Variational Combinatorial Sequential Monte Carlo for Bayesian Phylogenetics in Hyperbolic Space

Alex Chen[‡] **Phillipe Chlenski**[‡] **Kenneth Munyuza**[‡]

Antonio Khalil Moretti[†] **Christian A. Naesseth**[°] **Itsik Pe'er**[‡]

Spelman College[†] University of Amsterdam[°] Columbia University[‡]

Abstract

Hyperbolic space naturally encodes hierarchical structures such as phylogenies (binary trees), where inward-bending geodesics reflect paths through least common ancestors, and the exponential growth of neighborhoods mirrors the super-exponential scaling of topologies. This scaling challenge limits the efficiency of Euclidean-based approximate inference methods. Motivated by the geometric connections between trees and hyperbolic space, we develop novel hyperbolic extensions of two sequential search algorithms: Combinatorial and Nested Combinatorial Sequential Monte Carlo (Csmc and Ncsmc). Our approach introduces consistent and unbiased estimators, along with variational inference methods (H-Vcsmc and H-Vncsmc), which outperform their Euclidean counterparts. Empirical results demonstrate improved speed, scalability and performance in high-dimensional phylogenetic inference tasks.

1 Introduction

Phylogenetic inference, a key component of molecular biology, involves reconstructing ancestral relationships between organisms by modeling their evolutionary histories as trees. However, this task becomes computationally challenging due to the super-exponential scaling of tree topologies, which increases as $(2N-3)!!$ with the number of organisms. This scaling challenge often limits the efficiency of traditional approximate inference methods which rely on Euclidean geometry. Hyperbolic space naturally encodes hierarchical structures where inward-bending geodesics reflect paths through least common ancestors in trees, and the exponential growth of neighborhoods mirrors the super-exponential scaling of tree topologies. For example, Sarkar (2012) demonstrates that trees can be embedded into the Poincaré disk, a two-dimensional model of hyperbolic space, with arbitrarily low distortion. Motivated by geometric connections between trees and hyperbolic space, we develop novel hyperbolic extensions of two sequential search algorithms, Combinatorial and Nested Combinatorial Sequential Monte Carlo (Csmc and Ncsmc) (Wang et al., 2015; Moretti et al., 2021). Our approach introduces consistent and unbiased estimators used to form approximate posteriors and variational inference methods (H-Vcsmc and H-Vncsmc), outperforming their Euclidean counterparts. By leveraging the structure of hyperbolic geometry, both methods significantly enhance speed, scalability and performance in high-dimensional phylogenetic Inference tasks. Our contributions are as follows:

1. We develop novel hyperbolic extensions of the Csmc and Ncsmc algorithms (H-Csmc and H-Ncsmc). We define proposal distributions for phylogenetic inference in hyperbolic space and prove that the resulting estimators are consistent and unbiased.

2. We utilize these estimators to construct two novel approximate posteriors and corresponding variational inference methods, referred to as H-Vcsmc and H-Vncsmc.

3. We empirically validate our methods on high-dimensional phylogenetic inference tasks, showing that they outperform existing approximate Bayesian Inference techniques, in some cases achieving speedups of 20 to 50 times by utilizing Gpu acceleration.

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

1.1 Related Work

There is a substantial body of work on Bayesian phylogenetic inference, including local search methods such as MCMC (e.g., MrBayes (Huelsenbeck & Ronquist, 2001)), ppHmc (Dinh et al., 2017), and particle MCMC (Wang & Wang, 2020). Sequential search methods include Csmc (Wang et al., 2015) and Ncsmc (Moretti et al., 2021). A number of Variational Inference (VI) techniques have also been developed. Vcsmc (Moretti et al., 2021) is based on sequential search, while the majority of approaches are based on standard VI (Zhang & Matsen IV, 2019; Zhang, 2020; Koptagel et al., 2022; Mimori & Hamada, 2023; Zhang, 2023).

Recent work has explored phylogenetic inference in hyperbolic space through neighbor joining. Macaulay & Fourment (2024) introduced a differentiable variant of neighbor joining in hyperbolic space, while GeoPhy (Mimori & Hamada, 2023) combines hyperbolic leaf node embeddings with neighbor-joining for tree construction. Our method takes a probabilistic approach, using sequential node resampling based on conditional likelihoods to explore multiple tree structures simultaneously through variational sequential search in hyperbolic space. Our work builds on two recent advancements in machine learning within hyperbolic space. Nickel & Kiela (2017) proposed a method for embedding hierarchical structures in the Poincaré disk. Additionally, Nagano et al. (2019) developed the *Wrapped Normal*, a distribution on hyperbolic space with an analytically tractable density and differentiable parameters.

2 Background

We review key concepts from hyperbolic geometry and approximate Bayesian phylogenetic inference.

2.1 Geometry in Riemannian Manifolds

Hyperbolic geometry is a type of non-Euclidean geometry defined by constant negative curvature. This geometry can be represented through four equivalent models: the hyperboloid (or Lorentz/Minkowski model), the Poincaré disk model, the Poincaré half-plane model, and the Klein model. While our approach applies to each of these representations due to their equivalence, we focus on the Poincaré disk model due to its intuitive geometric structure and simplicity for visualization.

2.1.1 Poincaré Disk Model

The *Poincaré disk model* represents hyperbolic geometry within the unit disk in Euclidean space. This model is defined on the open unit disk, denoted as

$$\mathcal{P} \coloneqq \left\{ \mathbf{z} \in \mathbb{R}^2 : \|\mathbf{z}\| < 1 \right\} . \tag{1}$$

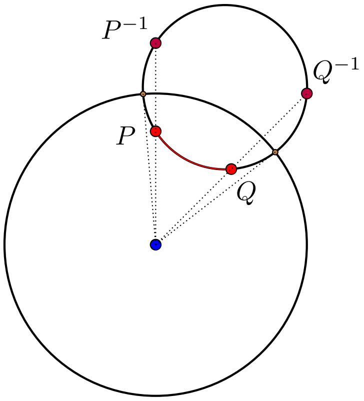

Figure 1: Geodesic in the Poincaré disk connecting points P and Q depicted in red, with their corresponding inverses P^{-1} and Q^{-1}. A circle orthogonal to the unit circle that passes through points P and Q must also intersect inverses P^{-1} and Q^{-1}, which are the reflections of P and Q with respect to the unit circle.

Let $\|\mathbf{z}\|$ denote the Euclidean norm of the point $\mathbf{z}$. The hyperbolic distance between two points $\mathbf{z}_1$ and $\mathbf{z}_2$ inside the unit disk is given by:

$$d(\mathbf{z}_1, \mathbf{z}_2) = \operatorname{arcosh}\left(1 + \frac{2\|\mathbf{z}_1 - \mathbf{z}_2\|^2}{(1 - \|\mathbf{z}_1\|^2)(1 - \|\mathbf{z}_2\|^2)}\right) . \tag{2}$$

Metric Tensor. The *metric tensor* in the Poincaré disk is defined using a conformal modification of the Euclidean metric:

$$ds^2 = \frac{4(dx^2 + dy^2)}{(1 - x^2 - y^2)^2} , \tag{3}$$

where $dx^2 + dy^2$ is the standard Euclidean metric in $\mathbb{R}^2$ and $4/(1 - x^2 - y^2)^2$ is a conformal scaling factor dependent upon distance from the origin. As points approach the boundary of the disk ($x^2 + y^2 \to 1$), distances tend to infinity and the metric becomes increasingly large.

Geodesics. In the Poincaré disk model, *geodesics* which represent the “shortest paths" in hyperbolic space are defined by arcs of circles that are orthogonal to the boundary of the disk or by diameters of the disk. Given two points $\mathbf{z}_1$ and $\mathbf{z}_2$ inside the disk, their geodesic can be constructed as the unique circle passing through both points and perpendicular to the disk boundary. If the points lie on a diameter, the geodesic is simply the Euclidean straight line segment connecting them. Further details are provided in A.1.2.

2.1.2 Parallel Transport and Exponential Map

Hyperbolic space is not a vector space, which renders the linear operations typically employed in vector spaces inapplicable. Riemannian geometry requires special tools to understand how vectors change as they move across curved surfaces. *Parallel transport* adjusts vectors along a curve while preserving their inner product structure, ensuring consistent comparisons between different points. The *exponential map* projects directions from the tangent space to the manifold along geodesics, allowing us to translate local, linear information into motion across the curved space.

Parallel transport. Given two vectors $\mathbf{x}, \mathbf{y} \in \mathcal{P}$, the parallel transport from $\mathbf{x}$ to $\mathbf{y}$ is defined as a map $\mathrm{PT}_{\mathbf{x}\to\mathbf{y}}$ from the tangent plane at $\mathbf{x}$, $T_{\mathbf{x}}\mathcal{P}$, to the tangent plane at $\mathbf{y}$, $T_{\mathbf{y}}\mathcal{P}$. This map transports a vector in $T_{\mathbf{x}}\mathcal{P}$ along the geodesic from $\mathbf{x}$ to $\mathbf{y}$ in a parallel manner, preserving the metric tensor throughout the process. When $\mathbf{x}$ is the origin μ_0, parallel transport can be expressed in simple closed form:

$$\mathrm{PT}_{\mu_0\to\mathbf{y}}(\mathbf{v}) := (1 - |\mathbf{y}|_2^2)\mathbf{v}\,, \tag{4}$$

where $\mathbf{v}$ denotes the vector being transported from the tangent space at the origin μ_0 to the tangent space at point $\mathbf{y}$ in the Poincaré disk. Further details are provided in Ganea et al. (2018) and Appendix A.1.3.

Möbius Addition. Möbius addition defines an operation for combining points taking into account the negative curvature of the manifold. This can be used to define a simple closed form for the exponential map. Within the Poincaré disk, Möbius addition reduces to:

$$\mathbf{x} \oplus \mathbf{y} := \frac{(1 + 2\langle \mathbf{x}, \mathbf{y}\rangle + ||\mathbf{y}||_2^2)\mathbf{x} + (1 - ||\mathbf{x}||_2^2)\mathbf{y}}{1 + 2\langle \mathbf{x}, \mathbf{y}\rangle + ||\mathbf{x}||_2^2 ||\mathbf{y}||_2^2}\,. \tag{5}$$

For additional details see Ungar (2009).

Exponential and Logarithmic Map. The exponential map converts directions from the tangent plane into geodesic movement on the curved surface. The exponential map and its inverse, the logarithmic map, are defined on the Poincaré disk as:

$$\exp_{\mathbf{x}}^{-1}(\mathbf{v}) := \mathbf{x} \oplus \left(\tanh\left((1 - ||\mathbf{x}||_2^2)||\mathbf{v}||_2\right) \frac{\mathbf{v}}{||\mathbf{v}||_2}\right)\,, \tag{6}$$

$$\log_{\mathbf{x}}(\mathbf{y}) := \tanh^{-1}\left(\|\mathbf{y}\|_2\right) \frac{\mathbf{y}}{\|\mathbf{y}\|_2}\,. \tag{7}$$

See Appendix A.1.3 for additional details.

2.2 Phylogenetics

We aim to infer a latent bifurcating tree that represents the evolutionary relationships among a set of observed molecular sequences. A phylogeny is characterized by both a tree topology, denoted by τ, and a corresponding set of branch lengths $\mathcal{B}$. Each edge $e \in E$ has an associated *branch length*, represented as $\beta(e) \in \mathbb{R}_{>0}$, with $\mathcal{B} = \{\beta(e)\}_{e\in E}$. The branch length reflects the rate of evolutionary change between two vertices. An *ultrametric tree* assumes constant rate of evolution along all paths from root to leaf nodes. *Nonclock trees* do not assume constant evolutionary rate.

Bayesian Phylogenetic Inference. Let $\mathbf{Y} = \{Y_1, \cdots, Y_S\} \in \Omega^{N\times S}$ represent the observed molecular sequences, where Ω denotes the set of characters, S is the sequence length, and N is the number of species. Bayesian phylogenetic inference involves specifying a prior distribution and a likelihood function over the tree topology τ, the set of branch lengths $\mathcal{B}$, and the model parameters θ to express the joint posterior:

$$P_\theta(\mathcal{B}, \tau|\mathbf{Y}) = \frac{P_\theta(\mathbf{Y}|\tau, \mathcal{B})P_\theta(\tau, \mathcal{B})}{P_\theta(\mathbf{Y})}. \tag{8}$$

Here, the prior is uniform over the possible topologies, while the branch lengths follow independent exponential distributions with rate λ_{bl}. Each site in the sequence evolves independently according to a continuous-time Markov chain characterized by a rate matrix $\mathbf{Q}$. Let $\zeta_{v,s}$ represent the genomic state at site s for species v, and define the evolutionary model for branch $b(v \to v')$ as follows:

$$P_\theta(\zeta_{v',s} = j|\zeta_{v,s} = i) = \exp\left(\beta(e)\mathbf{Q}_{i,j}\right). \tag{9}$$

The likelihood for a given phylogeny is then expressed as $P_\theta(\mathbf{Y}|\tau, \mathcal{B}) = \prod_{i=1}^{S} P_\theta(Y_i|\tau, \mathcal{B})$, and can be computed efficiently in linear time using the sum-product algorithm or Felsenstein's pruning algorithm (Felsenstein, 1981):

$$P_\theta(\mathbf{Y}|\tau, \mathcal{B}) := \prod_{i=1}^{S} \sum_{a^i} \eta(a^i_\rho) \prod_{(u,v)\in E(\tau)} \exp\left(-\beta_{u,v}\mathbf{Q}_{a^i_u, a^i_v}\right), \tag{10}$$

where ρ is the root node, a^i_u is the character assigned to node u, $E(\tau)$ denotes the set of edges in the tree τ, and η is the stationary distribution of the Markov chain. The normalization constant $P_\theta(\mathbf{Y})$ involves marginalizing over $(2N-3)!!$ topologies, which is computationally infeasible (Semple & Steel, 2003).

2.3 Variational Inference

Variational Inference (VI) is used to approximate the posterior distribution $P_\theta(\mathcal{B}, \tau|\mathbf{Y})$ when directly marginalizing latent variables is not feasible. By introducing a simpler, tractable distribution $Q_\phi(\mathcal{B}, \tau|\mathbf{Y})$, we can derive a lower bound for the log-likelihood:

$$\log P_\theta(\mathbf{Y}) \geq \mathcal{L}_{\mathrm{ELBO}}(\theta, \phi, \mathbf{Y}) := \mathop{\mathbb{E}}_{Q}\left[\log \frac{P_\theta(\mathbf{Y}, \mathcal{B}, \tau)}{Q_\phi(\mathcal{B}, \tau|\mathbf{Y})}\right]. \tag{11}$$

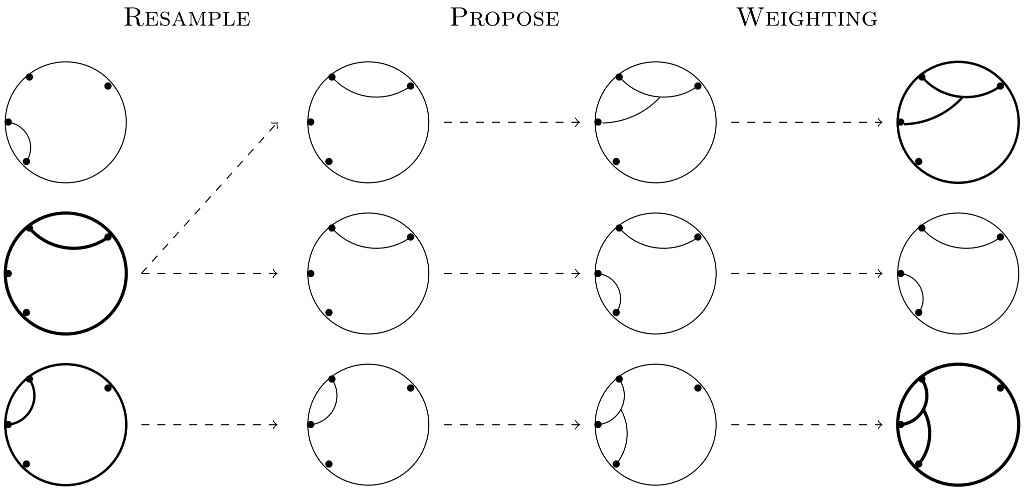

Figure 2: Overview of CSMC in the Poincaré disk. Each iteration involves three steps: (1) RESAMPLE partial states by importance weights, (2) PROPOSE extensions by connecting two trees, and (3) WEIGHT using Felsenstein's pruning. The progression of $K = 3$ samples are shown with taxa embedded near the boundary of the disk. Linewidths represent weights.

Auto-Encoding Variational Bayes (AEVB) (Kingma & Welling, 2013) trains both the distributions $Q_\phi(\mathcal{B}, \tau|\mathbf{Y})$ and $P_\theta(\mathbf{Y}, \mathcal{B}, \tau)$ simultaneously. The expectation in Eq. 11 is estimated by averaging Monte Carlo samples from $Q_\phi(\mathcal{B}, \tau|\mathbf{Y})$, with the samples reparameterized using a deterministic function of a random variable independent of ϕ.

Constructing a tractable approximation $Q_\phi(\mathcal{B}, \tau|\mathbf{Y})$ can be difficult, prompting the use of CSMC.

2.4 Combinatorial Sequential Monte Carlo

CSMC approximates a sequence of probability spaces that converge to the target distribution over $N-1$ steps, indexed by r. It uses sequential importance resampling to estimate the unnormalized target distribution π and its normalization constant $\|\pi\|$, yielding the desired posterior. This is achieved by sampling K *partial states* (see A.2.1) $\{s_r^k\}_{k=1}^K \in \mathcal{S}_r$ at each step r, gradually constructing an approximation to the full distribution:

$$\widehat{\pi}_r = \|\widehat{\pi}_{r-1}\| \frac{1}{K} \sum_{k=1}^{K} w_r^k \delta_{s_r^k}(s) \qquad \forall s \in \mathcal{S}\,. \tag{12}$$

Unlike typical SMC methods, CSMC works with both a combinatorial set of tree topologies and the continuous branch lengths. At each step, Monte Carlo samples are resampled, keeping them concentrated in regions of higher probability with importance weights given by:

$$w_r^k = w(s_{r-1}^{a_{r-1}^k}, s_r^k) = \frac{\pi(s_r^k)}{\pi(s_{r-1}^{a_{r-1}^k})} \cdot \frac{\nu^-(s_{r-1}^{a_{r-1}^k})}{q(s_r^k|s_{r-1}^{a_{r-1}^k})}\,, \tag{13}$$

where $q(s_r^k|s_{r-1}^{a_{r-1}^k})$ is a proposal distribution, and $a_{r-1}^k \in \{1, \cdots, K\}$ denotes the resampled ancestor indices chosen with probabilities proportional to the weights: $\mathbb{P}(a_{r-1}^k = i) = w_{r-1}^i / \sum_{l=1}^K w_{r-1}^l$. The factor ν^- corrects for overcounting. After resampling, new states are proposed and extended (see Fig. 2). This gives an unbiased estimate of $\|\pi\|$ converging in L_2 norm:

$$\widehat{\mathcal{Z}}_{CSMC} \coloneqq \|\widehat{\pi}_R\| = \prod_{r=1}^{R} \left(\frac{1}{K} \sum_{k=1}^{K} w_r^k \right) \to \|\pi\|. \tag{14}$$

2.5 Variational Combinatorial Sequential Monte Carlo

VCSMC (Moretti et al., 2021) learns model and proposal parameters by maximizing a lower bound on $\|\pi\|$, using CSMC (or NCSMC) as an unbiased estimator of $\|\pi\|$:

$$\mathcal{L}_{VCSMC} \coloneqq \mathop{\mathbb{E}}_{Q} \left[\log \widehat{\mathcal{Z}}_{CSMC} \right]. \tag{15}$$

In CSMC, nodes are sampled to coalesce uniformly, even when many resulting topologies have low probabilities. NCSMC (Moretti et al., 2021) uses information from one future iteration to guide the exploration of partial states, providing an exact approximation of the *locally optimal proposal* (Doucet et al., 2000; Naesseth et al., 2019b). See A.5 for details of NCSMC and V-NCSMC.

3 Methodology

In 3.1, we outline the methodology for embedding sequences into hyperbolic space. In 3.2 we introduce the algorithm that samples embedded points to construct a tree, while 3.3 defines the approximate posterior. In 3.4, we describe the process of scoring a tree and assigning importance weights. Finally, **??** extends the process to Ncsmc.

3.1 Embedding Sequences

We embed sequences into the Poincaré disk by mapping sequences Y_n into hyperbolic space while preserving pairwise relationships (Nickel & Kiela, 2017). This involves minimizing a loss function to align hyperbolic distances in the Poincaré disk with original Hamming distances:

$$\mathcal{L} = \sum_{i,j} \left(d_{\mathbb{H}}(\mathbf{z}_i, \mathbf{z}_j) - d_H(\mathbf{x}_i, \mathbf{x}_j) \right)^2 , \tag{16}$$

where $d_{\mathbb{H}}(\mathbf{z}_i, \mathbf{z}_j)$ is the hyperbolic distance between embedded points $\mathbf{z}_i$ and $\mathbf{z}_j$ in the Poincaré disk, and $d_H(\mathbf{x}_i, \mathbf{x}_j)$ is the Hamming distance between original points $\mathbf{x}_i$ and $\mathbf{x}_j$. Alternatively, a multilayer perceptron (MLP) can be employed to map sequences directly into the Poincaré disk (Marblestone et al., 2017).

3.2 Hyperbolic CSMC

After embedding the leaf nodes in the Poincaré disk, two nodes are sampled for coalescence. To assign the parent node's embedding, we parameterize the geodesic arc $\gamma(\mathbf{v}_{L,r}, \mathbf{v}_{R,r})$ between the two sampled child nodes $\mathbf{v}_{L,r}$ and $\mathbf{v}_{R,r}$, and assign $\mathbf{v}_p$ to the point on this curve that is closest to the origin μ_0:

$$\mathbf{v}_p \leftarrow \underset{\mathbf{v} \in \gamma(\mathbf{v}_{L,r}, \mathbf{v}_{R,r})}{\arg\min} \; d(\mu_0, \mathbf{v}) \, . \tag{17}$$

The minimum geodesic distance is obtained using non-ambient distance coordinates. Conveniently, the distance to the origin is monotonic in the Euclidean norm; therefore, with a parameterization of the geodesic, we can simply minimize the Euclidean distance $d(\mu_0, \mathbf{v}_p)$.

Once the nodes to coalesce have been sampled, the lengths of the left and right branches, denoted by $\beta_{L,r}$ and $\beta_{R,r}$, can be defined as the geodesic distances to the embedding of $\mathbf{v}_p$. However, this results in identical branch lengths for trees with equivalent topologies. Phylogenetic models typically account for variation in branch lengths across equivalent topologies. To account for this, we use the embedded parent node $\mathbf{v}_p$ to define a proposal distribution on the Poincaré disk.

3.2.1 Sampling Branch Lengths

The Lebesgue measure is uniform in $\mathbb{R}^2$; however, the hyperbolic measure is non-uniform in the Poincaré disk. As $x^2 + y^2 \to 1$, the metric $ds^2 \to \infty$. Moving toward the disk boundary causes the measure of a grid square to increase exponentially, which has significant implications for sampling.

Algorithm 1 Sampling Trees in Hyperbolic Space

```
1: Input: Coalescing points $\mathbf{v}_{L,r}$, $\mathbf{v}_{R,r}$ on $\mathcal{P}$
2: Find the point on the geodesic arc between $\mathbf{v}_{L,r}$ and $\mathbf{v}_{R,r}$ that is closest to the origin:
       $\mathbf{v}_p \leftarrow \underset{\mathbf{v} \in \gamma(\mathbf{v}_{L,r}, \mathbf{v}_{R,r})}{\arg\min} \; d(\mu_0, \mathbf{v})$
3: Sample a normal distribution in $\mathbb{R}^2$ to yield a vector in the tangent plane at the origin, $T_{\mu_0}\mathcal{P}$:
       $\mathbf{v}^k \sim \mathcal{N}(0, \Sigma)$
4: Parallel transport the tangent vector from the origin, $\mu_0$, to the point $\mathbf{v}_p$:
       $\tilde{\mathbf{v}}^k \leftarrow PT_{\mu_0 \to \mathbf{v}_p}(\mathbf{v}^k) = (1 - \|\mathbf{v}_p\|_2^2)\, \mathbf{v}^k$
5: Apply the exponential map to the transported vector:
       $\dot{\mathbf{v}}^k \leftarrow \exp_{\mathbf{v}_p}(\tilde{\mathbf{v}}^k)$
       $= \mathbf{v}_p \oplus \left( \tanh\left( (1 - \|\mathbf{v}_p\|_2^2)\, \|\tilde{\mathbf{v}}^k\|_2 \right) \frac{\tilde{\mathbf{v}}^k}{\|\tilde{\mathbf{v}}^k\|_2} \right)$
6: Compute the left and right branch lengths using geodesic distance (Eq. 2):
       $\beta^k_{L,r} \leftarrow d(\mathbf{v}_{L,r}, \dot{\mathbf{v}}^k)\, ,\, \beta^k_{R,r} \leftarrow d(\mathbf{v}_{R,r}, \dot{\mathbf{v}}^k)$
7: Output: Sampled parent node embedding $\dot{\mathbf{v}}^k$ and branch lengths $\beta^k_{L,r}$ , $\beta^k_{R,r}$
```

We define a proposal distribution over parent node embeddings and branch lengths by adapting the *Wrapped Normal* ($\mathcal{WN}$) distribution introduced by Nagano et al. (2019). Algorithm 1 provides the pseudocode for sampling parent node embeddings in hyperbolic space. Given two coalescing embedded points $\mathbf{v}^k_{L,r}$ and $\mathbf{v}^k_{R,r}$ where (L, R) denotes the left and right child nodes and r denotes the rank event, the subroutine begins by identifying the point in the geodesic between $\mathbf{v}^k_{L,r}$ and $\mathbf{v}^k_{R,r}$ that is closest to the origin (Step 2). To sample the location of a parent node, we sample in the tangent plane at the origin where the geometry is simpler (Step 3), and then use parallel transport to move the samples to the desired mean (Step 4). Next, we apply the exponential map to project the samples back onto the manifold (Step 5). Finally, lengths of the left and right branches $\beta^k_{L,r}$, $\beta^k_{R,r}$ are obtained using the geodesic distance between the embedding of the sampled parent and the left and right children (Step 6). Pseudocode for the full hyperbolic Csmc (H-Csmc)

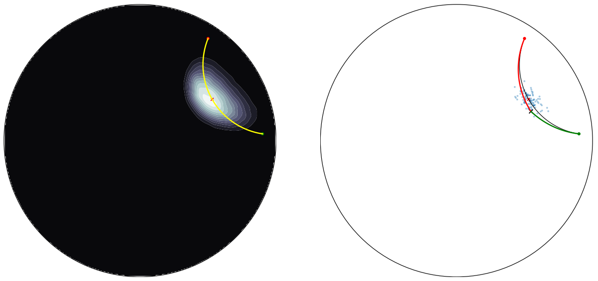

Figure 3: **Left:** Contours of the $\mathcal{WN}$ distribution over parent node embedding for two coalescing leaves. **Right:** After drawing $\mathbf{v}^k \sim \mathcal{N}(0, \Sigma)$, the vector is projected via $\mathrm{proj}_{\mathbf{v}_p} := \exp_{\mathbf{v}_p} \circ \mathrm{PT}_{\mu_0 \to \mathbf{v}_p}$. Left and right branch lengths are mapped via geodesic distance.

procedure is provided in Algorithm 3 of A.4.

Fig. 3 (left) shows the contours of a $\mathcal{WN}$ distribution on the Poincaré disk for the parent node $\dot{\mathbf{v}}^k$ of two child nodes, $\mathbf{v}^k_{L,r}$ and $\mathbf{v}^k_{R,r}$. Note the curved shape due to the concentration of the measure with respect to the geodesic distance. Fig. 3 (right) displays samples from the proposal distribution $\dot{\mathbf{v}}^k \sim \mathcal{WN}(\mathbf{v}_p, \Sigma)$, which implicitly defines a distribution over the branch lengths $\beta^k_{L,r}$ and $\beta^k_{R,r}$. Let $\mathrm{proj}_{\mathbf{v}_p} := \exp_{\mathbf{v}_p} \circ \mathrm{PT}_{\mu_0 \to \mathbf{v}_p}$. The log density can be expressed as follows:

$$\log p(\dot{\mathbf{v}}^k) = \log p(\mathbf{v}_p) - \log\det\left(\frac{\partial\, \mathrm{proj}_{\mathbf{v}_p}}{\partial\, \mathbf{v}_p}\right). \tag{18}$$

The determinant can be computed in $\mathcal{O}(n)$:

$$\det\left(\frac{\partial\, \mathrm{proj}_{\mathbf{v}_p}(\mathbf{v}^k)}{\partial\, \mathbf{v}_p}\right) \tag{19}$$
$$= \det\left(\frac{\partial\, \exp_{\mathbf{v}_p}(\tilde{\mathbf{v}}^k)}{\partial\, \tilde{\mathbf{v}}^k}\right) \cdot \det\left(\frac{\partial\, \mathrm{PT}_{\mu_0 \to \mathbf{v}_p}(\mathbf{v}^k)}{\partial\, \mathbf{v}_p}\right).$$

Since each operation is differentiable, we can backpropagate through all necessary transformations.

3.3 Approximate Posterior

The proposal returned by H-Vcsmc factorizes as:

$$Q\left(s^{1:K}_{1:R}, a^{1:K}_{1:R-1}\right) := \tag{20}$$
$$\prod_{k=1}^{K} q(s^k_1) \times \prod_{r=2}^{R} \prod_{k=1}^{K} \left[\frac{w^{a^k_{r-1}}_{r-1}}{\sum_{l=1}^{K} w^l_{r-1}} \cdot q\left(s^k_r | s^{a^k_{r-1}}_{r-1}\right)\right],$$

where the term for each step is the product of Uniform and $\mathcal{WN}$ terms for topologies and embeddings:

$$q(s^k_r | s^{a^k_{r-1}}_{r-1}) := q(\tau^k_r | \tau^{a^k_{r-1}}_{r-1}) \cdot q(\dot{\mathbf{v}}^k_r | \dot{\mathbf{v}}^{a^k_{r-1}}_{r-1}, \tau^{a^k_{r-1}}_{r-1}). \tag{21}$$

The probability that a point came from the $\mathcal{WN}$ distribution is defined by inverting $\mathrm{proj}_{\mathbf{v}_p}$ using parallel transport and the logarithmic map. Conveniently, since branch lengths depend only on $\dot{\mathbf{v}}^k$, and $\dot{\mathbf{v}}_k = \mathrm{proj}_{\mathbf{v}_p}$ we can evaluate the probability $N(\mathbf{v}^k | \mu_0, \Sigma)$ under the appropriate change of coordinates. See A.3 for the details of defining and evaluating $q(\dot{\mathbf{v}}^k_r | \dot{\mathbf{v}}^{a^k_{r-1}}_{r-1}, \tau^{a^k_{r-1}}_{r-1})$.

3.4 Scoring Tree Likelihoods

Algorithm 1 maps a point in the Poincaré disk to branch lengths, however the target π is defined on (β, τ), while the proposal q is defined on $(\dot{\mathbf{v}}, \tau)$. To define importance weights, a change of variables is needed between branch lengths and the sampled points:

$$w^k_r := \frac{\pi(s^k_r)}{\pi(s^{a^k_{r-1}}_{r-1})} \cdot \frac{\nu^-(s^{a^k_{r-1}}_{r-1})}{q(s^k_r | s^{a^k_{r-1}}_{r-1})} \cdot \det\left(\frac{\partial \beta^k_{i,r}(\dot{\mathbf{v}}^k)}{\partial \dot{\mathbf{v}}^k}\right), \tag{22}$$

where $i \in \{L, R\}$ denotes left or right child nodes and $\beta^k_{i,r}(\dot{\mathbf{v}}^k) = d(\dot{\mathbf{v}}^k, \mathbf{v}^k_{i,r})$ are the sampled branch lengths, defined as the hyperbolic distance $d(\cdot,\cdot)$ from Eq. 2.

Theoretical Justification. H-Csmc is an Smc algorithm on the extended space of all random variables generated by Algorithm 3. Therefore, it keeps the favorable properties of Csmc, such as unbiasedness of the normalization constant estimate and asymptotic consistency. The key property that ensures this for H-Csmc is *proper weighting* (Liu, 2008; Naesseth et al., 2015, 2019a).

Definition 1 (Proper Weighting)**.** We say that the random pair (s_r, w_r) are *properly weighted* for the unnormalized distribution $\frac{\pi(s_r)\nu^-(s_{r-1})}{\pi(s_{r-1})}$ if $w_r \geq 0$ almost surely, and for all measurable functions h,

$$\mathbb{E}[w_r h(s_r)] = \int h(s_r) \frac{\pi(s_r)\nu^-(s_{r-1})}{\pi(s_{r-1})}\, \mathrm{d}s_r. \tag{23}$$

Proposition 1 (Proper Weighting)**.** The particles s^k_r and weights w^k_r are properly weighted for $\frac{\pi(s_r)\nu^-(s_{r-1})}{\pi(s_{r-1})}$.

Proof.

$$\mathop{\mathbb{E}}_{q}[w^k_r h(s^k_r)] = \int q(s^k_r | s^{a^k_{r-1}}_{r-1})\, w^k_r\, h(s^k_r)\, ds^k_r$$
$$= \sum_{\tau} \int h(\dot{\mathbf{v}}^k_r, \tau^k_r) \frac{\pi(\dot{\mathbf{v}}^k_r, \tau^k_r)\nu^-(\dot{\mathbf{v}}^{a^k_{r-1}}_{r-1}, \tau^{a^k_{r-1}}_{r-1})}{\pi(\dot{\mathbf{v}}^{a^k_{r-1}}_{r-1}, \tau^{a^k_{r-1}}_{r-1})}$$
$$\times \det\left(\frac{\partial \beta^k_{i,r}(\dot{\mathbf{v}}^k)}{\partial \dot{\mathbf{v}}^k}\right) d\dot{\mathbf{v}}^k$$
$$= \sum_{\tau} \int h(\beta^k_{i,r}, \tau^k_r) \frac{\pi(\beta^k_{i,r}, \tau^k_r)\nu^-(\beta^{a^k_{r-1}}_{i,r-1}, \tau^{a^k_{r-1}}_{r-1})}{\pi(\beta^{a^k_{r-1}}_{i,r-1}, \tau^{a^k_{r-1}}_{r-1})}\, \mathrm{d}\beta^k_{i,r}$$
$$= \int h(s^k_r) \frac{\pi(s^k_r)\nu^-(s^{a^k_{r-1}}_{r-1})}{\pi(s^{a^k_{r-1}}_{r-1})}\, \mathrm{d}s^k_r$$

□

3.5 Hyperbolic Nested CSMC

We extend Ncsmc (Moretti et al., 2021; Yang et al., 2024) to hyperbolic space through marginalizing over one-step lookahead partial states. This extension, referred to as H-Ncsmc, is a hyperbolic adaptation of Ncsmc, retaining favorable properties such as unbiasedness and consistency. Pseudocode for H-Ncsmc is presented in Algorithm 2 and detailed in A.5.

Proposition 2. The particles s_r^k and weights w_r^k generated by Algorithm 2 are properly weighted for $\frac{\pi(s_r)\nu^-(s_{r-1})}{\pi(s_{r-1})}$.

Proof.

$$\begin{aligned}\mathbb{E}[w_r^k h(s_r^k)] &= \mathbb{E}\left[w_r^k \cdot h(s_r^{k,I}[J])\right]\\ &= \mathbb{E}\left[\sum_{j=1}^{L}\sum_{i=1}^{M} w_r^k \frac{w_r^{k,i}[j]}{\sum_l \sum_m w_r^{k,m}[l]} h(s_r^{k,i}[j])\right]\\ &= \frac{1}{ML}\sum_{j=1}^{L}\sum_{i=1}^{M}\mathbb{E}\left[w_r^{k,i}[j]\cdot h(s_r^{k,i}[j])\right]\\ &= \mathbb{E}\left[w_r^{k,i}[j]\cdot h(s_r^{k,i}[j])\right]\end{aligned}$$

$$= \sum_{\tau}\int h(\dot{\mathbf{v}}_r^{k,m}[j], \tau_r^k[j]) \frac{\pi(\dot{\mathbf{v}}_r^{k,m}[j], \tau_r^k)\cdot\nu^-(\dot{\mathbf{v}}_{r-1}^{a_{r-1}^k}, \tau_{r-1}^{a_{r-1}^k})}{\pi(\dot{\mathbf{v}}_{r-1}^{a_{r-1}^k}, \tau_{r-1}^{a_{r-1}^k})}$$

$$\cdot \det\left(\frac{\partial\beta_{i,r}^{k,m}(\dot{\mathbf{v}}^{k,m}[j])}{\partial\dot{\mathbf{v}}^{k,m}[j]}\right) d\dot{\mathbf{v}}_r^{k,m}[j]$$

Replacing index i for index m so that i can be used to denote left and right branch lengths β_i:

$$= \sum_{\tau}\int h(\beta_{i,r}^{k,m}[j], \tau_r^k[j]) \frac{\pi(\beta_{i,r}^{k,m}[j], \tau_r^k)\nu^-(\beta_{i,r-1}^{a_{r-1}^k}, \tau_{r-1}^{a_{r-1}^k})}{\pi(\beta_{i,r-1}^{a_{r-1}^k}, \tau_{r-1}^{a_{r-1}^k})}$$

$$\cdot\, \mathrm{d}\beta_{i,r}^{k,m}[j]$$

$$= \int h(s_r^k)\frac{\pi(s_r^k)\nu^-(s_{r-1}^{a_{r-1}^k})}{\pi(s_{r-1}^{a_{r-1}^k})}\,\mathrm{d}s_r^k$$

□

Algorithm 2 Nested Combinatorial Sequential Monte Carlo with Poincaré Embeddings

```
1:  Input: $\mathbf{Y} \in \Omega^{N\times M}$, $\theta = (\mathbf{Q}, \{\lambda_i\}_{i=1}^{|E|})$; embed $Y_n \to f(Y_n) \in \mathcal{P}$
2:  Initialize: $\forall k$, $s_0^k \leftarrow \perp$, $w_0^k \leftarrow 1/K$
3:  for $r = 0$ to $R = N-1$ do
4:      for $k = 1$ to $K$ do
5:          Resample: $\mathbb{P}(a_{r-1}^k = i) = w_{r-1}^i / \sum_{l=1}^K w_{r-1}^l$
6:          for $j = 1$ to $L = \binom{N-r}{2}$ do
7:              for $m = 1$ to $M$ do
8:                  Find closest point:
                        $\mathbf{v}_p^{k,m}[j] \leftarrow \underset{\mathbf{v}\in\gamma(\mathbf{v}_{L,r},\mathbf{v}_{R,r})}{\arg\min}\ d(\mu_0, \mathbf{v})$
9:                  Sample tangent: $\mathbf{v}^{k,m}[j] \sim \mathcal{N}(0, \Sigma)$
10:                 Transport:
                        $\tilde{\mathbf{v}}^{k,m}[j] \leftarrow (1 - \|\mathbf{v}_p^{k,m}[j]\|_2^2)\mathbf{v}^{k,m}[j]$
11:                 Map to disk: $\dot{\mathbf{v}}^{k,m}[j] \leftarrow \mathbf{v}_p^{k,m}[j] \oplus$
                        $\tanh\left((1 - \|\mathbf{v}_p^{k,m}[j]\|_2^2)\cdot\|\tilde{\mathbf{v}}^{k,m}[j]\|_2\right)\frac{\tilde{\mathbf{v}}^{k,m}[j]}{\|\tilde{\mathbf{v}}^{k,m}[j]\|_2}$
12:                 Branch lengths:
                        $\mathcal{B}_{L,r}^{k,m}[j] \leftarrow d(\mathbf{v}_{L,r}^{k,m}[j], \dot{\mathbf{v}}^{k,m}[j])$
                        $\mathcal{B}_{R,r}^{k,m}[j] \leftarrow d(\mathbf{v}_{R,r}^{k,m}[j], \dot{\mathbf{v}}^{k,m}[j])$
13:                 Compute potentials:
                        $w_r^{k,m}[j] := \frac{\pi(s_r^{k,m}[j])}{\pi(s_{r-1}^{a_{r-1}^k})} \cdot \frac{\nu^-(s_{r-1}^{a_{r-1}^k})}{q(s_r^{k,m}[j] | s_{r-1}^{a_{r-1}^k})}$
                        $\cdot \det\left(\frac{\partial\beta_{i,r}^{k,m}(\dot{\mathbf{v}}^{k,m}[j])}{\partial\dot{\mathbf{v}}^{k,m}[j]}\right)$
14:             end for
15:         end for
16:         Extend:
                $s_r^k = s_r^{k,i}[J]$ where $P(J = j, I = i) \propto w_r^{k,i}[j]$
17:         Update: $w_r^k = \frac{1}{ML}\sum_{l=1}^{L}\sum_{m=1}^{M} w_r^{k,i}[j]$
18:     end for
19: end for
20: Output: $s_R^{1:K}$, $w_{1:R}^{1:K}$
```

H-Ncsmc iterates through rank events (Step 3) and Monte Carlo samples (Step 4), performing a Resample step (Step 5). For each sample k, the algorithm enumerates all $\binom{N-r}{2}$ possible one-step-ahead topologies and samples M corresponding sub-branch lengths. For each of the $j = 1, \ldots, \binom{N-r}{2}$ topologies, the two coalescing embedded points $\mathbf{v}_{L,r}^k[j]$ and $\mathbf{v}_{R,r}^k[j]$ (the left and right child nodes) are used to identify the closest point on the geodesic between them to the origin (Step 10). The parent node is then sampled in the tangent plane at the origin (Step 11) and mapped via parallel transport to its mean (Step 12). The exponential map projects these samples back onto the manifold (Step 13). The left and right branch lengths $\beta_{L,r}^{k,m}[j]$, $\beta_{R,r}^{k,m}[j]$

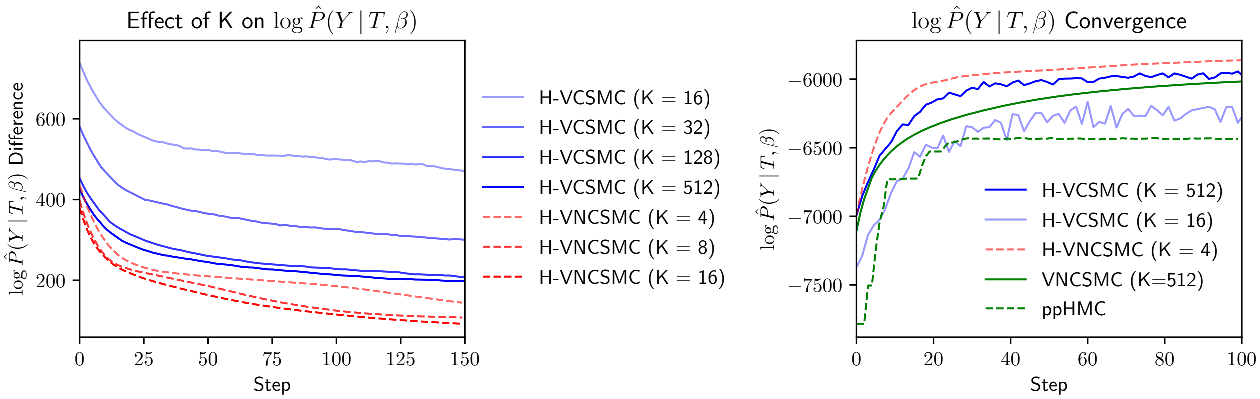

Figure 4: **Left:** As K increases, the difference in log conditional likelihood $\log \widehat{P}(Y|\tau, \beta)$ between both H-Vcsmc and H-Vncsmc and MrBayes approaches zero. MrBayes runs MCMC for 20,000 iterations. **Right:** Convergence of $\log \widehat{P}(Y|\tau, \beta)$ for H-Vcsmc, H-Vncsmc, Vcsmc, Vncsmc, and ppHmc. In particular, even at $K = 4$, H-Vncsmc consistently outperforms the other methods. The results are averaged over three random seeds.

are computed using the geodesic distance between the sampled parent and child nodes (Step 14). Each sampled look-ahead state $s_r^{k,m}[j]$ receives a sub-weight or potential function (Step 15). The partial state for s_r^k is extended by selecting one of the $\binom{N-r}{2}$ topologies and corresponding branch lengths based on their weights (Step 18). Finally, each sample's weight is computed by averaging over all sub-weights (Step 19).

We utilize the H-Csmc and H-Ncsmc algorithms to establish objective functions for Vi, resulting in the H-Vcsmc and H-Vncsmc methods. These algorithms are also employed to formulate variational objectives via a lower bound on the marginal likelihood estimate.

4 Experiments

Experiments were run on a Lambda Cloud Gpu instance equipped with a 30-core Amd Epyc 7J13 Cpu, plus an Nvidia A100 Gpu with 40 GB of VRAM. We assumed a uniform prior over topologies and a $\lambda_{bl} = 10$ exponential prior over branch lengths.

Primates Dataset. We applied H-Vcsmc and H-Vncsmc to the primate mitochondrial DNA dataset (Hayasaka et al., 1988), consisting of 12 taxa across 898 sites, yielding 13,749,310,575 distinct topologies. Our methods were compared to MrBayes, ppHmc along with reimplementations of Vcsmc and Vncsmc. In Fig. 4 (left), we find that as K increases, the difference in log conditional likelihood $\log \widehat{P}(Y|\tau, \beta)$ between both H-Vcsmc and H-Vncsmc and MrBayes approaches zero. MrBayes runs MCMC for 20,000 iterations. Fig. 4 (right) displays convergence of $\log \widehat{P}(Y|\tau, \beta)$ for H-Vcsmc, H-Vncsmc, Vcsmc, Vncsmc, and ppHmc. In particular, even at $K = 4$, H-Vncsmc consistently outperforms the other methods. Results are averaged over three random seeds.

Large Taxa Benchmarks. We evaluated H-Vcsmc on seven large phylogenetic benchmark datasets, ranging from 27 to 64 eukaryote species and spanning 378 to 2520 sites (Hedges et al., 1990; Garey et al., 1996; Yang & Yoder, 2003; Henk et al., 2003; Lakner et al., 2008; Zhang & Blackwell, 2001; Rossman et al., 2001). Table 1 compares log conditional likelihood values across several methods. H-Vcsmc and H-Vncsmc show competitive performance compared to existing approaches. Vbpi-Nf uses precomputed tree topology support and extensive bootstrap sampling. While MrBayes achieves the highest likelihood on most datasets, it requires $4.9\times - 14.4\times$ longer running times (see Table 2). Notably, H-Vcsmc demonstrates efficient convergence, running $40\times - 60\times$ faster than GeoPhy while achieving higher likelihood values on two datasets and closely matching its performance on another. Results use MrBayes Stepping Stone sampling with four chains for 10,000,000 iterations, sampling every 100 iterations, following VaiPhy's default settings.

Implementation Details. Algorithm 1 incurs minimal overhead, with complexity unchanged from the original methods except for embedding points in the Poincaré disk. The Ncsmc algorithm increases computational cost by marginalizing intermediate target densities to improve partial state exploration. While H-Csmc runs in $\mathcal{O}(KNM)$, H-Ncsmc requires $\mathcal{O}(KN^3M)$. The fast execution of H-Vcsmc is due to an efficient PyTorch implementation, fully vectorized across particles and sites except for resampling.

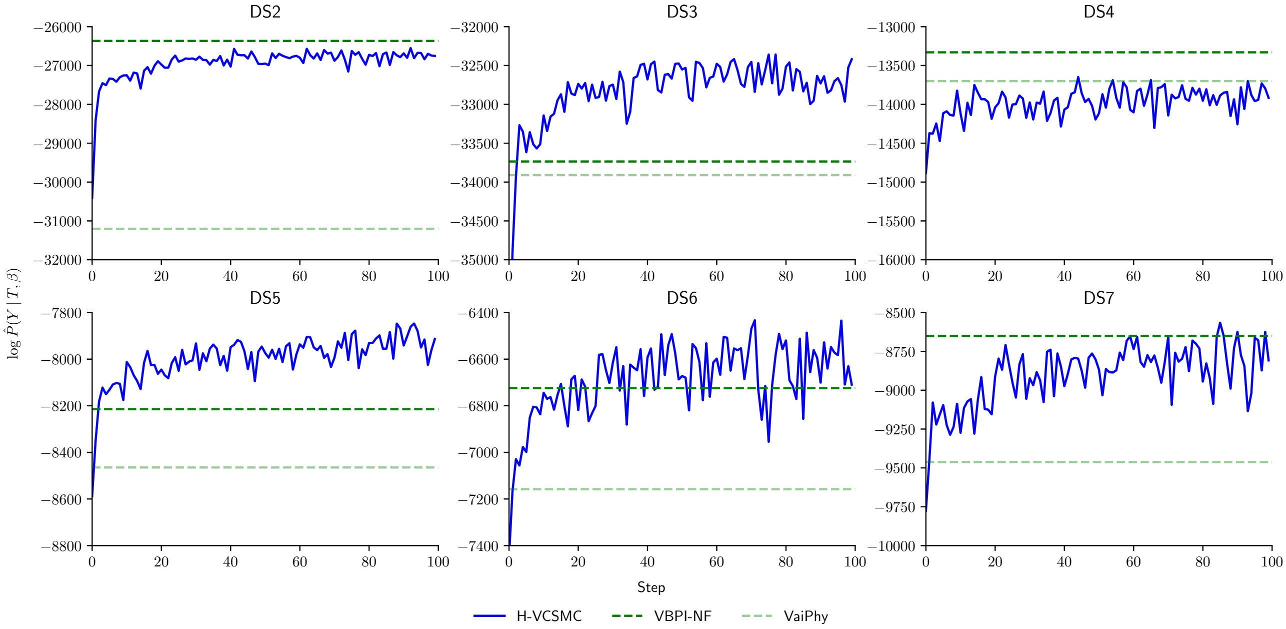

Figure 5: Convergence of H-VCSMC on 6 out of 7 benchmark datasets. H-VCSMC often reaches higher values than baseline methods prior to convergence, indicating greater efficiency.

Table 1: Comparison of marginal log likelihood estimates across 7 datasets using various inference techniques.

Dataset	H-VCSMC	H-VNCSMC	VBPI-NF	MrBayes	Vaiphy	VCSMC-PyTorch		GeoPhy	Dodonophy
						K=512	K=1024		
DS1	-7639.21	-7290.18	-7108.40	**-7032.45**	-7490.54	-7728.43	-7669.21	-7111.70	-8042.10
DS2	-26799.23	-26500.64	-26367.70	**-26363.85**	-31203.44	-27133.01	-27017.15	-26372.84	-26777.43
DS3	-32535.99	**-31818.26**	-33735.09	-33729.60	-33911.13	-33121.44	-32851.80	-33766.49	-34437.62
DS4	-13954.06	-13849.76	-13329.93	**-13292.37**	-13700.86	-14212.81	-14119.78	-13343.06	-15070.36
DS5	-8141.26	**-8012.42**	-8214.61	-8192.96	-8464.77	-8402.81	-8367.93	-8233.22	-13702.80
DS6	-6960.84	-6789.01	-6724.36	**-6571.02**	-7157.84	-6976.13	-6888.14	-6734.07	-9595.49
DS7	-9116.36	-8922.34	-8650.49	**-8438.78**	-9462.21	-9509.71	-9493.81	-8658.93	–

Table 2: Running Times in Decimal Hours for Various Datasets Across Different Methods.

Dataset	GeoPhy	VCSMC Original K=512	VCSMC Original K=1024	VCSMC K=512 (PyTorch)	VCSMC K=1024 (PyTorch)	H-VCSMC K=512	MrBayes
DS1	6.78	1.07	2.25	0.28	0.45	0.16	1.864
DS2	6.69	1.57	3.27	0.25	0.48	0.21	2.8275
DS3	8.08	1.85	3.77	0.23	0.44	0.21	3.0277
DS4	7.67	1.40	2.88	0.18	0.33	0.19	1.9722
DS5	8.67	0.55	1.17	0.09	0.16	0.15	0.9138
DS6	8.20	2.08	4.20	0.23	0.41	0.24	1.4638
DS7	8.88	2.63	5.28	0.26	0.65	0.31	1.5194

5 Conclusion

We have introduced hyperbolic extensions to the Combinatorial and Nested Combinatorial Sequential Monte Carlo algorithms, leading to the development of two hyperbolic VI methods. Both H-VCSMC and H-VNCSMC demonstrate strong empirical performance, significantly improving both scalability and effectiveness in high-dimensional inference. Our results show that these techniques offer a powerful and computationally efficient alternative for phylogenetic modeling tasks. An implementaiton of both H-VCSMC and H-VNCSMC along with a high-performance PyTorch reimplementation of VCSMC is available at https://github.com/axchen7/vcsmc.

Acknlowledgements We acknowledge funding from the Atlanta University Consortium Data Science Initiative.

References

Vu Dinh, Arman Bilge, Cheng Zhang, and Frederick A. Matsen, IV. Probabilistic path Hamiltonian Monte Carlo. volume 70 of *Proceedings of Machine Learning Research*, pp. 1009–1018, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

J Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 1981.

Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/dbab2adc8f9d078009ee3fa810bea142-Paper.pdf.

JR Garey, TJ Near, MR Nonnemacher, and SA Nadler. Molecular evidence for Acanthocephala as a subtaxon of Rotifera. *Journal of molecular evolution*, 43(3): 287–292, 1996.

Mikhail Gromov. *Hyperbolic groups*. Springer, 1987.

K Hayasaka, T Gojobori, and S Horai. Molecular phylogeny and evolution of primate mitochondrial DNA. *Molecular Biology and Evolution*, 5(6):626–644, 11 1988.

S Hedges, K Moberg, and LR Maxson. Tetrapod phylogeny inferred from 18S and 28S ribosomal RNA sequences and a review of the evidence for amniote relationships. *Molecular Biology and Evolution*, 7(6): 607–633, 11 1990.

Daniel A. Henk, Alex Weir, and Meredith Blackwell. Laboulbeniopsis termitarius, an ectoparasite of termites newly recognized as a member of the Laboulbeniomycetes. *Mycologia*, 95(4):561–564, 2003.

John P. Huelsenbeck and Fredrik Ronquist. MRBAYES: Bayesian inference of phylogenetic trees . *Bioinformatics*, 17(8):754–755, 08 2001.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes, 2013.

Hazal Koptagel, Oskar Kviman, Harald Melin, Negar Safinianaini, and Jens Lagergren. Vaiphy: a variational inference based algorithm for phylogeny. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 14758–14770. Curran Associates, Inc., 2022.

Clemens Lakner, Paul van der Mark, John P. Huelsenbeck, Bret Larget, and Fredrik Ronquist. Efficiency of Markov Chain Monte Carlo Tree Proposals in Bayesian Phylogenetics. *Systematic Biology*, 57(1): 86–103, 02 2008.

J.S. Liu. *Monte Carlo strategies in scientific computing*. Springer Verlag, New York, Berlin, Heidelberg, 2008. ISBN 0-387-95230-6.

Matthew Macaulay and Mathieu Fourment. Differentiable phylogenetics via hyperbolic embeddings with dodonaphy. *Bioinformatics Advances*, pp. vbae082, 2024.

Adam H Marblestone, Greg Wayne, and Konrad P Kording. Neural distance embeddings for biological sequences. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, pp. 2999–3007. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/59dfa2df.pdf.

Takahiro Mimori and Michiaki Hamada. Geophy: Differentiable phylogenetic inference via geometric gradients of tree topologies. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 36591–36619. Curran Associates, Inc., 2023.

Antonio Khalil Moretti, Liyi Zhang, Christian A. Naesseth, Hadiah Venner, David Blei, and Itsik Pe'er. variational combinatorial sequential monte carlo methods for bayesian phylogenetic inference. In Cassio de Campos and Marloes H. Maathuis (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 971–981. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/moretti21a.html.

C. A. Naesseth, F. Lindsten, and T. B. Schön. Nested sequential Monte Carlo methods. In *International Conference on Machine Learning (ICML)*, pp. 1292–1301, 2015.

C. A. Naesseth, F. Lindsten, and T. B. Schön. High-dimensional filtering using nested sequential Monte Carlo. *IEEE Transactions on Signal Processing*, 67 (16):4177–4188, 2019a.

C. A. Naesseth, F. Lindsten, and T. B. Schön. Elements of sequential Monte Carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392, 2019b.

Yoshihiro Nagano, Shoichiro Yamaguchi, Yasuhiro Fujita, and Masanori Koyama. A wrapped normal distribution on hyperbolic space for gradient-based learning. In Kamalika Chaudhuri and Ruslan

Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4693–4702. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/nagano19a.html.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, 2017.

Amy Y. Rossman, John M. McKemy, Rebecca A. Pardo-Schultheiss, and Hans-Josef Schroers. Molecular studies of the Bionectriaceae using large subunit rDNA sequences. *Mycologia*, 93(1):100–110, 2001.

Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *Graph Drawing*, Lecture Notes in Computer Science, pp. 355–366. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 3642258778.

Charles Semple and Mike Steel. Phylogenetics. 2003.

Robert R. Sokal and Charles D. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958.

Abraham A. Ungar. *A Gyrovector Space Approach to Hyperbolic Geometry*. Synthesis Lectures on Mathematics & Statistics. Springer, 2009.

Liangliang Wang, Alexandre Bouchard-Côté, and Arnaud Doucet. Bayesian phylogenetic inference using a combinatorial sequential Monte Carlo method. *Journal of the American Statistical Association*, 01 2015.

Shijia Wang and Liangliang Wang. Particle Gibbs sampling for Bayesian phylogenetic inference, 2020.

Hanming Yang, Antonio Khalil Moretti, Sebastian Macaluso, Philippe Chlenski, Christian A. Naesseth, and Itsik Pe'er. Variational pseudo marginal methods for jet reconstruction in particle physics. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=thqC1c8YZJ.

Ziheng Yang and Anne Yoder. Comparison of Likelihood and Bayesian Methods for Estimating Divergence Times Using Multiple Gene Loci and Calibration Points, with Application to a Radiation of Cute-Looking Mouse Lemur Species. *Systematic Biology*, 52(5):705–716, 10 2003.

Cheng Zhang. Improved variational Bayesian phylogenetic inference with normalizing flows, 2020.

Cheng Zhang. Learnable topological features for phylogenetic inference via graph neural networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hVVUY7p64WL.

Cheng Zhang and Frederick A Matsen IV. Variational Bayesian phylogenetic inference. In *International Conference on Learning Representations*, 2019.

Ning Zhang and Meredith Blackwell. Molecular phylogeny of dogwood anthracnose fungus (Discula destructiva) and the Diaporthales. *Mycologia*, 93(2): 355–365, 2001.

Checklist

1. For all models and algorithms presented, check if you include:
   - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
   - (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   - (b) Complete proofs of all theoretical results. [Yes]
   - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
   - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   - (a) Citations of the creator If your work uses existing assets. [Yes]
   - (b) The license information of the assets, if applicable. [Yes]
   - (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   - (d) Information about consent from data providers/curators. [Not Applicable]
   - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   - (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

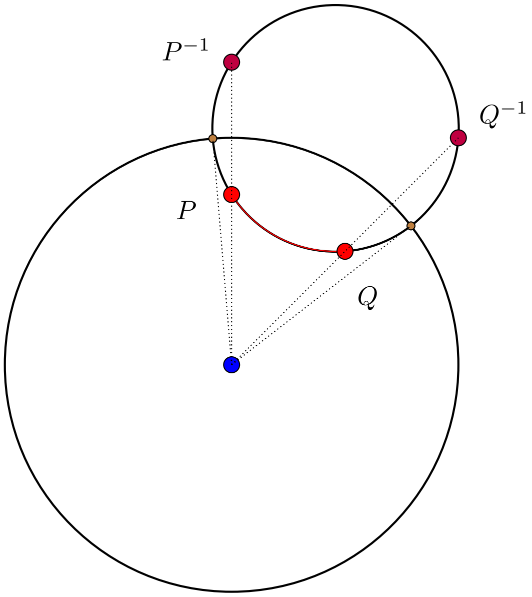

Figure 6: Geodesic arc in the Poincaré disk connecting points P and Q depicted in red, with their corresponding inverses P^{-1} and Q^{-1}. A circle orthogonal to the unit circle that passes through points P and Q must also intersect their inverses, P^{-1} and Q^{-1}, which are the reflections of P and Q with respect to the unit circle.

A Appendix

A.1 Poincaré Disk Model

The *Poincaré disk model* represents hyperbolic geometry within the unit disk in Euclidean space. This model is defined on the open unit disk, denoted as

$$\mathcal{P} \coloneqq \left\{\mathbf{z} \in \mathbb{R}^2 : \|\mathbf{z}\| < 1\right\} \equiv \left\{\mathbf{z} \in \mathbb{C} : |\mathbf{z}| < 1\right\}. \tag{24}$$

Let $\|\mathbf{z}\|$ denote the Euclidean norm of the point $\mathbf{z}$. The hyperbolic distance between two points $\mathbf{z}_1$ and $\mathbf{z}_2$ inside the unit disk is given by:

$$d(\mathbf{z}_1, \mathbf{z}_2) = \operatorname{arcosh}\left(1 + \frac{2\|\mathbf{z}_1 - \mathbf{z}_2\|^2}{(1 - \|\mathbf{z}_1\|^2)(1 - \|\mathbf{z}_2\|^2)}\right). \tag{25}$$

A.1.1 Metric Tensor

The *metric tensor* in the Poincaré disk is defined using a conformal modification of the Euclidean metric:

$$ds^2 = \frac{4(dx^2 + dy^2)}{(1 - x^2 - y^2)^2}, \tag{26}$$

where $dx^2 + dy^2$ is the standard Euclidean metric in $\mathbb{R}^2$ and $4/(1 - x^2 - y^2)^2$ is a conformal scaling factor dependent upon distance from the origin. As points approach the boundary of the disk ($x^2 + y^2 \to 1$), distances tend to infinity and the metric becomes increasingly large.

A.1.2 Geodesics in the Poincaré Disk

In the Poincaré disk model, geodesics are represented by arcs of circles that meet the boundary of the disk at right angles or by diameters of the disk, as shown in Fig. A.1.2. According to the Tangent-Secant Theorem, any circle orthogonal to the unit circle that passes through the points P and Q must also intersect their circular inverses, P^{-1} and Q^{-1}, which are reflections with respect to the unit circle. As a result, the hyperbolic line between $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ corresponds to an arc of the circle that passes through P, Q, and

$$P^{-1} = \left(\frac{x_1}{x_1^2 + y_1^2}, \frac{y_1}{x_1^2 + y_1^2}\right). \tag{27}$$

To plot the geodesic arc, find the center and radius of the circle passing through P, Q, and P^{-1}. This can be done by mapping each coordinate pair (x, y) to the complex plane as $z = x + iy$, and applying the linear transformation:

$$z \to \frac{z_3 - z_1}{z_2 - z_1}\,, \tag{28}$$

which maps the points (z_1, z_2, z_3) to new coordinates:

$$\left(0, 1, w = \frac{z_3 - z_1}{z_2 - z_1}\right)\,. \tag{29}$$

Imposing the condition that each transformed point satisfies the equation for a circle $|z - c| = r$, derive the system of equations:

$$|c|^2 = r^2 \tag{30}$$

$$1 - c - \bar{c} + |c|^2 = r^2 \tag{31}$$

$$|w|^2 - \bar{w}c - w\bar{c} + |c|^2 = r^2 \tag{32}$$

Solving this system yields the center of the transformed circle as:

$$c' = \frac{w - |w|^2}{w - \bar{w}}\,. \tag{33}$$

To revert this transformation back to the original coordinates, the center becomes:

$$c = (z_2 - z_1)\frac{w - |w|^2}{w - \bar{w}} + z_1\,. \tag{34}$$

Finally, the radius is determined by the relation:

$$r = |z_1 - c|\,. \tag{35}$$

A.1.3 Parallel Transport and Exponential Map

Hyperbolic space is not a vector space, which renders the linear operations typically employed in vector spaces inapplicable. Riemannian geometry requires special tools to understand how vectors change as they move across curved surfaces. *Parallel transport* adjusts vectors along a curve while preserving their relationship to the surface's curvature, ensuring consistent comparisons between different points. The *exponential map* projects directions from the tangent space to the manifold along geodesics, allowing us to translate local, linear information into motion across the curved space.

Parallel transport. Given two vectors $\mathbf{x}, \mathbf{y} \in \mathcal{P}$, the parallel transport from $\mathbf{x}$ to $\mathbf{y}$ is defined as a map $\mathrm{PT}_{\mathbf{x}\to\mathbf{y}}$ from the tangent plane at $\mathbf{x}$, $T_{\mathbf{x}}\mathcal{P}$, to the tangent plane at $\mathbf{y}$, $T_{\mathbf{y}}\mathcal{P}$. This map transports a vector in $T_{\mathbf{x}}\mathcal{P}$ along the geodesic from $\mathbf{x}$ to $\mathbf{y}$ in a parallel manner, preserving the metric tensor throughout the process. When $\mathbf{x}$ is the origin μ_0, Ganea et al. (2018) shows that parallel transport is defined as:

$$PT_{\mu_0 \to \mathbf{y}}(\mathbf{v}) := \frac{2}{\lambda_{\mathbf{y}}^K}\mathbf{v}\,, \tag{36}$$

where $\mathbf{v}$ denotes the vector being transported from tangent plane at the origin μ_0 to the tangent plane at point $\mathbf{y}$ in the Poincare disk and $\lambda_{\mathbf{y}}^K$ is the conformal factor defined below:

$$\lambda_{\mathbf{y}}^K := \frac{2}{1 + K||\mathbf{v}||_2^2}\,. \tag{37}$$

In the Poincaré disk, $K = -1$ so that the conformal factor simplifies to:

$$\lambda_{\mathbf{y}} = \frac{2}{1 - ||\mathbf{y}||_2^2}\,, \tag{38}$$

and parallel transport simplifies to:

$$PT_{\mu_0 \to \mathbf{y}}(\mathbf{v}) = (1 - ||\mathbf{y}||_2^2)\mathbf{v}\,. \tag{39}$$

Möbius addition. In hyperbolic space, Euclidean vector addition does not account for the curvature that affects distances and angles. Möbius addition is an operation for combining points taking into account the negative curvature of the manifold. Möbius addition $\oplus_K$ for $x, y \in \mathbb{M}_K$ (considering both signs of K) is defined as

$$\mathbf{x} \oplus_K \mathbf{y} := \frac{(1 - 2K\langle \mathbf{x}, \mathbf{y}\rangle - K||\mathbf{y}||_2^2)\mathbf{x} + (1 + K||\mathbf{x}||_2^2)\mathbf{y}}{1 - 2K\langle \mathbf{x}, \mathbf{y}\rangle + K^2||\mathbf{x}||_2^2||\mathbf{y}||_2^2} . \tag{40}$$

which simplifies when $K = -1$ to the following inside the Poincaré disk:

$$\mathbf{x} \oplus \mathbf{y} = \frac{(1 + 2\langle \mathbf{x}, \mathbf{y}\rangle + ||\mathbf{y}||_2^2)\mathbf{x} + (1 - ||\mathbf{x}||_2^2)\mathbf{y}}{1 + 2\langle \mathbf{x}, \mathbf{y}\rangle + ||\mathbf{x}||_2^2||\mathbf{y}||_2^2} . \tag{41}$$

For additional details see Ungar (2009).

Exponential and logarithmic map. The exponential map converts directions from the flat tangent plane into movement along the geodesics of the curved surface. The exponential map is defined as:

$$\exp_{\mathbf{x}}^K(\mathbf{v}) = \mathbf{x} \oplus_K \left(\tanh\left(\sqrt{-K}\frac{\lambda_x^K||\mathbf{v}||_2}{2}\right)\frac{\mathbf{v}}{\sqrt{-K}||\mathbf{v}||_2}\right) \tag{42}$$

which simplifies to:

$$\exp_x(\mathbf{v}) = x \oplus \left(\tanh\left(\frac{\lambda_x^{-1}||\mathbf{v}||_2}{2}\right)\frac{\mathbf{v}}{||\mathbf{v}||_2}\right) . \tag{43}$$

Recall that:

$$\lambda_{\mathbf{x}} = \frac{2}{1 - ||\mathbf{x}||_2^2} . \tag{44}$$

Therefore, we have:

$$\exp_x(\mathbf{v}) = \mathbf{x} \oplus \left(\tanh\left((1 - ||\mathbf{x}||_2^2)||\mathbf{v}||_2\right)\frac{\mathbf{v}}{||\mathbf{v}||_2}\right) . \tag{45}$$

The logarithmic map, which is the inverse of the exponential map, is defined in the Poincaré disk as follows:

$$\log_{\mathbf{x}}(\mathbf{y}) := \tanh^{-1}\left(\|\mathbf{y}\|_2\right)\frac{\mathbf{y}}{\|\mathbf{y}\|_2} . \tag{46}$$

A.1.4 Circles in the Poincaré Disk

In this section, we will derive a basic formula for converting between the hyperbolic center $\mathbf{c}$ and radius r of a circle and the corresponding Euclidean center $\mathbf{c}'$ and radius r'. This is useful for plotting and for the subsequent branch length derivation in Section A.3. For a visualization of the relationship between $\mathbf{c}, \mathbf{c}', r$, and r', see the left side of Figure 7.

Without loss of generality, let us assume that $\mathbf{c}$ lies on the x-axis. Throughout this derivation, we will use boldface (e.g. $\mathbf{c}$) to refer to points, and c to refer to just their x-coordinates.[1] Our goal is to find x_l and x_r such that $d(\mathbf{c}, \mathbf{x_l}) = d(\mathbf{c}, \mathbf{x_r}) = r$ and $x_l < c < x_r$, and use these diametrically-opposed points to compute a Euclidean center and radius.

We do so as follows:

$$d(\mathbf{c}, \mathbf{x_l}) = r = \text{arcosh}\left(1 + \frac{2(\|\mathbf{c} - \mathbf{x_l}\|^2)}{(1 - \|\mathbf{x_l}\|^2)(1 - \|\mathbf{c}\|^2}\right) \tag{47}$$

$$r = \text{arcosh}\left(1 + \frac{2(c - x_l)^2}{(1 - x_l^2)(1 - c^2)}\right) \tag{48}$$

$$\cosh(r) - 1 = \frac{2(c - x_l)^2}{(1 - x_l^2)(1 - c^2)} \tag{49}$$

[1] Our derivation is most simply expressed for points collinear on the line $y = 0$. To transform this to an arbitrary line, observe that the x-coordinates are also *norms* of vectors; thus, they can be used to multiply a unit vector pointing from the origin of the Poincaré disk to the (hyperbolic) center of our circle $\mathbf{c}$.

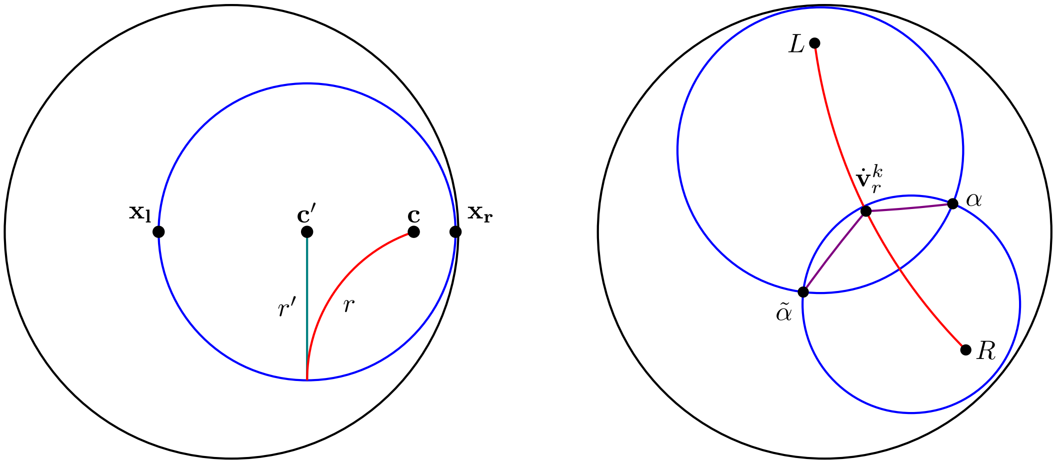

Figure 7: **Left:** Hyperbolic circles are also Euclidean circles (blue): the Euclidean distance from any point on the circle to $\mathbf{c}'$ is the Euclidean radius r' (teal), and the hyperbolic distance from any point on the circle to $\mathbf{c}$ is the hyperbolic radius r (red). **Right:** To visualize how branch lengths $(\beta^k_{R,r}, \beta^k_{L,r})$ are jointly distributed for coalescing nodes R and L, we plot circles (blue) with corresponding hyperbolic radii centered at each node. This shows that any pair of branch lengths corresponds to at most two points in the Poincaré disk. In the above, $\dot{\mathbf{v}}_r^k$ is the mean ancestor node, i.e. the point on the geodesic between L and R (red) closest to the origin. The likelihood of drawing either of the intersection points α or $\tilde{\alpha}$ is depends on their geodesic distances to $\dot{\mathbf{v}}_r^k$ (violet). In practice, since the branch lengths are computed by sampling from a $\mathcal{WN}$ centered at $\dot{\mathbf{v}}_r^k$, one of the intersections is typically much closer than the other.

Letting $u = \cosh(r) - 1$ and $v = 1 - c^2$:

$$\frac{2(c^2 + x_l^2 - 2cx_l)}{1 - x_l^2} = uv \tag{50}$$

$$2c^2 + 2x_l^2 - 4cx_l = uv - uvx_l^2 \tag{51}$$

$$\underbrace{(2 + uv)}_{A} x_l^2 + \underbrace{(-4c)}_{B} x_l + \underbrace{(2c^2 - uv)}_{C} = 0 \tag{52}$$

By the quadratic formula:

$$x_l = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \tag{53}$$

$$x_r = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \tag{54}$$

$$c' = \frac{x_l + x_r}{2} = \frac{-B}{2A} = \frac{2c}{2 + (\cosh(r) - 1)(1 - c^2)} \tag{55}$$

Thus, for any $c \in [-1, 1]$,

$$c' = \frac{c}{1 + \frac{1}{2}(\cosh(r) - 1)(1 - c^2)} \tag{56}$$

The Euclidean radius is simply:

$$r' = \frac{x_r - x_l}{2}. \tag{57}$$

A.2 Combinatorial Sequential Monte Carlo

We review several details of the Csmc algorithm. Additional details can be found in Wang et al. (2015).

A.2.1 Partial States and the Natural Forest Extension

Definition 2 (Partial State)**.** A partial state of rank $r \in 0, \cdots, N-1$, denoted by $s_r = (t_i, X_i)$, represents a collection of rooted trees and must satisfy three key conditions:

1. Partial states at different ranks do not overlap, meaning for any distinct ranks r and s, their corresponding sets of partial states are disjoint, i.e., $\forall r \neq s$, $\mathcal{S}_r \cap \mathcal{S}_s = \emptyset$.
2. The partial state set at the lowest rank consists of a single element, denoted as $S_0 = \perp$.
3. The partial states at the highest rank, $R = N-1$, correspond to the target space $\mathcal{X}$.

The likelihood and the measure π are specifically defined for the target space, with $\mathcal{S}_R = \mathcal{X}$. These definitions apply solely to the target space of trees and do not extend to the broader space of partial states $\mathcal{S}_{r<R}$, which is made up of forests with disjoint trees. Felsenstein's pruning algorithm (Felsenstein, 1981) is used to score the likelihood of a tree. However, partial states are explicitly collections of disjoint trees or leaf nodes. To extend the target measure π to include the partial state space $\mathcal{S}_{r<R}$, one method is to treat each disjoint tree in the forest independently, as described in Wang et al. (2015).

Definition 3 (Natural Forest Extension)**.** The natural forest extension (Nfe) expands the target measure π to include forests by taking the product over the trees that compose the forest:

$$\pi(s) \coloneqq \prod_{(t_i, X_i)} \pi_{Y_i(x_i)}(t_i), . \tag{58}$$

A key feature of the Nfe is its ability to incorporate information from non-coalescing elements into the local weight updates.

A.2.2 Encoding Leaf Nodes in the Poincare Disk

Definition 4 (δ-Hyperbolicity)**.** A geodesic metric space $\mathcal{X}$ is said to be *δ-hyperbolic* if there exists a constant $\delta \geq 0$ such that for any four points $\mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X}$, the following holds:

$$d(\mathbf{w},\mathbf{x}) + d(\mathbf{y},\mathbf{z}) \leq \max\{d(\mathbf{x},\mathbf{y}) + d(\mathbf{w},\mathbf{z})\,,\, d(\mathbf{x},\mathbf{z}) + d(\mathbf{w},\mathbf{y})\} + 2\delta \tag{59}$$

It is well established that trees are $0-$hyperbolic (Gromov, 1987), which aligns with the four-point condition used in Unweighted Pair Group Method with Arithmetic Mean (UPGMA) (Sokal & Michener, 1958). Intuitively, if the embedding is perfect, the distance between any two leaves in hyperbolic space should correspond to their Hamming distance. In cases of perfect ultrametricity, the points should approach the boundary of the Poincaré Disk. This alignment reflects the structure of the tree, where the distance from the origin can be interpreted as a molecular clock and represents the time until the divergence of organisms.

To further understand the implications of this structure, consider the four-point condition, which states that for any set of four leaves, there are three ways to group them into unordered pairs. The grouping that pairs closely related leaves should yield the minimal sum of pairwise distances, while the other two groupings—representing distances across subtrees—should be approximately equal. In hyperbolic spaces, this behavior is characterized by δ-hyperbolicity, which asserts that the deviation between the two maximal pairwise distances is bounded by some δ, and decreases as points move closer to the boundary. This relationship emphasizes how the molecular clock and the properties of hyperbolic space work in tandem to inform our understanding of evolutionary relationships.

A.3 Branch Length Probability Derivation

The mapping between embeddings and branch lengths is not injective. To see this, consider fixing two branch lengths, $\beta^k_{L,r}$ and $\beta^k_{R,r}$, and points for the left and right child nodes, L and R, on the disk. The points that are a geodesic distance $\beta^k_{L,r}$ from L form the circumference of a circle. Similarly, the points at a distance $\beta^k_{R,r}$ from R form the circumference of another circle. The intersection of these two circumferences can result in two possible locations for the parent point, as illustrated in the right side of Fig. 7. To take into account this two to one mapping, the density of branch lengths will be a sum of two terms, one for each intersection point (α and $\tilde{\alpha}$) on the disk:

$$q_\psi(\dot{\mathbf{v}}^k_r \mid \dot{\mathbf{v}}^{a^k_{r-1}}_{r-1}, \tau^{a^k_{r-1}}_{r-1}) = q_\psi(\dot{\mathbf{v}}^k_r \mid \dot{\mathbf{v}}^{a^k_{r-1}}_{r-1} = \alpha, \tau^{a^k_{r-1}}_{r-1}) + q_\psi(\dot{\mathbf{v}}^k_r \mid \dot{\mathbf{v}}^{a^k_{r-1}}_{r-1} = \tilde{\alpha}, \tau^{a^k_{r-1}}_{r-1}) \tag{60}$$

Due to concentration of measure, the further solution maps to a point where the $\mathcal{WN}$ density is exponentially smaller than the closer point.

A.3.1 Jacobian Derivation

Given the distance function between two points $\mathbf{z}_1$ and $\mathbf{z}_2$ in hyperbolic space:

$$d(\mathbf{z}_1, \mathbf{z}_2) = \operatorname{arcosh}\left(1 + \frac{2\|\mathbf{z}_1 - \mathbf{z}_2\|^2}{(1 - \|\mathbf{z}_1\|^2)(1 - \|\mathbf{z}_2\|^2)}\right) \tag{61}$$

The derivative of $\operatorname{arcosh}(x)$ is given by:

$$\frac{d}{dx}\operatorname{arcosh}(x) = \frac{1}{\sqrt{x^2 - 1}} \tag{62}$$

Let:

$$f(\mathbf{z}_1, \mathbf{z}_2) = 1 + \frac{2\|\mathbf{z}_1 - \mathbf{z}_2\|^2}{(1 - \|\mathbf{z}_1\|^2)(1 - \|\mathbf{z}_2\|^2)} \tag{63}$$

We can now apply the chain rule to compute the derivative of $d(\mathbf{z}_1, \mathbf{z}_2)$ with respect to $\mathbf{z}_1$:

$$\frac{\partial d(\mathbf{z}_1, \mathbf{z}_2)}{\partial \mathbf{z}_1} = \frac{1}{\sqrt{f(\mathbf{z}_1, \mathbf{z}_2)^2 - 1}} \cdot \frac{\partial f(\mathbf{z}_1, \mathbf{z}_2)}{\partial \mathbf{z}_1} \tag{64}$$

The derivative of $f(\mathbf{z}_1, \mathbf{z}_2)$ with respect to $\mathbf{z}_1$ is given by:

$$\frac{\partial f(\mathbf{z}_1, \mathbf{z}_2)}{\partial \mathbf{z}_1} = \frac{2\,(\mathbf{z}_1 - \mathbf{z}_2)}{(1 - \|\mathbf{z}_1\|^2)(1 - \|\mathbf{z}_2\|^2)} - \frac{4\|\mathbf{z}_1 - \mathbf{z}_2\|^2 \mathbf{z}_1}{(1 - \|\mathbf{z}_1\|^4)(1 - \|\mathbf{z}_2\|^2)} \tag{65}$$

The Jacobian matrix $J_{d(\mathbf{z}_1, \mathbf{z}_2)}$ is formed from the partial derivatives of $d(\mathbf{z}_1, \mathbf{z}_2)$ with respect to each component of $\mathbf{z}_1$. The determinant of this Jacobian matrix is given by:

$$\det(J_{d(\mathbf{z}_1, \mathbf{z}_2)}) = \det\left(\frac{\partial}{\partial \mathbf{z}_1}\left(\frac{1}{\sqrt{f(\mathbf{z}_1, \mathbf{z}_2)^2 - 1}} \cdot \frac{\partial f(\mathbf{z}_1, \mathbf{z}_2)}{\partial \mathbf{z}_1}\right)\right) \tag{66}$$

Since $\dot{\mathbf{v}}^{a^k_{r-1}}_{r-1}$ can take two possible values (α and $\tilde{\alpha}$), the Jacobian will capture how the change in $\dot{\mathbf{v}}^{a^k_{r-1}}_{r-1}$ (between α and $\tilde{\alpha}$) affects the total proposal expression. Thus, the Jacobian can be written as the sum of two partial derivatives, corresponding to each term in the expression:

$$\mathbf{J} = \frac{\partial \beta^k_{i,r}(\dot{\mathbf{v}}^k)}{\partial \dot{\mathbf{v}}^k} := \frac{\partial}{\partial \dot{\mathbf{v}}^{a^k_{r-1}}_{r-1}}\left(q_\psi(\dot{\mathbf{v}}^k_r \mid \dot{\mathbf{v}}^{a^k_{r-1}}_{r-1} = \alpha, \tau^{a^k_{r-1}}_{r-1})\right) + \frac{\partial}{\partial \dot{\mathbf{v}}^{a^k_{r-1}}_{r-1}}\left(q_\psi(\dot{\mathbf{v}}^k_r \mid \dot{\mathbf{v}}^{a^k_{r-1}}_{r-1} = \tilde{\alpha}, \tau^{a^k_{r-1}}_{r-1})\right) \tag{67}$$

In our implementation, we define the Jacobian via automatic differentiation in PyTorch.

A.4 Hyperbolic CSMC

H-Csmc is described in detail in Algorithm 3.

Algorithm 3 Combinatorial Sequential Monte Carlo with Poincaré Embeddings

```
**Input:** $\mathbf{Y} = \{Y_1, \cdots, Y_N\} \in \Omega^{N \times M}$, $\theta = (\mathbf{Q}, \{\lambda_i\}_{i=1}^{|E|})$
**Embed sequences in** $\mathcal{P}$**:** $\forall n, Y_n \to f(Y_n)$ where $f : \Omega \to \mathcal{P}$
**Initialization:** $\forall k$, $s_0^k \leftarrow \perp$, $w_0^k \leftarrow 1/K$
**for** $r = 0$ **to** $R = N - 1$ **do**
    **for** $k = 1$ **to** $K$ **do**
        Resample:
            $\mathbb{P}(a_{r-1}^k = i) = \frac{w_{r-1}^i}{\sum_{l=1}^K w_{r-1}^l}$
        Extend partial state:
            (a) Sample two embedded points $\mathbf{v}_{L,r}^k$, $\mathbf{v}_{R,r}^k$ to coalesce uniformly.
                $\mathbb{P}(\mathbf{v}_{L,r}^k = i, \mathbf{v}_{R,r}^k = j) = \frac{1}{\binom{N-r+1}{2}}$
            (b) Find the point on the geodesic arc between $\mathbf{v}_{L,r}$ and $\mathbf{v}_{R,r}$ closest to the origin:
                $\mathbf{v}_p \leftarrow \underset{\mathbf{v} \in \gamma(\mathbf{v}_{L,r}, \mathbf{v}_{R,r})}{\arg\min} \; d(\mu_0, \mathbf{v})$
            (c) Sample a Gaussian in $\mathbb{R}^2$ to yield a vector in the tangent plane at the origin, $T_{\mu_0}\mathcal{P}$:
                $\mathbf{v}^k \sim \mathcal{N}(0, \Sigma)$
            (d) Parallel transport the tangent vector from the origin, $\mu_0$, to the point $\mathbf{v}_p$:
                $\tilde{\mathbf{v}}^k \leftarrow PT_{\mu_0 \to \mathbf{v}_p}(\mathbf{v}^k) = (1 - \|\mathbf{v}_p\|_2^2)\mathbf{v}^k$
            (e) Apply the exponential map to the transported vector:
                $\dot{\mathbf{v}}^k \leftarrow \exp_{\mathbf{v}_p}(\tilde{\mathbf{v}}^k) = \mathbf{v}_p \oplus \left(\tanh\left((1 - \|\mathbf{v}_p\|_2^2)\|\tilde{\mathbf{v}}^k\|_2\right) \frac{\tilde{\mathbf{v}}^k}{\|\tilde{\mathbf{v}}^k\|_2}\right)$
            (f) Compute the left and right branch lengths via the geodesics:
                $\mathcal{B}_{L,r}^k \leftarrow d(\mathbf{v}_{L,r}, \dot{\mathbf{v}}^k)$, $\mathcal{B}_{R,r}^k \leftarrow d(\mathbf{v}_{R,r}, \dot{\mathbf{v}}^k)$
        Compute weights:
            $w_r^k = \frac{\pi(s_r^k)}{\pi(s_{r-1}^{a_{r-1}^k})} \cdot \frac{\nu^-(s_{r-1}^{a_{r-1}^k})}{q(s_r^k | s_{r-1}^{a_{r-1}^k})} \cdot \det\left(\frac{\partial d(\dot{\mathbf{v}}^k, \mathbf{v}_{i,r}^k)}{\partial \dot{\mathbf{v}}^k}\right)$
    **end for**
**end for**
**Output:** $s_R^{1:K}$ , $w_{1:R}^{1:K}$
```

The marginal likelihood estimator $\widehat{\mathcal{Z}}_{HCSMC}$ and the evidence lower bound (ELBO) $\mathcal{L}_{HCSMC}$ for Vi are defined below:

$$\widehat{\mathcal{Z}}_{HCSMC} \coloneqq \|\widehat{\pi}_R\| = \prod_{r=1}^R \left(\frac{1}{K}\sum_{k=1}^K w_r^k\right), \qquad \mathcal{L}_{HCSMC} \coloneqq \mathbb{E}\left[\log \widehat{\mathcal{Z}}_{HCSMC}\right] \tag{68}$$

A.5 Hyperbolic Nested Combinatorial Sequential Monte Carlo

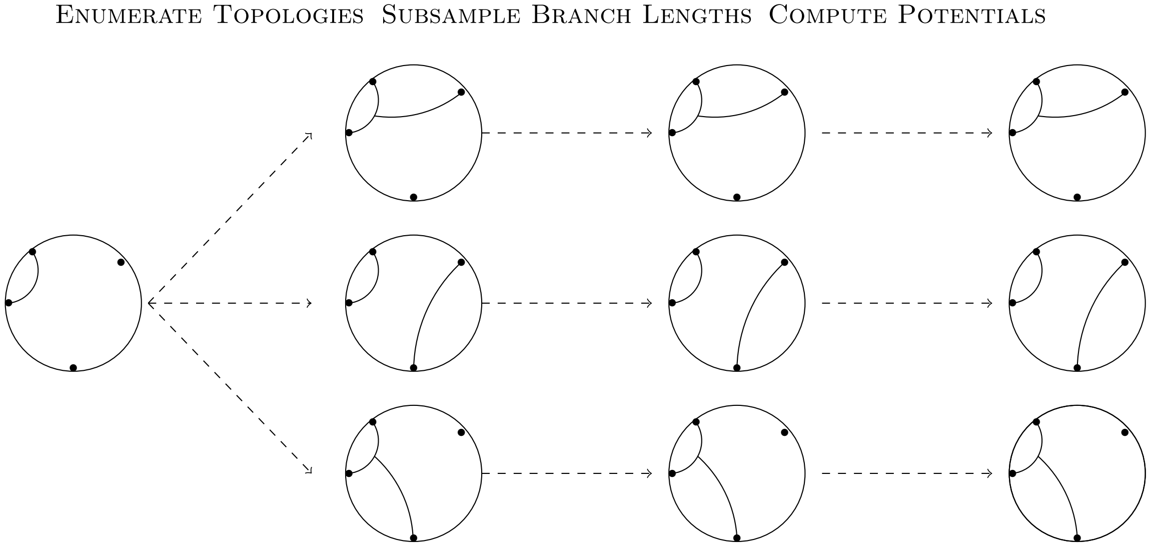

Figure 8: Overview of the Ncsmc framework in the Poincaré disk. To extend a partial state for a single sample, we enumerate all $\binom{N-r}{2}$ possible lookahead topologies, subsample branch lengths by drawing M points from a $\mathcal{WN}$ distribution, and compute sub-weights (potentials). One candidate is then sampled from these states based on its score. In the example shown, $N = 4$ leaf nodes are embedded, $r = 1$, yielding three topologies, and $M = 1$ Monte Carlo sample is used for branch lengths.

We present an overview of the Hyperbolic Nested Sequential Monte Carlo (H-Ncsmc) framework, described in detail in Algorithm 4.

H-Ncsmc iterates through rank events (Step 4) and Monte Carlo samples (Step 5), performing a Resample step. For each sample, H-Ncsmc enumerates all $\binom{N-r}{2}$ possible one-step-ahead topologies, and samples corresponding M sub-branch lengths using Algorithm 1 (Sampling Parent Node Embeddings in Hyperbolic Space). For each of the $j = 1, \ldots, \binom{N-r}{2}$ one-step-ahead topologies, corresponding to Monte Carlo sample k, the two coalescing embedded points $\mathbf{v}_{L,r}^{k}[j]$ and $\mathbf{v}_{R,r}^{k}[j]$ represent the left and right child nodes. The closest point on the geodesic between $\mathbf{v}_{L,r}^{k}[j]$ and $\mathbf{v}_{R,r}^{k}[j]$ to the origin is identified (Step 10). The parent node is then sampled in the tangent plane at the origin (Step 11) and mapped via parallel transport to its mean (Step 12). Afterward, the exponential map is used to project the samples back onto the manifold (Step 13). The left and right branch lengths $\beta_{L,r}^{k,m}[j]$, $\beta_{R,r}^{k,m}[j]$ are computed using the geodesic distance between the sampled parent and the child nodes (Step 14). Each sampled look-ahead state $s_r^{k,m}[j]$ is assigned a sub-weight or potential function (Step 15). The partial state for s_r^k is then extended by choosing one of the $\binom{N-r}{2}$ topologies and corresponding branch lengths based on their weights (Step 18). For each sample, we compute its weight by averaging over all sub-weights (Step 19).

In a slight abuse of notation, $i \in \{L, R\}$ refers to the left/right child nodes in Step 15, while $i \in \{1, \cdots, M\}$ refers to the sub-sampled parent embeddings in Steps 18 and 19. Although Algorithm 4 is explicitly outlined, it is efficiently implemented in a fully vectorized form across K and M. The marginal likelihood estimator $\widehat{\mathcal{Z}}_{H-NCSMC}$ and the evidence lower bound (ELBO) $\mathcal{L}_{H-NCSMC}$ for Vi are defined below:

$$\widehat{\mathcal{Z}}_{H-NCSMC} \coloneqq \|\widehat{\pi}_R\| = \prod_{r=1}^{R} \left(\frac{1}{K} \sum_{k=1}^{K} w_r^k \right), \qquad \mathcal{L}_{H-NCSMC} \coloneqq \mathbb{E}\left[\log \widehat{\mathcal{Z}}_{H-NCSMC}\right] \tag{69}$$

Algorithm 4 Nested Combinatorial Sequential Monte Carlo with Poincaré Embeddings

```
Input: $\mathbf{Y} = \{Y_1, \cdots, Y_N\} \in \Omega^{N \times M}$, $\theta = (\mathbf{Q}, \{\lambda_i\}_{i=1}^{|E|})$
Embed sequences in $\mathcal{P}$: $\forall n, Y_n \to f(Y_n)$ where $f : \Omega \to \mathcal{P}$
Initialization: $\forall k,\ s_0^k \leftarrow \perp,\ w_0^k \leftarrow 1/K$
for $r = 0$ to $R = N - 1$ do
    for $k = 1$ to $K$ do
        RESAMPLE:
            $$\mathbb{P}(a_{r-1}^k = i) = \frac{w_{r-1}^i}{\sum_{l=1}^K w_{r-1}^l}$$
        for $j = 1$ to $L = \binom{N-r}{2}$ do
            for $m = 1$ to $M$ do
                FORM LOOK-AHEAD PARTIAL STATE VIA ALGORITHM 1:
                    Find the point on the geodesic arc between $\mathbf{v}_{L,r}^{k,m}[j]$ and $\mathbf{v}_{R,r}^{k,m}[j]$ closest to the origin:
                        $$\mathbf{v}_p^{k,m}[j] \leftarrow \underset{\mathbf{v} \in \gamma(\mathbf{v}_{L,r}[j], \mathbf{v}_{R,r}[j])}{\arg\min} d(\mu_0, \mathbf{v})$$
                    Sample a Gaussian in $\mathbb{R}^2$ to yield a vector in the tangent plane at the origin, $T_{\mu_0}\mathcal{P}$:
                        $$\mathbf{v}^{k,m}[j] \sim \mathcal{N}(0, \Sigma)$$
                    Parallel transport the tangent vector from the origin, $\mu_0$, to the point $\mathbf{v}_p$:
                        $$\tilde{\mathbf{v}}^{k,m}[j] \leftarrow PT_{\mu_0 \to \mathbf{v}_p[j]}(\mathbf{v}^{k,m}[j]) = (1 - \|\mathbf{v}_p^{k,m}[j]\|_2^2)\mathbf{v}^{k,m}[j]$$
                    Apply the exponential map to the transported vector:
                        $$\dot{\mathbf{v}}^{k,m}[j] \leftarrow \exp_{\mathbf{v}_p[j]}(\tilde{\mathbf{v}}^{k,m}[j]) = \mathbf{v}_p^{k,m}[j] \oplus \left(\tanh\left((1 - \|\mathbf{v}_p^{k,m}[j]\|_2^2)\|\tilde{\mathbf{v}}^{k,m}[j]\|_2\right) \frac{\tilde{\mathbf{v}}^{k,m}[j]}{\|\tilde{\mathbf{v}}^{k,m}[j]\|_2}\right)$$
                    Compute the left and right branch lengths via the geodesics:
                        $$\mathcal{B}_{L,r}^{k,m}[j] \leftarrow d(\mathbf{v}_{L,r}^{k,m}[j], \dot{\mathbf{v}}^{k,m}[j]),\ \mathcal{B}_{R,r}^{k,m}[j] \leftarrow d(\mathbf{v}_{R,r}^{k,m}[j], \dot{\mathbf{v}}^{k,m}[j])$$
                COMPUTE POTENTIALS (SUB-WEIGHTS)
                    $$w_r^{k,m}[j] \coloneqq \frac{\pi(s_r^{k,m}[j])}{\pi(s_{r-1}^{a_{r-1}^k})} \cdot \frac{\nu^-(s_{r-1}^{a_{r-1}^k})}{q(s_r^{k,m}[j] | s_{r-1}^{a_{r-1}^k})} \cdot \det\left(\frac{\partial \beta_{i,r}^{k,m}(\dot{\mathbf{v}}^{k,m}[j])}{\partial \dot{\mathbf{v}}^{k,m}[j]}\right),$$
            end for
        end for
        EXTEND PARTIAL STATE
            $$s_r^k = s_r^{k,i}[J]$$
            $$P(J = j, I = i) = \frac{w_r^{k,i}[j]}{\sum_{l=1}^L \sum_{m=1}^M w_r^{k,m}[j]}$$
        COMPUTE WEIGHTS:
            $$w_r^k = \frac{1}{ML} \sum_{l=1}^L \sum_{m=1}^M w_r^{k,i}[j]$$
    end for
end for
Output: $s_R^{1:K}$ , $w_{1:R}^{1:K}$
```

Proposition 3. The particles s_r^k and weights w_r^k generated by Algorithm 4 are properly weighted for $\frac{\pi(s_r)\nu^-(s_{r-1})}{\pi(s_{r-1})}$.

Proof.

$$\begin{aligned}
\mathbb{E}[w_r^k h(s_r^k)] &= \mathbb{E}\left[w_r^k \cdot h(s_r^{k,I}[J])\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{L}\sum_{i=1}^{M} w_r^k \frac{w_r^{k,i}[j]}{\sum_l \sum_m w_r^{k,m}[l]} h(s_r^{k,i}[j])\right] \\
&= \frac{1}{ML}\sum_{j=1}^{L}\sum_{i=1}^{M} \mathbb{E}\left[w_r^{k,i}[j] \cdot h(s_r^{k,i}[j])\right] \\
&= \mathbb{E}\left[w_r^{k,i}[j] \cdot h(s_r^{k,i}[j])\right]
\end{aligned}$$

Replacing index i for index m so that i can be used to denote left and right branch lengths β_i:

$$\begin{aligned}
&= \sum_{\tau} \int h(\dot{\mathbf{v}}_r^{k,m}[j], \tau_r^k[j]) \frac{\pi(\dot{\mathbf{v}}_r^{k,m}[j], \tau_r^k) \cdot \nu^-(\dot{\mathbf{v}}_{r-1}^{a_{r-1}^k}, \tau_{r-1}^{a_{r-1}^k})}{\pi(\dot{\mathbf{v}}_{r-1}^{a_{r-1}^k}, \tau_{r-1}^{a_{r-1}^k})} \cdot \det\left(\frac{\partial \beta_{i,r}^{k,m}(\dot{\mathbf{v}}^{k,m}[j])}{\partial \dot{\mathbf{v}}^{k,m}[j]}\right) d\dot{\mathbf{v}}_r^{k,m}[j] \\
&= \sum_{\tau} \int h(\beta_{i,r}^{k,m}[j], \tau_r^k[j]) \frac{\pi(\beta_{i,r}^{k,m}[j], \tau_r^k)\nu^-(\beta_{i,r-1}^{a_{r-1}^k}, \tau_{r-1}^{a_{r-1}^k})}{\pi(\beta_{i,r-1}^{a_{r-1}^k}, \tau_{r-1}^{a_{r-1}^k})} \, \mathrm{d}\beta_{i,r}^{k,m}[j] \\
&= \int h(s_r^k) \frac{\pi(s_r^k)\nu^-(s_{r-1}^{a_{r-1}^k})}{\pi(s_{r-1}^{a_{r-1}^k})} \, \mathrm{d}s_r^k
\end{aligned}$$

□

A.6 Implementation Details

The fast execution time of H-Vcsmc is a result of an efficient PyTorch implementation of the method. With the exception of the resampling step, are fully vectorized across particles and sites. The high degree of parallelization allows the method to fully saturate high-end Gpus like the Nvidia A100.

Maximizing Gpu throughput also requires the method to work under `float32` precision. The primary numeric challenge is computing likelihoods via Felsenstein's pruning algorithm. Namely, each internal node is associated with a distribution $\mathbf{L} = (\exp(\mathbf{Q}\beta_{\mathrm{L}}))\mathbf{L}_{\mathrm{L}} \cdot (\exp(\mathbf{Q}\beta_{\mathrm{R}}))\mathbf{L}_{\mathrm{R}}$ over the alphabet, computed via the distributions $(\mathbf{L}_{\mathrm{L}}, \mathbf{L}_{\mathrm{R}})$ and branch lengths $(\beta_{\mathrm{L}}, \beta_{\mathrm{R}})$ of the left and right child. As a result of the recursive computation, $\mathbf{L}$ at the root node contains very small entries, leading to numeric instability in the forward and backward passes of the model. Performing these calculations in log space alleviates the numeric issues under `float32` precision.

We provide a modular PyTorch implementation of both H-Vcsmc and H-Vncsmc for comparative evaluation. The framework allows for various proposal distributions to be utilized. Additionally, we re-implemented Vcsmc and Vncsmc defining their proposal distributions and utilizing the same training code.

A.6.1 Decoding Embeddings into Elements of the Rate Matrix

An $A \times A$ rate matrix $\mathbf{Q}$ can be factored into A holding times h_i and A stationary probabilities s_i. A multi-layer perceptron maps embeddings to these $2A$ parameters. The softmax is used to obtain a valid stationary distribution, and the holding times are normalized so their mean is one. This normalization prevents the rate matrix from exploding and influencing the evolutionary time scale.

The embeddings are first feature-expanded so the rate matrix can depend more directly on an embedding's distance from the origin. In particular, embeddings are expanded into $D + 1$ dimensions via

$$\phi(p) = \left(\|p\|, \frac{p_1}{\|p\|}, \frac{p_2}{\|p\|}, \dots\right). \tag{70}$$

Constructing the full rate matrix involves first computing

$$\mathbf{Q}' = \begin{bmatrix} 1/h_1 & \cdots & 1/h_1 \\ \vdots & \ddots & \vdots \\ 1/h_A & \cdots & 1/h_A \end{bmatrix} \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_A \end{bmatrix}, \tag{71}$$

then taking $\mathbf{Q}$ to be $\mathbf{Q}'$, but with diagonal entries equal to the negative sum of each row's off-diagonal entries. By factorizing $\mathbf{Q}$, the stationary probabilities required by Felsenstein's pruning algorithm are directly available.

A.6.2 Deferring Branch Length Sampling

A large share of the variance in the gradient estimate can be attributed to the randomness in sampling branch lengths in the proposal step (i.e., sampling the location of the parent nodes). This variance has an adverse effect on the convergence speed of H-Vcsmc and H-Vncsmc. A convenient feature of these two methods, however, is that they can fit model parameters governing the proposal distribution and rate matrix without sampling branch lengths. One can substitute in a modified proposal that takes the parent node embedding directly to be the point on the geodesic closest to the origin. Embeddings can be optimized first under this deterministic regime, then "fine-tuned" after switching back to the proper proposal that includes parent node sampling. Our experiments show that for larger datasets, performing these two training phases separately leads to better likelihood estimates.

A.6.3 Memoizing Over Topologies

Under the modified proposal that produces deterministic branch lengths, two forests s_r^i and s_r^j with matching topologies will have matching likelihoods $\pi(s_r^i) = \pi(s_r^j)$. Therefore, the computation of the K important weights via Eq. (13) can be memoized for partial states with identical topologies, where the memoization key comes from a tree hashing algorithm that is invariant to child ordering. This memoization speeds up the forward pass and significantly reduces the size of the computation graph that must be saved in memory for computing gradients of Eq. (14).

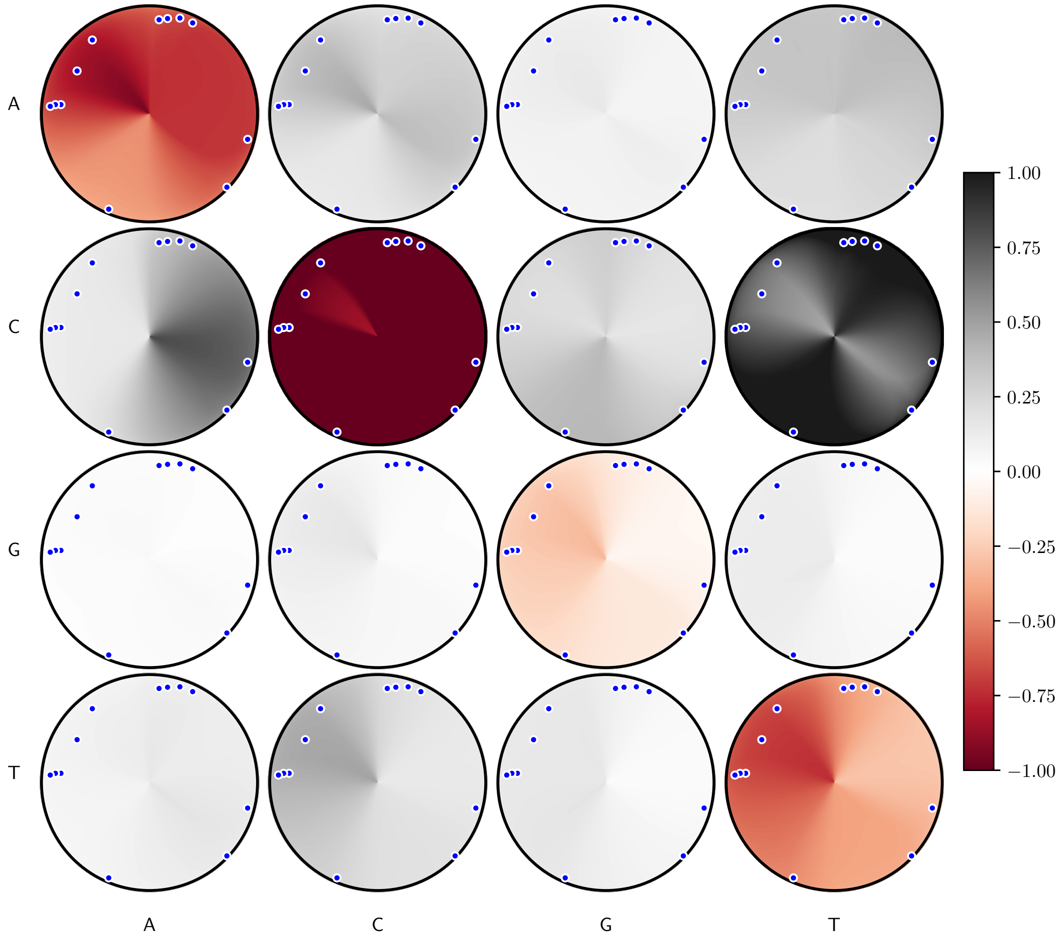

Figure 9: Visualization of the Q matrix entries as a function of position in the Poincaré disk. The matrix entry in row $i \in \{A, G, C, T\}$, column $j \in \{A, G, C, T\}$ is constructed from a holding time and stationary probability: $Q_{ij} = (1/h_i)s_j$. The diagonals are negative such that the rows sum to zero. Embedded sequences from the Primates dataset are shown as blue points. The Q matrix varies smoothly as a function of radius and polar angle, suggesting that the learned function does not overfit.

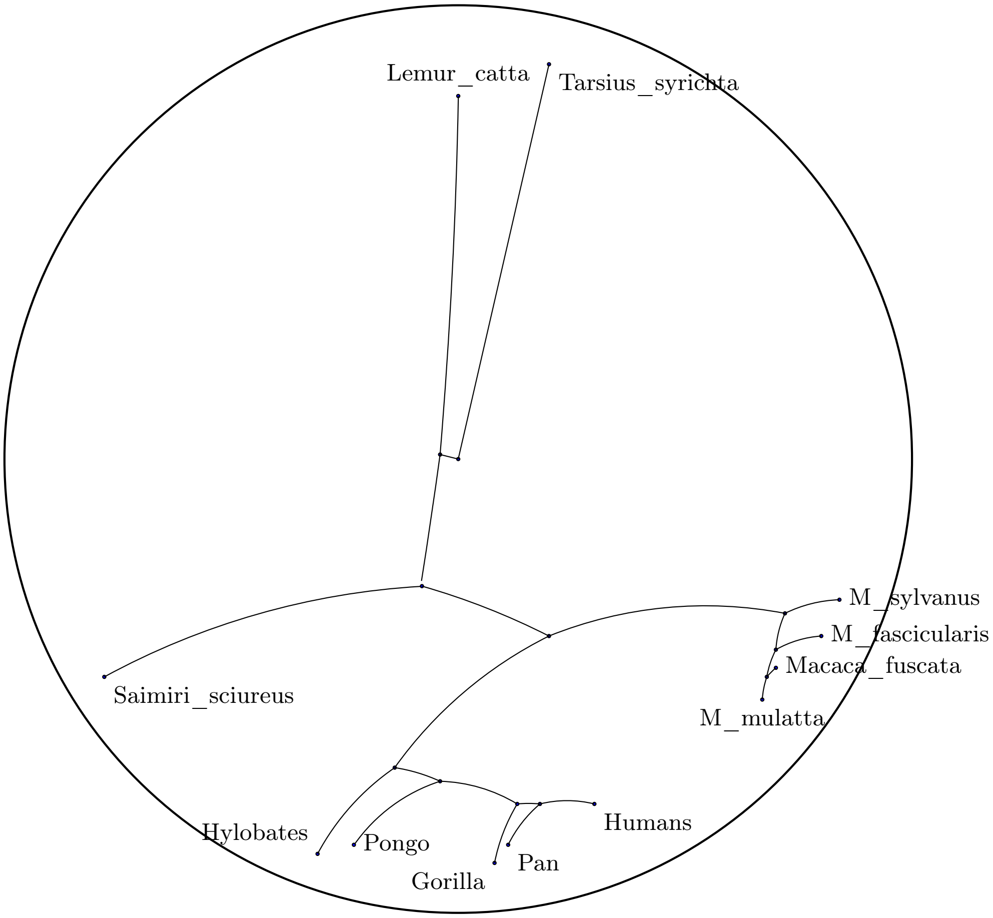

Figure 10: The maximum likelihood phylogeny inferred by H-VCSMC on the Primates dataset, with a topology consistent with that recovered by MrBayes. Monkeys, hominids, and prosimians are partitioned into distinct clades, each occupying separate regions of the Poincaré disk.