Ronquist, B. Larget, J. P. Huelsenbeck, J. B. Kadane, D. Simon, and P. van der Mark, *Comment on "phylogenetic MCMC algorithms are misleading on mixtures of trees"*, Science, 312 (2006), p. 367; author reply 367.

[29] F. Ronquist, M. Teslenko, P. van der Mark, D. L. Ayres, A. Darling, S. Höhna, B. Larget, L. Liu, M. A. Suchard, and J. P. Huelsenbeck, *MrBayes 3.2: efficient bayesian phylogenetic inference and model choice across a large model space*, Syst. Biol., 61 (2012), pp. 539–542.

[30] Y. Rubner, C. Tomasi, and L. J. Guibas, *The earth mover's distance as a metric for image retrieval*, International journal of computer vision, 40 (2000), pp. 99–121.

[31] R. Sandhu, T. Georgiou, E. Reznik, L. Zhu, I. Kolesov, Y. Senbabaoglu, and A. Tannenbaum, *Graph curvature for differentiating cancer networks*, Scientific reports, 5 (2015).

[32] Y. S. Song, *On the combinatorics of rooted binary phylogenetic trees*, Ann. Comb., 7 (2003), pp. 365–379.

[33] ———, *Properties of subtree-prune-and-regraft operations on totally-ordered phylogenetic trees*, Ann. Comb., 10 (2006), pp. 147–163.

[34] D. A. Spade, R. Herbei, and L. S. Kubatko, *A note on the relaxation time of two markov chains on rooted phylogenetic tree spaces*, Statistics & Probability Letters, 84 (2014), pp. 247–252.

[35] W. Stein and D. Joyner, *SAGE: System for algebra and geometry experimentation*, ACM SIGSAM Bulletin, 39 (2005), pp. 61–64. http://sagemath.org/.

[36] B. Venkatachalam, J. Apple, K. St John, and D. Gusfield, *Untangling tanglegrams: comparing trees by their drawings*, IEEE/ACM Trans. Comput. Biol. Bioinform., 7 (2010), pp. 588–597.

[37] C. Villani, *Topics in Optimal Transportation*, Graduate studies in mathematics, American Mathematical Society, Providence, 2003.

[38] D. Štefankovič and E. Vigoda, *Fast convergence of Markov chain Monte Carlo algorithms for phylogenetic*

reconstruction with homogeneous data on closely related species, SIAM J. Discrete Math., 25 (2011), pp. 1194–1211.

[39] C. Whidden, *random_spr_walk*. https://github.com/cwhidden/random_spr_walk, 2015. http://dx.doi.org/10.5281/zenodo.16541.

[40] ———, *spr_neighbors*. https://github.com/cwhidden/spr_neighbors, 2015. http://dx.doi.org/10.5281/zenodo.16543.

[41] ———, *sprspace*. https://github.com/cwhidden/sprspace, 2015. http://dx.doi.org/10.5281/zenodo.16542.

[42] C. Whidden, R. G. Beiko, and N. Zeh, *Fast FPT algorithms for computing rooted agreement forests: Theory and experiments*, in Experimental Algorithms, P. Festa, ed., vol. 6049 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 141–153.

[43] ———, *Fixed-parameter algorithms for maximum agreement forests*, SIAM J. Comput., 42 (2013), pp. 1431–1466.

[44] C. Whidden and F. A. Matsen IV, *Quantifying MCMC exploration of phylogenetic tree space*, Syst. Biol., (2015).

[45] C. Whidden and F. A. Matsen IV, *Ricci-Ollivier curvature of the rooted phylogenetic subtree-prune-regraft graph*, arXiv preprint, (2015). http://arxiv.org/abs/1504.00304.

[46] C. Whidden and N. Zeh, *A unifying view on approximation and FPT of agreement forests*, in Proceedings of the 9th International Workshop, WABI 2009, vol. 5724 of Lecture Notes in Bioinformatics, Springer-Verlag, 2009, pp. 390–401.

[47] Wikipedia, *Newick format*, 2015. [Online; accessed 30-March-2015].

A Selected Proofs

Theorem 3.1. *The subgraph of the rSPR graph induced by a set $\mathcal{T}$ of m trees with n leaves can be constructed in $O(mn^3)$-time.*

Proof. The correctness of the procedure follows by induction on the number of trees already processed, i, by observing that the procedure has constructed the subgraph of vertices $1, 2, \ldots i$ and will construct the subgraph of vertices $1, 2, \ldots i+1$.

We implement the graph with an adjacency list representation with integer-labelled vertices that supports $O(\log n)$ edge insertions and lookups (with e.g. red-black trees [10], as the vertex degrees are $O(n^2)$). As described above, the integer labels are simply the order of the input trees. Adding the vertices to the graph requires $O(m)$-time, as they are added in ascending order to the end of the vertex list, which can be stored as a fixed-size array. Adding the $O(mn^2)$ edges to the graph requires $O(mn^2 \log n)$-time. Enumerating the neighbors of T_i requires $O(n^3)$-time for each T_i, for a total of $O(mn^3)$-time. We discuss below, in Section 3.2 how to do so efficiently without considering duplicate neighbors. We store the tree to index mappings for current vertices of G in a trie [8] using Newick representation. This requires only $O(n)$-time for each tree (i.e. a total of $O(mn^3)$-time) using a standard nodes-and-pointers representation of the tree and assuming integer leaf labels (a simple $O(mn \log n)$ leaf preprocessing step could be applied to extend this procedure to phylogenetic trees with string labels). Similarly, it takes $O(n)$-time to determine the index of each of the $O(mn^2)$ considered neighbors. Therefore the graph can be constructed in $O(mn^3)$-time, as claimed.

Lemma 3.2. *For a tree T with n leaves,*

$$|N(T)| = \sum_{u \in T} |N(T,u)|,$$

for nodes u of T, where $N(T,u)$ is as defined above, and:

$$|N(T,u)| = \begin{cases} 2n-x-5 & \text{if depth}(u) > 1, \\ 2n-x-3 & \text{if depth}(u) = 1 \\ 0 & \text{if depth}(u) \leq 0 \end{cases}.$$

Proof. The statement follows if each of the neighbor assignments are disjoint, that is $N(T,u) \cap N(T,v) = \emptyset$, for all nodes u, v of T. So, suppose, for the purpose of obtaining a contradiction, that there exist two nodes u and v of T such that there exists a tree $S \in (N(T,u) \cap N(T,v))$. Then S can be obtained from T by moving the subtrees rooted at u or v. Call these U and V, respectively. This implies that both $T \setminus U = S \setminus U$ and $T \setminus V = S \setminus V$ by the definition of an rSPR operation. Then the rSPR moves that move U or V to obtain S must be nearest neighbor interchanges (NNIs), that is, rSPR moves which move their subtree to one of four locations: their grandparent edge, aunt edge, sibling's left child edge or sibling's right child edge. This implies that, without loss of generality, U is moved to its grandparent edge and V to $U's$ sibling (move type (iii)) or U is moved to its aunt edge and V to U's edge (move type (iv)), a contradiction. Thus the claim holds.

Lemma 3.3. *An rSPR neighbor of a tree T can be chosen uniformly at random in $O(n)$-time using $O(n)$ space.*

Proof. We apply the above procedure. We use a standard nodes-and-pointers representation of the trees, which can be constructed in $O(n)$-time from a Newick string representation and uses linear space in n. We can compute the degree of T in linear time and space using Lemma 3.2. To efficiently compute $|N(T,u)|$ for

each node u of T, we require the number of nodes x in the subtree rooted at u. We pre-compute these by (1) labeling each node with its preorder number in a preorder traversal and (2) summing the number of descendant nodes in a postorder traversal and storing the results in an array indexed by preorder number. Both of these traversals require $O(n)$-time. There are $2n-1 = O(n)$ nodes of T, and $|N(T,u)|$ can be computed in constant time using the subtree sizes. Moreover, the tree S can be found in $O(n)$-time by iterating over the edges of T that are not contained within u's subtree to select the corresponding rSPR destination. Finally, we require linear time to apply the chosen rSPR operation which entails removing a node, adding a node, and updating a constant number of pointers. Thus, the for loop requires linear time. By Lemma 3.2 the chosen tree is an rSPR neighbor of T and is chosen uniformly at random. Therefore, the procedure uses linear time and space and selects an rSPR neighbor of T uniformly at random.

LEMMA 5.5. *Let T, S be trees with $n \geq 3$ leaves, s.t. $|N(T)| \leq |N(S)|$ and $d_{\mathrm{SPR}}(T,S) = 1$. Then:*

i. $|N(S)| - |N(T)| \leq 2\lfloor \frac{n-2}{2} \rfloor \lceil \frac{n-2}{2} \rceil \leq \frac{1}{2}(n-2)^2$,
ii. $\frac{|N(T)|}{|N(S)|} \geq \frac{5}{6}$, $\forall n \geq 4$, *and*
iii. $\lim_{n\to\infty} \frac{|N(T)|}{|N(S)|} = \frac{6}{7}$.

Proof. We first prove (i). By Lemma 5.3, $|N(S)| - |N(T)| = 2(k(a-b)+i-j)$. This value is maximized by making L the root and minimizing b, namely by setting $b = 0$. The resulting equation $2(ka+i-j)$ is similarly maximized by setting $i = 1$ (which allows us to increase a) then maximally balancing the terms in the product ka as follows.

There are two cases, depending on whether the subtree of k leaves is moved to the root or not. If not, then we set $j = 1$ and split the remaining $n-b-i-j = n-2$ leaves between k and a in as balanced a way as possible, giving (i). Note that this corresponds to moving the bottom subtree of $\lfloor \frac{n-2}{2} \rfloor$ or $\lceil \frac{n-2}{2} \rceil$ leaves in a ladder tree to the root-most leaf of the tree.

If the subtree of k leaves is moved to the root, then we do not need to exclude the target branch from k and a, gaining an additional leaf to balance the product ka at the cost of increasing j. This corresponds to moving the bottom subtree of $\lfloor \frac{n}{2} \rfloor$ or $\lceil \frac{n}{2} \rceil$ leaves in a ladder tree to the root. Namely, we have $2(ka+1-j)$, where $j = n-k = a+1$. Let $\Delta N = |N(S)| - |N(T)|$. In both cases we have

$$\Delta N \leq 2 \left\lfloor \frac{n-2}{2} \right\rfloor \left\lceil \frac{n-2}{2} \right\rceil,$$

proving (i).

The relative change in degree, $\frac{|N(T)|}{|N(S)|}$, can also be written as $\frac{|N(T)|}{|N(T)|+(|N(S)|-|N(T)|)}$. By (i), we have that $|N(S)|-|N(T)| \leq \frac{1}{2}(n-2)^2$, so $\frac{|N(T)|}{|N(S)|} \geq \frac{|N(T)|}{|N(T)|+\frac{1}{2}(n-2)^2}$. This bound is minimized when $|N(T)|$ is minimized, and recall by Lemma 5.1 that $|N(T)|$ is bounded below by $3n^2 - 13n + 14$. Thus

$$\frac{|N(T)|}{|N(S)|} \geq \frac{3n^2-13n+14}{3n^2-13n+14+\frac{1}{2}(n-2)^2} \geq \frac{3n^2-13n+14}{3.5n^2-15n+16}.$$

Statements (ii) and (iii) follow from this bound.

LEMMA 5.6. *Let T and S be trees such that $d_{\mathrm{SPR}}(T,S) = 1$. Then $|N(T) \cap N(S)| \leq 6n - 17$.*

Proof. T and S differ by one rSPR move that moves a subtree R. Pick a neighbor $U \in N(T) \cap N(S)$ of both T and S (this intersection is not empty: T and S are different, so R contains at most $n-2$ of the leaves, thus there must be at least one other tree U obtained by moving R in T and S). Then either (i) T and U differ in the location of R, or (ii) T and U differ in the location of another subtree Q. In the latter case, $T|(X \backslash L(Q)) = S|(X \backslash L(Q))$ because T and S differ only in the location of R and $d_{\mathrm{SPR}}(T,U) = d_{\mathrm{SPR}}(S,U) = 1$. Then leaves $r' \in R$, $q' \in Q$, and $u' \in U$, for some subtree U, form a triple of T and a different triple in S. This incompatible triple can be resolved in at most $6n-17$ ways, the maximum of which is reached when Q, U, and R are themselves a "triple" of subtrees. By Lemma 3.2, each of the subtrees is assigned to at most $2n-6$ unique moves. Moreover, one additional overlapping move also moves one of the subtrees (that of the aunt of the LCA of the three subtrees). The number of shared neighbors is thus at most $3(2n-6)+1 = 6n-17$. Note that this bound is tight when, for example, T and S are ladders with a different configuration of 3 leaves at maximum depth.

THEOREM 6.1. *Fix a positive integer k and let R be a tree with k leaves. Let $\{T_n \mid n > k\}$ be a sequence of trees all containing R, and let $\{S_n \mid n > k\}$ be the same sequence T_n but with R cut off and attached at a different location. Then $\lim_{n\to\infty} \kappa(T_n, S_n) = 0$ for the uniform random walk on the rSPR graph.*

Proof. Because $d(T_n, S_n) = 1$, we will prove the theorem by showing that the mass transport term $W_{1,n}$ sits between two bounds, each of which has limit 1 as n goes to infinity.

To start we demonstrate the theorem in the case that T_n and S_n have the same number of neighbors.

First we claim that $W_{1,n}$ is bounded above by $(|N(T_n)|+ \mathrm{O}(kn))/|N(T_n)|$ by exhibiting a mass transport program satisfying that bound. Let (T'_n, S'_n) be any of the γ pairs of neighbors of (T_n, S_n) which are one rSPR move apart as per Corollary 5.4. We pair these trees in the mass transport. There are $\mathrm{O}(kn)$ trees unmatched by this pairing, and we can pair each of them arbitrarily with another tree of distance at most 3. Thus, $W_{1,n}$ is bounded above by $(|N(T_n)| + \mathrm{O}(kn))/|N(T_n)|$.

A lower bound is also available because we can't do better than distance 1 for all trees except for shared neighbors, of which there are $\mathrm{O}(n)$ by Lemma 5.6. By ignoring these trees we get a lower bound of $(|N(T_n)| - (\mathrm{O}(n)))/|N(T_n)|$ for $W_{1,n}$.

The desired control of $W_{1,n}$ is thus obtained because $|N(T_n)|$ is quadratic in n.

Now we prove the theorem when the number of neighbors differ. Assume without loss of generality that $|N(T_n)| < |N(S_n)|$. By Lemma 5.3, $|N(S_n)| - |N(T_n)| = 2(k(a-b)+i-j)$, where each of $\{a, b, i, j\}$ is less than n. Thus, $|N(S_n)| - |N(T_n)| = \mathrm{O}(kn)$. We again pair neighbor T'_n of T with neighbor S'_n of S such that $d_{\mathrm{SPR}}(T'_n, S'_n) = 1$ but, as $|N(T_n)| < |N(S_n)|$ we can only account for at most $|N(T_n)|/|N(S_n)|$ of the mass directly and may have to move the $(|N(S_n)| - |N(T_n)|)/|N(S_n)|$ remainder to trees a distance at most 3. Thus, $W_{1,n}$ is bounded above by $(|N(T_n)| + \mathrm{O}(kn))/|N(S_n)| = (|N(S_n)| + \mathrm{O}(kn))/|N(S_n)|$. We again bound $W_{1,n}$ from below with $(|N(T_n)| - \mathrm{O}(n))/|N(T_n)|$ by ignoring the mass in common neighbors of T_n and S_n. The theorem again follows because $|N(T_n)|$ is quadratic in n.

LEMMA 6.5. *The curvature between adjacent trees with n leaves is at least*

$$\frac{-n^2+2n}{3.5n^2-15n+16}.$$

Proof. In light of Corollary 5.4, the optimal mass transport cost is maximized (and therefore curvature minimized) across adjacent trees T and S by a combination of two effects: trees that cannot be paired at distance 1 and mass that must be moved between unpaired trees due to differing degrees of T and S. As we will show, these effects can be optimized simultaneously. To bound these effects, let m be the maximum (across T and S) proportion of mass that cannot be moved between adjacent neighbors of those trees. We can bound the mass transport cost from above by $1 + 2m$ because pairs of neighbors of adjacent trees are at most distance 3 apart. This gives a lower bound of $1 - (1+2m)/1) = -2m$ on the curvature.

By Lemmas 5.3 and 5.5, the latter effect is maximized when the relative degree change is maximized. By Corollary 5.4, there are at most $\gamma := |N(T)| - 2ka - 2(i-1)$ paired trees, bounding the former effect. We now construct a pair of trees that maximizes both effects. Let S be the ladder tree with degree $3n^2 - 13n + 14$ and T be the adjacent tree constructed by moving the lower $\lfloor \frac{n}{2} \rfloor$ leaves of S to the root. T has degree at most $3.5n^2 - 15n + 16$. There are thus $2ka + 2(i-1) = 2\left(\lceil \frac{n-2}{2} \rceil \lfloor \frac{n}{2} \rfloor + (1-1)\right) \leq \frac{1}{2}n^2 - n$ unpaired neighbors, the maximum possible. Moreover, as shown by Lemma 5.3 this pair of trees obtains the maximum (absolute and relative) degree change. Thus, the maximum m is:

$$\frac{\frac{1}{2}n^2 - n}{3.5n^2 - 15n + 16}.$$

The claim follows from multiplying this value by -2.

LEMMA 6.7. *Let T and S be trees with n leaves. Then:*

i. $\mathrm{ric}(T,S) = \kappa(T,S)$, *if* $d_{\mathrm{SPR}}(T,S) > 1$,
ii. $\kappa(T,S) \leq \mathrm{ric}(T,S) \leq \kappa(T,S) + \frac{2}{\max(|N(T)|,|N(S)|)}$, *if* $d_{\mathrm{SPR}}(T,S) = 1$.

Proof. We first prove the lower bound in the uniform case, that is $\kappa(T,S) \leq \mathrm{ric}(T,S)$. Let $W_1(T,S)$ be the mass transport cost in the uniform case, and $W'_1(T,S)$ be the same for the lazy uniform case with parameter p. Recall that $\kappa(T,S) = \kappa_1(T,S) = 1 - \frac{W_1(T,S)}{d_{\mathrm{SPR}}(T,S)}$, and $\kappa_p(T,S)/p = \left(1 - \frac{W'_1(T,S)}{d_{\mathrm{SPR}}(T,S)}\right) \Big/ p$. Observe that

$$W'_1(T,S) \leq pW_1(T,S) + (1-p)\, d_{\mathrm{SPR}}(T,S),$$

by the simple mass transport program obtained by treating the mass at T and S as separate from that of the neighbors. Then:

$$\begin{aligned}
\frac{\kappa_p(T,S)}{p} &= \left(1 - \frac{W'_1(T,S)}{d_{\mathrm{SPR}}(T,S)}\right) \Big/ p \\
&\geq \left(1 - \frac{pW_1(T,S) + (1-p)d_{\mathrm{SPR}}(T,S)}{d_{\mathrm{SPR}}(T,S)}\right) \Big/ p \\
&= 1 - \frac{W_1(T,S)}{d_{\mathrm{SPR}}(T,S)} \\
&= \kappa(T,S).
\end{aligned}$$

For the upper bound, we observe that $W'_1(T,S) \geq$

$$pW_1(T,S) + (1-p)\, d_{\mathrm{SPR}}(T,S) - \frac{2}{\max(|N(T)|,|N(S)|)},$$

as at most $1/\max(|N(T)|,|N(S)|)$ of the mass can remain at each of T and S, paired with the lazy remainder. The upper bound then follows analogously to the lower bound. Moreover, no mass can remain at T or S when $d_{\mathrm{SPR}}(T,S) > 1$, in which case the curvatures are equal.