Randomized HyperSteiner: A Stochastic Delaunay Triangulation Heuristic for the Hyperbolic Steiner Minimal Tree

Aniss Aiman Medbouhi[*1] **Alejandro García-Castellanos**[*2] **Giovanni Luca Marchetti**[3]
Daniël M. Pelt[4] **Erik J. Bekkers**[2] **Danica Kragic**[1]
[1]School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology
[2]Amsterdam Machine Learning Lab (AMLab), University of Amsterdam
[3]School of Engineering Sciences, KTH Royal Institute of Technology
[4]Leiden Institute of Advanced Computer Science, Universiteit Leiden

Abstract

We study the problem of constructing Steiner Minimal Trees (SMTs) in hyperbolic space. Exact SMT computation is NP-hard, and existing hyperbolic heuristics such as HyperSteiner are deterministic and often get trapped in locally suboptimal configurations. We introduce **Randomized HyperSteiner** (RHS), a stochastic Delaunay triangulation heuristic that incorporates randomness into the expansion process and refines candidate trees via Riemannian gradient descent optimization. Experiments on synthetic data sets and a real-world single-cell transcriptomic data show that RHS outperforms Minimum Spanning Tree (MST), Neighbour Joining, and vanilla HyperSteiner (HS). In near-boundary configurations, RHS can achieve a 32% reduction in total length over HS, demonstrating its effectiveness and robustness in diverse data regimes.

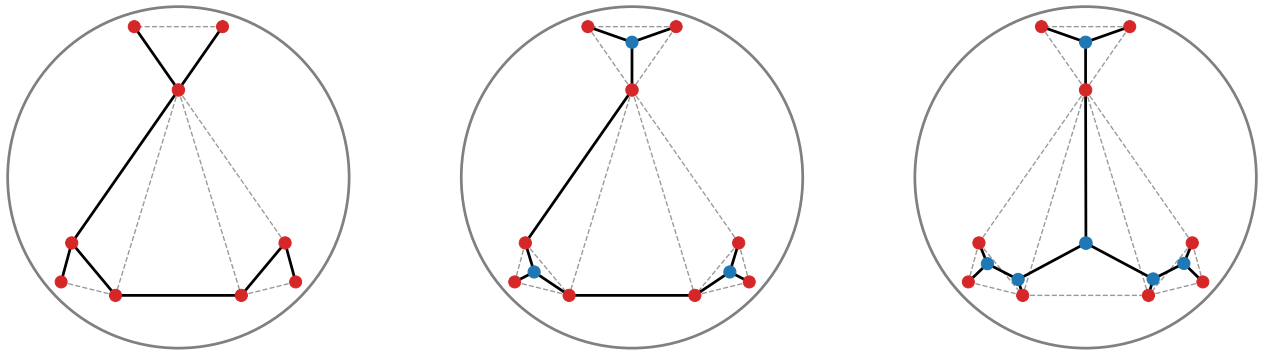
1 INTRODUCTION

The principle of *maximum parsimony*, in line with Occam's razor, states that among competing explanations consistent with the evidence, the simplest should be preferred. In phylogenetics, for example, this principle manifests through tree-based models, where minimality of length provides a natural notion of parsimony (Fitch, 1971). Steiner Minimal Trees (SMTs) formalize this objective by seeking the shortest tree connecting a set of points, potentially augmented with additional *Steiner points*. In contrast to the Minimum Spanning Tree (MST), which only uses the input terminals, SMTs may reduce total length substantially. This makes SMTs a particularly compelling model when inferring hierarchical structures from data, where the ground truth tree is typically unknown, while tree length is an interpretable and measurable quantity that allows systematic comparison across methods. From a broader perspective, our main contribution lies in non-Euclidean computational geometry, with biological hierarchical inference serving only as one potential application of the proposed framework.

Hyperbolic geometry offers a suitable ambient space for tree inference, as it can embed hierarchical and exponentially growing structures with arbitrary low distortion compared to the Euclidean space (Sarkar, 2011). This has motivated the development of machine learning models grounded in hyperbolic geometry, with applications in domains such as language (Tifrea et al., 2018; Nickel and Kiela, 2017), biology (Klimovskaia et al., 2020; Zhou and Sharpee, 2021), games (Cetin et al., 2022), robotics (Jaquier et al., 2024), and neurosciences (Zhou et al., 2018; Taleb et al., 2025). However, while hyperbolic embeddings effectively capture hierarchy, they do not provide direct algorithms for reconstructing explicit shortest trees embedded within hyperbolic space. Moreover, the SMT problem is NP-hard (Garey et al., 1977), making exact solutions infeasible in hyperbolic settings and necessitating heuristic approaches. A first step in this direction was the recently proposed HyperSteiner algorithm (García-Castellanos et al., 2025), which adapts a classical Delaunay-based heuristic to

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

[*]Equal contribution. E-mails: <medbouhi@kth.se>, <a.garciacastellanos@uva.nl>.



Minimum Spanning Tree | Vanilla HyperSteiner | Randomized HyperSteiner (ours)

Figure 1: Comparison of MST and heuristic SMTs using deterministic versus stochastic sampling over Delaunay triangles. Red (●) denotes terminals, blue (●) denotes Steiner points, and dashed lines correspond to the auxiliary hyperbolic DT. The deterministic (vanilla) HyperSteiner exhibits myopic behavior, whereas our randomized variant achieves superior global topology and reduces total tree length.

hyperbolic geometry, achieving an efficient construction of approximate SMTs. Yet, like other deterministic methods, the vanilla HyperSteiner inherits a major limitation: its MST-based heuristic tends to become trapped in locally optimal but globally suboptimal tree topologies.

In Euclidean space, this limitation has motivated stochastic variants of Delaunay triangulation (DT) heuristics, (Beasley and Goffinet, 1994; Laarhoven and Ohlmann, 2011), which showed that randomized Delaunay triangles selection can substantially improve solution quality by exploring a broader range of candidate topologies. Extending these ideas to hyperbolic space and combining them with novel theoretical guarantees enabling the use of Riemannian gradient descent (Bonnabel, 2013) for global optimization, we revisit HyperSteiner from a stochastic perspective and propose the first randomized SMT heuristic tailored to hyperbolic geometry. As illustrated in Figure 1, while the vanilla HyperSteiner (center) improves upon the MST baseline (left) through local Steiner point insertion, it becomes *myopic* due to its deterministic MST-based strategy, preventing exploration of alternative tree structures that could yield significantly shorter total lengths (right). Our work addresses this challenge by introducing stochasticity, thereby broadening topological exploration and reducing the risk of suboptimal solutions, yielding shorter tree configurations. Our contributions are as follows:

- We propose **Randomized HyperSteiner** (RHS), a stochastic Delaunay triangulation heuristic for constructing Steiner Minimal Trees in hyperbolic space.
- We extend the Euclidean uniqueness theorem to hyperbolic space, proving that Steiner trees have a unique minimizing configuration for fixed topology and terminals.
- We empirically demonstrate that RHS consistently outperforms Minimum Spanning Tree, Neighbour Joining, and original HyperSteiner across synthetic and real datasets.
- We show that RHS achieves reductions over MST values up to 43%, approaching the theoretical upper bound of 50% in near-boundary settings.

2 RELATED WORK

We review Delaunay triangulation heuristics for Euclidean Steiner Minimal Trees, non-Euclidean generalizations, and hierarchical inference methods. We add a review on hyperbolic machine learning in the Appendix (Section F).

Euclidean Delaunay-Based Heuristic SMTs. Heuristic methods for Euclidean Steiner Minimal Trees (SMTs) commonly exploit geometric structures, particularly Delaunay triangulations (DT), to identify promising local configurations for optimization (Zachariasen and Winter, 1999). These algorithms typically follow a three-phase structure: *Expansion* selects triangles from the DT as candidates for local 3-point Steiner problems (i.e., Fermat points); *Heuristic Construction* assembles local solutions into a global tree; and *Exploration Policy* determines refinement strategies. The seminal Smith-Lee-Liebman (SLL) algorithm (Smith et al., 1981) exemplifies this framework through deterministic MST-based expansion. SLL selects Delaunay triangles containing two

MST edges as candidates for Steiner point insertion. When all internal angles are below 120°, the triangle is replaced by its Fermat point, minimizing total edge length while ensuring 120° edge angles (Gilbert and Pollak, 1968). This yields locally-optimal full Steiner topologies with high computational efficiency. However, deterministic expansion strategies suffer from limited topological exploration, often becoming trapped in local minima for clustered or symmetric point configurations (Laarhoven and Ohlmann, 2011). To address this limitation, researchers have introduced stochasticity at multiple algorithmic levels. Beasley and Goffinet (1994) pioneered this direction with two key innovations: iterative expansion that recomputes the DT after each Steiner point addition, enabling dynamic geometric adaptation; and simulated annealing as stochastic exploration policy. Beasley's method examines all DT triangles at each expansion step, ensuring comprehensive local configuration analysis. Laarhoven and Ohlmann (2011) further advanced this approach by demonstrating that stochastic triangle sampling achieves comparable results with significantly reduced computational cost. Instead of exhaustively processing all triangles, their method randomly samples Delaunay simplices during expansion, improving solution quality through broader topological exploration while maintaining efficiency. Similarly, our proposed approach employs stochastic exploration to overcome the limitations of deterministic hyperbolic methods such as García-Castellanos et al. (2025)

Non-Euclidean SMTs. Steiner Minimal Tree problems have been extended to various non-Euclidean metric spaces, including rectilinear SMTs in Manhattan metrics (Hanan, 1966), orientation-dependent metrics (Yan et al., 1997), and Riemannian manifolds (Ivanov et al., 2000; Logan, 2015). Euclidean heuristics have been successfully adapted to spaces of constant curvature, with SLL algorithm extensions to spherical (Dolan et al., 1991) and hyperbolic (García-Castellanos et al., 2025) geometries. Alternative approaches include analytic constructions for hyperbolic Steiner trees using hyperbolic trigonometry in the upper half-plane model (Halverson and March, 2005), though these become computationally intractable for larger terminal sets. To address scalability limitations, García-Castellanos et al. (2025) introduced HyperSteiner, an efficient heuristic that adapts Delaunay-based SMT construction to hyperbolic space. Operating in the Klein model, their method employs angle-based insertion criteria for Steiner points, achieving both effectiveness and scalability (see Algorithm 4). However, like its Euclidean counterparts, this deterministic approach suffers from myopic optimization, limiting topological exploration and potentially yielding suboptimal configurations. Extending Beasley and Goffinet (1994); Laarhoven and Ohlmann (2011), we introduce a stochastic framework that samples candidate Steiner topologies through hyperbolic Delaunay triangulations and refines them via Riemannian gradient descent (Bonnabel, 2013). As demonstrated in Figure 1 (right), our Randomized HyperSteiner overcomes the topological limitations of deterministic methods, discovering superior global configurations through strategic randomization combined with geometric optimization in hyperbolic space.

Hierarchical Inference. Inferring hierarchical structure from observational data is central to fields such as evolutionary biology, phylogenetics, and cell lineage reconstruction, where tree-like relationships describe divergence processes (Cieslik, 2006; Gong et al., 2021). Traditional methods reconstruct trees from sequence data through pairwise distance estimation—as in Unweighted Paired Group Mean Arithmetic (Sokal and Michener, 1958) and Neighbor Joining (Saitou and Nei, 1987)—or probabilistic character-based models such as Maximum Parsimony (Fitch, 1971), Maximum Likelihood (Felsenstein, 1981), and Bayesian frameworks (Huelsenbeck et al., 2001). When formulated to minimize mutational events, phylogenetic inference can be modeled as a Steiner tree problem (Foulds et al., 1979). This connection has been explored in Euclidean geometry (Blelloch et al., 2006; Brazil et al., 2009; Foroughmand-Araabi et al., 2022) and is now investigable in hyperbolic space through García-Castellanos et al. (2025). Recent work has applied variational Bayesian inference in hyperbolic geometry (Mimori and Hamada, 2023; Macaulay and Fourment, 2024; Chen et al., 2025), learning latent representations of molecular data conditioned on biological evolutionary models. However, the resulting phylogenetic trees are not necessarily embedded within hyperbolic space, nor is their construction guided by tree minimality objectives. In contrast, our method applies generally to arbitrary datasets with hyperbolic representations, constructing minimal trees directly embedded within the same hyperbolic space. This enables hierarchical structure inference compatible with existing hyperbolic embedding techniques, such as those developed for biological data (Klimovskaia et al., 2020; Zhou and Sharpee, 2021).

3 BACKGROUND

We focus on the Klein-Beltrami model, Steiner minimal trees, and Delaunay triangulations.

3.1 The Klein-Beltrami Model

The hyperbolic space of dimension n is a complete, simply-connected Riemannian manifold with

constant negative curvature. Several models exist to represent this geometry. Among them, the Klein-Beltrami model is particularly convenient due to its straight-line geodesics facilitating computations. In this model, points lie in the open unit ball $\mathbb{K}^n = \{z \in \mathbb{R}^n \mid \|z\|_2 < 1\}$, and the Riemannian metric is defined by

$$g(z) = \frac{-1}{\langle z, z\rangle} I_n + \frac{1}{\langle z, z\rangle^2} z \otimes z, \tag{1}$$

where $\langle \cdot, \cdot \rangle := -1 + \langle \cdot, \cdot \rangle_E$ denotes the Lorentzian Inner Product in homogeneous coordinates (with $\langle \cdot, \cdot \rangle_E$ being the standard Euclidean inner product), I_n the identity matrix, and $\otimes$ the tensor product. The corresponding geodesic distance between two points $x, y \in \mathbb{K}^n$ is given by:

$$d(x, y) = \operatorname{arccosh}\left(\frac{-\langle x, y\rangle}{\sqrt{\langle x, x\rangle \langle y, y\rangle}}\right). \tag{2}$$

3.2 Steiner Minimal Trees

Let $\mathcal{X}$ be a metric space equipped with a distance function $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$. The *Steiner minimal tree* (SMT) problem seeks to find the shortest network connecting a given set of points (the *terminals*), potentially including additional auxiliary points (the *Steiner points*) not in the original input. This can be formally stated as follows:

Problem 3.1. *Given a finite set of points $P \subseteq \mathcal{X}$, find a finite undirected connected graph* $\mathrm{SMT}(P) = (V, E)$ *embedded in $\mathcal{X}$ such that $P \subseteq V$ and the total edge length $L(\mathrm{SMT}(P))$ is minimized, with:*

$$L(\mathrm{SMT}(P)) = \sum_{(x,y)\in E} d(x, y). \tag{3}$$

The optimal solution is always acyclic and hence forms a tree. Thus, the SMT problem can be reformulated as finding the tree with minimal length that connects a set of points. The number of Steiner points is bounded by $|P| - 2$, and the tree is said to be a *full Steiner tree* (FST) when this bound is reached. For instance, an FST for three terminals contains a single Steiner point located at the *Fermat point*, while configurations with four terminals admit two Steiner points.

The SMT problem is known to be NP-hard (Garey et al., 1977). If no Steiner points are allowed (that is, $V = P$), the solution is simply the *Minimum Spanning Tree* (MST), which can be efficiently computed via algorithms such as the one of Kruskal (1956) with computational time complexity of $\mathcal{O}(|P|^2 \log |P|)$.

SMTs always achieve a length less than or equal to that of MSTs. The quality of the improvement is quantified by the *Steiner ratio*.

Definition 1 (Steiner ratio)**.** *For a finite subset $P \subseteq \mathcal{X}$, the* local Steiner ratio *is defined as*

$$\rho(P) = \frac{L(\mathrm{SMT}(P))}{L(\mathrm{MST}(P))} \leq 1.$$

The global Steiner ratio *of $\mathcal{X}$ is the infimum of $\rho(P)$ over all finite subsets $P \subseteq \mathcal{X}$.*

The Gilbert–Pollak Conjecture states that for the Euclidean plane $\mathcal{X} = \mathbb{R}^2$, the global Steiner ratio equals $\sqrt{3}/2$. The conjecture is still open to this date (Ivanov and Tuzhilin, 2012). For n-dimensional Riemannian manifolds, the global Steiner ratio does not exceed the one of the Euclidean space $\mathbb{R}^n$ of the same dimension (Cieslik, 2001). Notably, in the hyperbolic plane, the global Steiner ratio has been shown to be equal to $\frac{1}{2}$ (Innami and Kim, 2006). In other words, a Steiner tree in hyperbolic space can achieve in theory a total length up to 50% shorter than that of the corresponding MST, whereas in the Euclidean plane the maximum achievable reduction is approximately 13.4%.

3.3 Delaunay Triangulation

As mentioned in Section 2, we focus on the family of heuristics that computes SMT by restricting the search space of potential tree topologies through *Delaunay triangulation* (DT). Due to its relevance in this work, we recall its definition below in any metric space $\mathcal{X}$ equipped with a distance d.

Definition 2. *The* Voronoi cell *of $p \in P \subseteq \mathcal{X}$ is defined as:*

$$V(p) = \{x \in \mathcal{X} \mid \forall q \in P \; d(x, q) \geq d(x, p)\}. \tag{4}$$

The Delaunay triangulation *is defined as the collection of simplices with vertices $\sigma \subseteq P$ such that $\bigcap_{p\in\sigma} V(p) \neq \emptyset$.*

This definition extends to hyperbolic space by replacing the distance function d with the hyperbolic metric given in Equation 2.

4 METHOD

Algorithm 1 summarizes our method. For detailed descriptions of the sub-protocols, see Appendix (Section A). The Randomized HyperSteiner follows the three characteristic stages of DT-based methods (Section 2):

Expansion. At iteration n, given the current best Steiner points, we iteratively compute the hyperbolic DT and probabilistically sample triangles. Following Laarhoven and Ohlmann (2011), we initialize Steiner point candidates with hyperbolic barycenters for fast computation. The step function $\lfloor 2\sqrt{n} - 1 \rfloor$ controls the number of expansions per exploration trial.

Algorithm 1 Randomized HyperSteiner

Input: Terminal point set $P \subset \mathbb{K}^2$, maximum iterations N^*, insertion probability range $[l, u]$

Initialize $n \leftarrow 1$, $S \leftarrow \emptyset$, $S^* \leftarrow \emptyset$, $T^* \leftarrow \mathrm{MST}(P)$.
while $n \leq N^*$ **do**

▷ **Fast Iterative Stochastic Expansion**

for $\lfloor 2\sqrt{n} - 1 \rfloor$ iterations **do**
Compute hyperbolic Delaunay triangulation $\mathcal{T} = \mathrm{DT}(P \cup S)$
Randomly generate the insertion probability, p, from the range $[l, u]$.
for each triangle $\tau \in \mathcal{T}$ **do**
With probability p, compute the **barycenter** point b of τ and let $S = S \cup \{b\}$.
end for
end for

▷ **Heuristic Construction**

▷ **Reduction Step** via Algorithm 2
$S, T \leftarrow$ Degree-Condition(P, S).
Optimize S' via **Riemannian GD**.
▷ **Expansion Step** via Algorithm 3
$S, T \leftarrow$ Angle-Condition(T, P, S).
Optimize S' via **Riemannian GD**.
if $L(T) < L(T^*)$ **then**
▷ **Refinement Step** via Algorithm 2
$S, T \leftarrow$ Degree-Condition(P, S).
Optimize S' via **Riemannian GD**.
end if

▷ **Exploration Policy**

if $L(T) < L(T^*)$ **then**
Set $S^* \leftarrow S$, $T^* \leftarrow T$, $n \leftarrow 1$.
else
Set $S \leftarrow S^*$, $n \leftarrow n + 1$.
end if

end while

Output: A heuristic SMT T^* with Steiner points S^*

Heuristic Construction. Using the expanded candidate set, construct SMT approximations that respect the following properties:

- *Angle condition:* All angles in an SMT are at least 120°, with Steiner point angles exactly 120°.
- *Degree condition:* All SMT vertices have degree at most three, with Steiner points having degree exactly three.
- *Topology–geometry duality:* Given optimal Steiner point locations S, the Steiner tree is $\mathrm{MST}(P \cup S)$. Conversely, given an optimal topology, Steiner point optimization becomes a convex problem minimizing tree length (Laarhoven and Ohlmann, 2011).

Exploration Policy. If the new tree has shorter length than the best-so-far, update the solution and reset the iteration counter; otherwise, discard the proposal and increment n.

We next describe the three core computational components in detail: (i) hyperbolic Delaunay triangulation for expansion, (ii) local full Steiner tree (FST) solutions for heuristic construction, and (iii) global refinement of Steiner points via Riemannian Gradient Descent (GD).

4.1 Hyperbolic Delaunay Triangulation

The Klein-Beltrami model enables efficient computation of hyperbolic Delaunay triangulations, since geodesics are straight lines. In this model, hyperbolic Voronoi cells can be derived from *Euclidean power diagrams* (Nielsen and Nock, 2009), which generalize standard Voronoi diagrams.

Let $\mathcal{X}$ be a metric space with a distance function d. The power cell associated with point $p \in P$ and weight $r_p \geq 0$ is defined as:

$$R(p) = \{x \in \mathcal{X} \mid d(x,p)^2 - r_p^2 \leq d(x,q)^2 - r_q^2,\ \forall q \in P\}.$$

When all weights vanish, power cells reduce to Voronoi cells.

The Klein-Beltrami model admits a non-isometric embedding into Euclidean space, allowing hyperbolic Voronoi cells to be computed as restricted Euclidean power cells. The following result formalizes this equivalence:

Theorem 4.1 (Nielsen and Nock, 2009)**.** *Given $P \subseteq \mathbb{K}^n$, there exists an explicit set $\widetilde{P} \subseteq \mathbb{R}^n$ and weights $\{r_{\widetilde{p}}\}_{\widetilde{p} \in \widetilde{P}}$ such that the hyperbolic Voronoi cells of P correspond to restrictions to $\mathbb{K}^n$ of power cells of $\widetilde{P}$.*

Specifically, the set $\widetilde{P}$ (for computing power cells) and weights $\{r_{\widetilde{p}}\}_{\widetilde{p} \in \widetilde{P}}$ are derived from points $p \in P$ as follows:

$$\widetilde{p} = \frac{p}{2\sqrt{1 - \|p\|^2}}, \qquad r_{\widetilde{p}}^2 = \frac{\|p\|^2}{4(1 - \|p\|^2)} - \frac{1}{\sqrt{1 - \|p\|^2}}. \tag{5}$$

Since power diagrams are efficiently computable in Euclidean space, this allows Delaunay triangulations in hyperbolic geometry to be obtained using standard computational geometry tools.

4.2 Local FST Solutions via Fermat Points

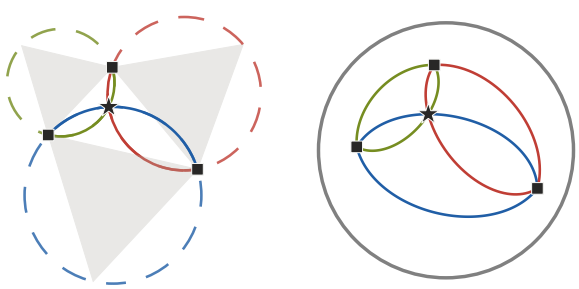
Local refinement of candidate Steiner trees can be reduced to solving small full Steiner tree (FST) subproblems. The previously presented *angle condition*, i.e., incident edges at Steiner points meet at 120°, motivates the introduction of *isoptic curves*:

Definition 3. *Let $\mathcal{X}$ be a complete Riemannian surface and $x, y \in \mathcal{X}$. The* isoptic curve *with angle α, denoted $C_\alpha(x, y)$, is the locus of points $s \in \mathcal{X}$ such that a geodesic from s to x forms an angle α at s with a geodesic from s to y.*

Isoptic curves reduce to circular arcs when $\mathcal{X} = \mathbb{R}^2$. As a direct consequence, Fermat points can be characterized by intersecting three isoptic curves at angle $2\pi/3$: for terminals $P = \{x, y, z\} \subseteq \mathcal{X}$, the Fermat point s satisfies

$$\{s\} = C_{\frac{2\pi}{3}}(x, y) \cap C_{\frac{2\pi}{3}}(y, z) \cap C_{\frac{2\pi}{3}}(z, x). \tag{6}$$

This generalizes the classical Torricelli construction in the Euclidean plane (Figure 2a) to hyperbolic geometry (Figure 2b).



(a) Euclidean, with Torricelli's construction (b) Klein-Beltrami

Figure 2: Example of isoptic curves for $\alpha = 2\pi/3$. Each color represents a different curve. Square (■) denotes terminals while star (★) denotes Steiner points. Source: García-Castellanos et al. (2025).

We now provide an explicit formula for isoptic curves in the Klein-Beltrami model.

Proposition 4.2 (García-Castellanos et al., 2025)**.** *Given two points $x, y \in \mathcal{X} = \mathbb{K}^2$ and an angle $0 < \alpha < \pi$, the isoptic curve $C_\alpha(x, y)$ in the Klein-Beltrami model is given by $\varphi_{x,y,\alpha}(s) = 0$, where:*

$$\varphi_{x,y,\alpha}(s) = \langle x, s\rangle\langle y, s\rangle - \langle x, y\rangle\langle s, s\rangle - \\ - \cos(\alpha)\sqrt{\left(\langle x, s\rangle^2 - \langle x, x\rangle\langle s, s\rangle\right)\left(\langle y, s\rangle^2 - \langle y, y\rangle\langle s, s\rangle\right)}.$$

As the intersection of any two isoptic curves suffices to determine the Fermat point, computing the point s for a terminal set $P = \{x, y, z\} \subseteq \mathbb{K}^2$ reduces to solving a system of two equations, such as:

$$\begin{cases} \varphi_{x,y,\frac{2\pi}{3}}(s) = 0 \\ \varphi_{y,z,\frac{2\pi}{3}}(s) = 0. \end{cases} \tag{7}$$

The resulting nonlinear system can be solved numerically in $\mathcal{O}(1)$ time (see Appendix E.4 for details), for example via the hybrid algorithm proposed by Powell (1970), which extends Newton's method to higher dimensions for root finding.

4.3 Global Refinement via Riemannian GD

While local FST corrections improve topology, they may yield suboptimal global configurations. To address this, we refine Steiner positions by minimizing total tree length over the manifold.

Given a candidate tree $T = (V, E)$ with terminals P and Steiner set $S = V \setminus P$, the objective is

$$\min_{S \subset \mathbb{K}^2} L(S) = \min_{S \subset \mathbb{K}^2} \sum_{(u,v) \in E} d_{\mathbb{K}}(u, v). \tag{8}$$

This non-linear optimization problem can be solved via Riemannian GD (Bonnabel, 2013), which performs updates

$$S^{(t+1)} = \exp_{S^{(t)}}\left(-\eta \nabla L\left(S^{(t)}\right)\right), \tag{9}$$

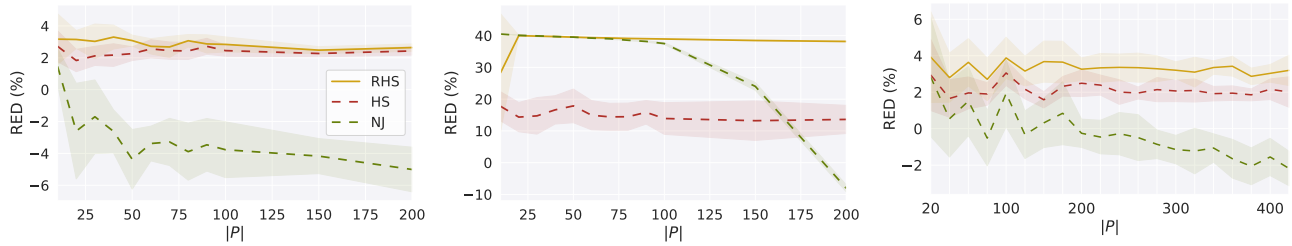
where exp denotes the Riemannian exponential map (applied to each Steiner point) and ∇L the Riemannian gradient of total tree length. This ensures descent along geodesics, extending Euclidean GD to hyperbolic space. Convergence properties mirror the Euclidean case presented in Gilbert and Pollak (1968). Indeed, we show the following uniqueness result:

Theorem 4.3. *Let $P_1, \ldots, P_n$ be fixed terminal points in the Klein disk model of hyperbolic space. Fix a tree topology with s Steiner points $S_1, \ldots, S_s$. Then there exists a unique placement of the Steiner points $S_1, \ldots, S_s$ that minimizes the total tree length L.*

Further details and proof of Theorem 4.3 can be found in the Appendix (Section C).

5 EXPERIMENTS

We empirically evaluate Randomized HyperSteiner (RHS) for constructing hyperbolic Steiner minimal trees with two objectives: (i) quantify tree quality via reduction over the Minimum Spanning Tree (RED, %), and (ii) benchmark against standard



(a) Centered Gaussian $\mathcal{G}(0, 0.5)$ (b) Mixture of 10 Gaussians $\mathcal{G}(\mu_{10,k}(1-10^{-10}), 0.1)$ (c) *Planaria* dataset

Figure 3: Performance comparison across data distributions. Randomized HyperSteiner (orange), vanilla HyperSteiner (red), and Neighbour Joining (green). Curves report mean tree-length reduction over MST (RED, %) with standard deviation (shaded), averaged across 10 runs.

methods—Minimum Spanning Tree (MST), Neighbour Joining (NJ) (Saitou and Nei, 1987), and the deterministic HyperSteiner (HS) heuristic (García-Castellanos et al., 2025). Experiments were conducted on Google Cloud (N2D CPUs, 1 core, 128GB RAM). Source code is available in `https://github.com/AGarciaCast/RandomizedHyperSteiner`; additional details appear in Appendix D.

5.1 Performance Across Data Distributions

We evaluate RHS across three regimes reflecting typical hyperbolic embedding scenarios (Figure 3). For each cardinality $|P|$, results are averaged over 10 independent samplings with standard deviation reported. Runtime and detailed tables appear in Appendix E.

Centered Gaussian. This regime models the initialization phase of hyperbolic embeddings, where points concentrate near the origin (Nickel and Kiela, 2017; Chami et al., 2020). Terminals are sampled from a pseudo-hyperbolic Gaussian $\mathcal{G}(0, 0.5)$ by mapping an isotropic Euclidean Gaussian from the tangent space via the exponential map (Nagano et al., 2019), yielding moderate spread around the disk center.

Mixture of Gaussians Near Boundary. To capture the phenomenon where tree-embedded points concentrate at the boundary in specialized groups (Nickel and Kiela, 2018; Klimovskaia et al., 2020; Kleinberg, 2007; Chami et al., 2020), we construct ten pseudo-hyperbolic Gaussians $\mathcal{G}_k(\mu_{10,k}, 0.1)$ centered at a regular 10-gon near the boundary: $\mu_{10,k} = (1-10^{-10})e^{2i\pi k/10}$ for $k = 1, \ldots, 10$, producing well-separated clusters near the boundary.

Real Biological Data. We validate generalization on the *Planaria* single-cell RNA-sequencing data (Plass et al., 2018), containing 21,612 cells from *S. mediterranea* undergoing hierarchical differentiation. Following PCA dimensionality reduction to 50 components, cells are embedded into two-dimensional hyperbolic space using the method of Klimovskaia et al. (2020), which preserves the biological hierarchy.

Results. Figure 3 demonstrates that RHS consistently outperforms all baselines. In the centered Gaussian regime (Figure 3a), the dense central distribution yields Euclidean behavior as $|P|$ increases due to hyperbolic space's local flatness, producing modest RHS improvements over HS ($\sim$1%) comparable to 2D Euclidean stochastic DT-based heuristics (Laarhoven and Ohlmann, 2011). Randomization becomes critical in the boundary mixture (Figure 3b): RHS maintains substantial reductions as sample size grows, while HS provides only moderate gains and NJ deteriorates to negative performance. On real biological data (Figure 3c), RHS sustains stable improvements over HS, whereas NJ exhibits inconsistent behavior. Notably, despite being a common computational biology baseline, NJ fails to respect the *principle of maximum parsimony*. The marked performance advantage for boundary-concentrated clusters (Figure 3b) motivates a deeper investigation of the relationship between boundary proximity and tree-length reduction.

5.2 Near Boundary Performance Analysis

The substantial improvements for boundary-concentrated clusters raise two fundamental questions: (i) how closely can heuristic methods approach the theoretical upper bound on reduction, and (ii) at what radial distance do these gains materialize? We address both through controlled experiments that systematically vary boundary proximity.

Approaching the Theoretical Limit. We first evaluate performance in the extremal configuration that maximizes tree-length reduction. Following Innami and Kim (2006), the optimal arrangement for $|P|$ terminals is a regular $|P|$-gon positioned at the boundary. For $k = 1, \ldots, |P|$, we sample terminal k from $\mathcal{G}(\mu_{|P|,k}(t), 0.1)$, where $\mu_{|P|,k}(t) = t\, e^{2\pi i k/|P|}$. Setting

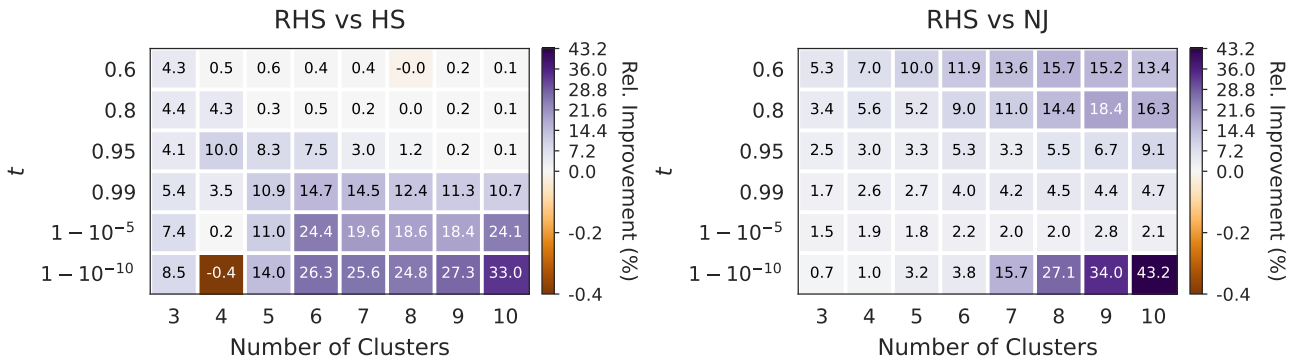


Figure 4: Convergence analysis for mixtures of $\mathcal{G}(\mu_{d,k}(t), 0.1)$, $k \in \{1, \ldots, d\}$ with $d \in \{3, \ldots, 10\}$ and varying radial parameter t, sampling 20 points per Gaussian. Values show percentage reduction in tree-length of RHS vs. HS (left) and NJ (right) as points approach the boundary.

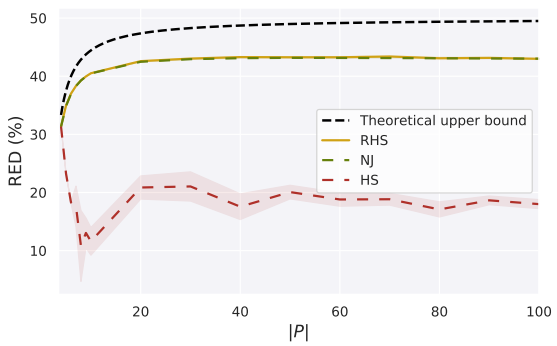


Figure 5: Tree-length reduction over the MST with $|P|$ terminals sampled near the boundary ($t = 1 - 10^{-10}$). RHS (orange) and NJ (green) approach the theoretical bound, while HS (red) lags significantly.

$t = 1 - 10^{-10}$ (the largest numerically stable value), the theoretical upper bound on reduction is $\frac{|P|}{2(|P|-1)}$, which converges to 50% as $|P| \to \infty$. Results for each $|P|$ are averaged over three independent runs.

As we see in Figure 5, RHS and NJ both track this theoretical bound closely, while HS consistently underperforms. RHS achieves a maximum average reduction of 43.39% at $|P| = 70$, compared to 43.16% for NJ at $|P| = 50$, whereas HS plateaus at 31.31%. This translates to a relative tree-length improvement exceeding 32% for RHS over the original HS heuristic. Critically, HS deteriorates rapidly beyond $|P| = 5$, failing to produce competitive solutions, whereas RHS remains robust and near-optimal as terminal count grows.

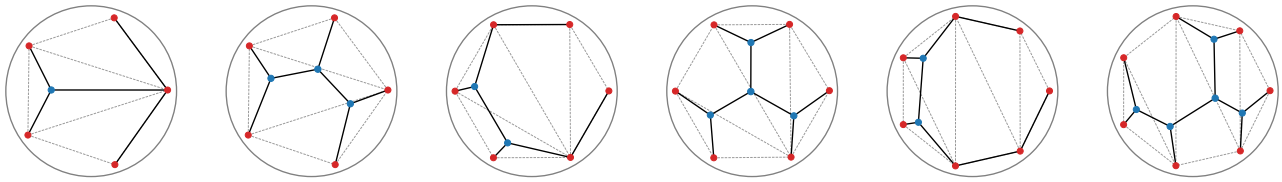
Characterizing the Transition Zone. To identify when proximity-driven gains emerge, we sample 20 points per cluster from mixtures of $d \in \{3, \ldots, 10\}$ Gaussians $\mathcal{G}(\mu_{d,k}(t), 0.1)$, varying the radial parameter t to control distance from the origin. Figure 4 reveals a sharp performance transition. Comparing RHS to HS (left panel), gains are negligible for $t \leq 0.8$ but increase dramatically at $t \geq 0.95$, exceeding 30% improvement with $d = 10$ clusters at the extreme boundary. This transition marks where hyperbolic curvature dominates local Euclidean approximations. While NJ performed comparably to RHS in the single point-per-cluster configuration of Figure 5, the cluster-based setting reveals a significant performance gap, with RHS achieving over 40% improvement in tree-length.

These results demonstrate that RHS's stochastic construction bridges heuristic and optimal performance. As established in Section 3, the hyperbolic plane admits maximum possible MST reduction among all complete boundaryless surfaces. RHS exploits this geometric property to attain near-optimal reductions, representing a significant advance over existing heuristics and revealing a key insight: hyperbolic geometry provides computational advantages for the Steiner tree problem unattainable in Euclidean spaces.

6 LIMITATIONS

Our method shares a key limitation with prior Euclidean stochastic Steiner tree heuristics (Beasley and Goffinet, 1994; Laarhoven and Ohlmann, 2011; Yang, 2006): for general point clouds, it does not provide a theoretical guarantee of finding the globally optimal topology. While Theorem 4.3 shows that, for a fixed topology, the Steiner point configuration is uniquely determined, and thus the global refinement step showcased in Section 4.3 is well posed, the topology itself remains unknown and must be explored heuristically.

Accordingly, RHS relies on stochastic exploration over candidate topologies while retaining the best solution found so far. This makes standard convergence analysis infeasible. In particular, changes in topology can produce discontinuous changes in tree length, which



HS: $|P| = 5$ RHS: $|P| = 5$ HS: $|P| = 6$ RHS: $|P| = 6$ HS: $|P| = 7$ RHS: $|P| = 7$

Figure 6: Heuristic Steiner Minimum Trees computed by the deterministic HyperSteiner (HS) and the proposed Randomized HyperSteiner (RHS) on regular $|P|$-gons placed near the boundary ($t = 0.9$). Red (●) denotes terminals, blue (●) denotes Steiner points, and dashed lines correspond to the auxiliary hyperbolic DT. Our method consistently produces more hierarchical tree structures than its deterministic counterpart.

makes it difficult to obtain a straightforward characterization of convergence or theoretical computational complexity, as in other non-deterministic Steiner tree methods (Beasley and Goffinet, 1994; Laarhoven and Ohlmann, 2011).

Despite these theoretical limitations, the results in Section 5 and Appendix E show that the proposed pipeline consistently improves over existing baselines across a range of synthetic and real hyperbolic datasets.

7 DISCUSSION

We introduced **Randomized HyperSteiner** (RHS), a stochastic Delaunay triangulation heuristic that overcomes the fundamental limitation of deterministic approaches to constructing hyperbolic Steiner Minimal Trees. By incorporating randomness into the expansion process and combining it with Riemannian gradient descent optimization, RHS avoids the local optima that trap existing methods. Moreover, as shown in Figure 6, the myopic behavior of the original HyperSteiner prevents it from generating complex hierarchical trees, whereas our proposed RHS, thanks to our iterative exploration procedure, is able to construct richer hierarchical structures with shorter total tree length.

Empirically, RHS consistently outperforms existing baselines across diverse synthetic and real data distributions, with particularly strong performance in near-boundary configurations where hyperbolic geometry's advantages are most pronounced. However, as is common with stochastic heuristics for the SMT problem (Laarhoven and Ohlmann, 2011), this improved solution quality comes at the cost of increased computational time compared to deterministic approaches like the vanilla HyperSteiner. This presents practitioners with a clear trade-off: RHS should be chosen when solution quality is paramount, while the vanilla HyperSteiner remains suitable for applications requiring fast approximations. We refer the reader to Appendix E.4 for further discussion on the computational trade-off.

Despite these advances, our results reveal that we have not yet achieved the full potential of hyperbolic geometry for tree length minimization. A gap of approximately 7% remains between our best empirical results and the theoretical upper bound. A promising direction for future work involves developing Polynomial-Time Approximation Schemes (PTAS) using quadtrees instead of Delaunay triangulations (Arora et al., 1998; Arora, 1998; Mitchell, 1999). PTAS approaches would allow users to explicitly control the performance-time trade-off by adjusting the approximation parameter. While the construction of quadtrees in hyperbolic space has been shown to be theoretically feasible (Kisfaludi-Bak and van Wordragen, 2025), the practical implementation of these polynomial-time approximation algorithms in hyperbolic geometry remains an open challenge that could finally bridge the gap to optimal performance.

Acknowledgements

This work has been supported by the Swedish Research Council, the Knut and Alice Wallenberg Foundation, the European Research Council (ERC AdG BIRD), and the Swedish Foundation for Strategic Research (SSF BOS). Alejandro García Castellanos is funded by the Hybrid Intelligence Center, a 10-year programme funded through the research programme Gravitation which is (partly) financed by the Dutch Research Council (NWO). This publication is part of the project SIGN with file number VI.Vidi.233.220 of the research programme Vidi which is (partly) financed by the Dutch Research Council (NWO) under the grant `https://doi.org/10.61686/PKQGZ71565`.

References

Absil, P.-A., Mahony, R., and Sepulchre, R. (2008). *Optimization algorithms on matrix manifolds*. Princeton University Press.

Arora, S. (1998). Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782.

Arora, S., Lund, C., Motwani, R., Sudan, M., and Szegedy, M. (1998). Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555.

Beasley, J. E. and Goffinet, F. (1994). A delaunay triangulation-based heuristic for the euclidean steiner problem. *Networks*, 24(4):215–224.

Blelloch, G. E., Dhamdhere, K., Halperin, E., Ravi, R., Schwartz, R., and Sridhar, S. (2006). Fixed parameter tractability of binary near-perfect phylogenetic tree reconstruction. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part I*, ICALP'06, page 667–678, Berlin, Heidelberg. Springer-Verlag.

Bonnabel, S. (2013). Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229.

Bose, J., Smofsky, A., Liao, R., Panangaden, P., and Hamilton, W. (2020). Latent variable modelling with hyperbolic normalizing flows. In *International Conference on Machine Learning*, pages 1045–1055. PMLR.

Boumal, N., Absil, P.-A., and Cartis, C. (2016). Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*.

Brazil, M., Thomas, D., Nielsen, B., Winter, P., Wulff-Nilsen, C., and Zachariasen, M. (2009). A novel approach to phylogenetic tress: d-dimensional geometric steiner tress. *Networks*, 53:104 – 111.

Bridson, M. R. and Haefliger, A. (1999). *Metric Spaces of Non-Positive Curvature*. Springer Berlin Heidelberg.

Cetin, E., Chamberlain, B., Bronstein, M., and Hunt, J. J. (2022). Hyperbolic deep reinforcement learning. *arXiv preprint arXiv:2210.01542*.

Chamberlain, B. P., Clough, J., and Deisenroth, M. P. (2017). Neural embeddings of graphs in hyperbolic space. *arXiv preprint arXiv:1705.10359*.

Chami, I., Gu, A., Chatziafratis, V., and Ré, C. (2020). From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. *Advances in Neural Information Processing Systems*, 33:15065–15076.

Chami, I., Gu, A., Nguyen, D. P., and Re, C. (2021). Horopca: Hyperbolic dimensionality reduction via horospherical projections. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1419–1429. PMLR.

Chen, A., Chlenski, P., Munyuza, K., Moretti, A. K., Naesseth, C. A., and Pe'er, I. (2025). Variational combinatorial sequential monte carlo for bayesian phylogenetics in hyperbolic space. In Li, Y., Mandt, S., Agrawal, S., and Khan, E., editors, *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 2962–2970. PMLR.

Chen, T. and Li, T.-Y. (2015). Homotopy continuation method for solving systems of nonlinear and polynomial equations. *Communications in Information and Systems*, 15(2):119–307.

Chen, W., Han, X., Lin, Y., Zhao, H., Liu, Z., Li, P., Sun, M., and Zhou, J. (2022). Fully hyperbolic neural networks. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5672–5686, Dublin, Ireland. Association for Computational Linguistics.

Chung, F. and Graham, R. (2006). A new bound for euclidean steiner minimal trees. *Annals of the New York Academy of Sciences*, 440:328 – 346.

Cieslik, D. (2001). The steiner ratio of metric spaces. *The Steiner Ratio*, pages 79–98.

Cieslik, D. (2006). *Shortest Connectivity: An Introduction with Applications in Phylogeny*. Combinatorial Optimization. Springer US.

Dolan, J., Weiss, R., and MacGregor Smith, J. (1991). Minimal length tree networks on the unit sphere. *Annals of Operations Research*, 33:501–535.

Elias, I. and Lagergren, J. (2009). Fast neighbor joining. *Theor. Comput. Sci.*, 410:1993–2000.

Felsenstein, J. (1981). Evolutionary trees from dna sequences: a maximum likelihood approach. *Journal of molecular evolution*, 17(6):368—376.

Fitch, W. M. (1971). Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. *Systematic Biology*, 20(4):406–416.

Foroughmand-Araabi, M.-H., Goliaei, S., and McHardy, A. C. (2022). Scelestial: Fast and accurate single-cell lineage tree inference based on a steiner tree approximation algorithm. *PLOS Computational Biology*, 18(8):1–27.

Foulds, L., Hendy, M., and Penny, D. (1979). A graph theoretic approach to the development of minimal phylogenetic trees. *Journal of molecular evolution*, 13:127–49.

Ganea, O., Bécigneul, G., and Hofmann, T. (2018). Hyperbolic neural networks. *Advances in neural information processing systems*, 31.

García-Castellanos, A., Medbouhi, A. A., Marchetti, G. L., Bekkers, E. J., and Kragic, D. (2025). Hypersteiner: Computing heuristic hyperbolic steiner minimal trees. In *2025 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, pages 194–208. SIAM.

Garey, M. R., Graham, R. L., and Johnson, D. S. (1977). The complexity of computing steiner minimal trees. *SIAM journal on applied mathematics*, 32(4):835–859.

Gilbert, E. N. and Pollak, H. O. (1968). Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29.

Gong, W., Granados, A. A., Hu, J., Jones, M. G., Raz, O., Salvador-Martínez, I., Zhang, H., Chow, K.-H. K., Kwak, I.-Y., Retkute, R., Prusokiene, A., Prusokas, A., Khodaverdian, A., Zhang, R., Rao, S., Wang, R., Rennert, P., Saipradeep, V. G., Sivadasan, N., Rao, A., Joseph, T., Srinivasan, R., Peng, J., Han, L., Shang, X., Garry, D. J., Yu, T., Chung, V., Mason, M., Liu, Z., Guan, Y., Yosef, N., Shendure, J., Telford, M. J., Shapiro, E., Elowitz, M. B., and Meyer, P. (2021). Benchmarked approaches for reconstruction of in vitro cell lineages and in silico models of c. elegans and m. musculus developmental trees. *Cell Systems*, 12(8):810–826.e4.

Gulcehre, C., Denil, M., Malinowski, M., Razavi, A., Pascanu, R., Hermann, K. M., Battaglia, P., Bapst, V., Raposo, D., Santoro, A., and de Freitas, N. (2019). Hyperbolic attention networks. In *International Conference on Learning Representations*.

Guo, Y., Guo, H., and Yu, S. X. (2022). Co-sne: Dimensionality reduction and visualization for hyperbolic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21–30.

Halverson, D. and March, D. (2005). Steiner tree constructions in hyperbolic space. *preprint*.

Hanan, M. (1966). On steiner's problem with rectilinear distance. *SIAM Journal on Applied mathematics*, 14(2):255–265.

He, N., Anand, R., Madhu, H., Maatouk, A., Krishnaswamy, S., Tassiulas, L., Yang, M., and Ying, R. (2025a). Helm: Hyperbolic large language models via mixture-of-curvature experts.

He, N., Madhu, H., Bui, N., Yang, M., and Ying, R. (2025b). Hyperbolic deep learning for foundation models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 6021–6031. ACM.

Huang, C.-W., Aghajohari, M., Bose, J., Panangaden, P., and Courville, A. C. (2022). Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35:2750–2761.

Huelsenbeck, J. P., Ronquist, F., Nielsen, R., and Bollback, J. P. (2001). Bayesian inference of phylogeny and its impact on evolutionary biology. *Science*, 294(5550):2310–2314.

Innami, N. and Kim, B. H. (2006). Steiner ratio for hyperbolic surfaces. *Proceedings of the Japan Academy, Series A, Mathematical Sciences*, 82(6):77–79. Publisher: The Japan Academy.

Ivanov, A. O. and Tuzhilin, A. A. (2012). The steiner ratio gilbert–pollak conjecture is still open: Clarification statement. *Algorithmica*, 62:630–632.

Ivanov, A. O., Tuzhilin, A. A., and Cieslik, D. (2000). Steiner ratio for riemannian manifolds. *Russian Mathematical Surveys*, 55(6):1150.

Jaquier, N., Rozo, L., González-Duque, M., Borovitskiy, V., and Asfour, T. (2024). Bringing motion taxonomies to continuous domains via gplvm on hyperbolic manifolds. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Kisfaludi-Bak, S. and van Wordragen, G. (2025). Near-optimal dynamic steiner spanners for constant-curvature spaces. *arXiv preprint arXiv:2509.01443*.

Kleinberg, R. (2007). Geographic routing using hyperbolic space. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pages 1902–1909. IEEE.

Klimovskaia, A., Lopez-Paz, D., Bottou, L., and Nickel, M. (2020). Poincaré maps for analyzing complex hierarchies in single-cell data. *Nature Communications*.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50.

Kunzmann, P. and Hamacher, K. (2018). Biotite: a unifying open source computational biology framework in python. *BMC bioinformatics*, 19:1–8.

Laarhoven, J. and Ohlmann, J. (2011). A randomized delaunay triangulation heuristic for the euclidean steiner tree problem in $\mathbb{R}^d$. *J. Heuristics*, 17:353–372.

Lezcano Casado, M. (2019). Trivializations for gradient-based optimization on manifolds. *Advances in Neural Information Processing Systems*, 32.

Logan, A. E. (2015). *The Steiner Problem on Closed Surfaces of Constant Curvature*. Brigham Young University.

Macaulay, M. and Fourment, M. (2024). Differentiable phylogenetics via hyperbolic embeddings with dodonaphy. *Bioinformatics Advances*, 4(1):vbae082.

Mathieu, E., Le Lan, C., Maddison, C. J., Tomioka, R., and Teh, Y. W. (2019). Continuous hierarchical representations with poincaré variational auto-encoders. *Advances in neural information processing systems*, 32.

Medbouhi, A. A., Marchetti, G. L., Polianskii, V., Kravberg, A., Poklukar, P., Varava, A., and Kragic, D. (2024). Hyperbolic delaunay geometric alignment. In Bifet, A., Davis, J., Krilavičius, T., Kull, M., Ntoutsi, E., and Žliobaitė, I., editors, *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 111–126, Cham. Springer Nature Switzerland.

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

Mishne, G., Wan, Z., Wang, Y., and Yang, S. (2023). The numerical stability of hyperbolic representation learning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 24925–24949. PMLR.

Mitchell, J. S. B. (1999). Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric tsp, k-mst, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309.

Nagano, Y., Yamaguchi, S., Fujita, Y., and Koyama, M. (2019). A wrapped normal distribution on hyperbolic space for gradient-based learning. In *International Conference on Machine Learning*.

Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30.

Nickel, M. and Kiela, D. (2018). Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. *ArXiv*.

Nielsen, F. and Nock, R. (2009). Hyperbolic voronoi diagrams made easy. *CoRR*, abs/0903.3287.

Plass, M., Solana, J., Wolf, F., Ayoub, S., Misios, A., Glažar, P., Obermayer, B., Theis, F., Kocks, C., and Rajewsky, N. (2018). Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics. *Science*, 360:eaaq1723.

Powell, M. J. D. (1970). A hybrid method for nonlinear equations. In Rabinowitz, P., editor, *Numerical Methods for Nonlinear Algebraic Equations*. Gordon and Breach.

Preparata, F. P. and Shamos, M. I. (2012). *Computational geometry: an introduction*. Springer Science & Business Media.

Price, M. N., Dehal, P. S., and Arkin, A. P. (2009). FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix. *Molecular Biology and Evolution*, 26(7):1641–1650.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425.

Sala, F., De Sa, C., Gu, A., and Re, C. (2018). Representation tradeoffs for hyperbolic embeddings. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4460–4469. PMLR.

Sarkar, R. (2011). Low distortion delaunay embedding of trees in hyperbolic plane. In *International symposium on graph drawing*, pages 355–366. Springer.

Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. *Biometrics*, 2 6:110–4.

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *J. Pharmacokinet. Biopharm.*, 15(6):657–680.

Shimizu, R., Mukuta, Y., and Harada, T. (2021). Hyperbolic neural networks++. In *International Conference on Learning Representations*.

Smith, J. M., Lee, D., and Liebman, J. S. (1981). An o (n log n) heuristic for steiner minimal tree problems on the euclidean metric. *Networks*, 11(1):23–39.

Sokal, R. R. and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas science bulletin*, 38:1409–1438.

Studier, J. and Keppler, K. (1988). A note on the neighbor-joining algorithm of saitou and nei. *Molecular biology and evolution*, 5:729–31.

Taleb, F., Medbouhi, A. A., Marchetti, G. L., and Kragic, D. (2025). Towards discovering the hierarchy of the olfactory perceptual space via hyperbolic embeddings. *Science Communications Worldwide*.

Thompson, E. (1973). The method of minimum evolution.

Tifrea, A., Bécigneul, G., and Ganea, O.-E. (2018). Poincar\'e glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*.

Ungar, A. A. (2009). *A Gyrovector Space Approach to Hyperbolic Geometry*. Springer International Publishing.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Welch, B. L. (1938). The significance of the difference between two means when the population variances are unequal. *Biometrika*, 29(3-4):350–362.

Yan, G., Albrecht, A., Young, G., and Wong, C.-K. (1997). The steiner tree problem in orientation metrics. *journal of computer and system sciences*, 55(3):529–546.

Yang, B. (2006). A hybrid evolutionary algorithm for the euclidean steiner tree problem using local searches. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 60–67. Springer.

Yang, M., Verma, H., Zhang, D. C., Liu, J., King, I., and Ying, R. (2024). Hypformer: Exploring efficient transformer fully in hyperbolic space. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 3770–3781, New York, NY, USA. Association for Computing Machinery.

Zachariasen, M. and Winter, P. (1999). Concatenation-Based Greedy Heuristics for the Euclidean Steiner Tree Problem. *Algorithmica*, 25(4):418–437.

Zhou, Y. and Sharpee, T. O. (2021). Hyperbolic geometry of gene expression. *iScience*, 24(3):102225.

Zhou, Y., Smith, B. H., and Sharpee, T. O. (2018). Hyperbolic geometry of the olfactory space. *Science Advances*, 4(8):eaaq1458.

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.
   Answer: [**Yes**].
   Justification: Section 3 provides comprehensive background on the Klein-Beltrami model, Steiner Minimal Trees, and Delaunay triangulation. Section 4 details the method with Algorithm 1, and Appendix A provides additional implementation details (Algorithms 2-3).
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm.
   Answer: [**Yes**].
   Justification: Appendices A and B provide theoretical complexity analysis of main algorithm components, while Appendix E presents detailed empirical runtime measurements across varying problem sizes and includes an extended scalability discussion.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries.
   Answer: [**Yes**].
   Justification: As stated in Section 5, the source code and experiment scripts are available in the Supplementary Material.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results.
   Answer: [**Yes**].
   Justification: Theorem 4.3 clearly states assumptions (fixed terminal points in Klein disk, fixed tree topology). Proposition 4.2 and other theoretical results explicitly state their prerequisites.
   (b) Complete proofs of all theoretical results.
   Answer: [**Yes**].
   Justification: Appendix C provides complete proofs including Lemma C.1, Theorem 4.3, and Lemma C.3. Other results contain citations to references with their corresponding proofs.
   (c) Clear explanations of any assumptions.
   Answer: [**Yes**].
   Justification: Appendix C provides complete proofs including Lemma C.1, Theorem 4.3, and Lemma C.3. Other results contain citations to references with their corresponding proofs.

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).
   Answer: [**Yes**].
   Justification: As stated in Section 5, the source code and experiment scripts are available in the Supplementary Material (for the submission). For the camera-ready version, we included a github public link containing all the code to reproduce the experiments.

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen).
   Answer: [**Yes**].
   Justification: Details regarding the experimental section configuration can be found in Section 5, as well as in Appendix D.

   (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).
   Answer: [**Yes**].
   Justification: For each experiment, results are averaged over 10 independent samplings, reporting mean reduction over MST (RED, %) with standard deviation.

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider).
   Answer: [**Yes**].
   Justification: As stated in Section 5, all the experiments were conducted on Google Cloud (n2d CPUs, 1 core, 128GB RAM).

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.
   Answer: [**Yes**].
   Justification: The paper properly cites: García-Castellanos et al. (2025) for HyperSteiner, Plass et al. (2018) for the Planaria dataset and Klimovskaia et al. (2020) for its hyperbolic embedding, Saitou and Nei (1987) for Neighbour Joining, and all other methods/datasets used.

   (b) The license information of the assets, if applicable.
   Answer: [**Not Applicable**].
   Justification: All the used assets are available under Creative Commons Attribution (CC BY) licenses, which permit reuse with appropriate citation as provided in the references.

   (c) New assets either in the supplemental material or as a URL, if applicable.
   Answer: [**Yes**].
   Justification: Code and scripts mentioned in supplementary material (during submission) and a provided github link in the camera-ready version.

   (d) Information about consent from data providers/curators.
   Answer: [**Not Applicable**].
   Justification: All the used assets are available under Creative Commons Attribution (CC BY) licenses, which permit reuse with appropriate citation as provided in the references.

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content.
   Answer: [**Not Applicable**].
   Justification: The paper uses synthetic data and single-cell RNA-sequencing data with no PII or sensitive content.

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.
   Answer: [**Not Applicable**].
   Justification: This is a purely computational/algorithmic study with no human subjects or crowdsourcing.

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable.
   Answer: [**Not Applicable**].
   Justification: This is a purely computational/algorithmic study with no human subjects or crowdsourcing.

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation.
   Answer: [**Not Applicable**].
   Justification: This is a purely computational/algorithmic study with no human subjects or crowdsourcing.

Randomized HyperSteiner: Supplementary Materials

A Additional Details of Randomized HyperSteiner

This appendix provides further details on the design and implementation of the proposed Randomized HyperSteiner (RHS) method.

A.1 Expansion Phase

In the expansion phase, we iteratively construct the Delaunay triangulation (DT) of the union of the terminal set and the current Steiner candidates S. Following Laarhoven and Ohlmann (2011), we begin by sampling an insertion probability $p \sim \mathcal{U}[0.3, 0.6]$. Then, for each triangle in $\mathrm{DT}(P \cup S)$, we sample from a Bernoulli distribution with success probability p to decide whether to insert its hyperbolic barycenter as a new Steiner candidate. For three points (u, v, w) in the Klein disk, the hyperbolic barycenter is given by

$$m_{uvw} = \frac{\gamma_u u + \gamma_v v + \gamma_w w}{\gamma_u + \gamma_v + \gamma_w}, \quad \gamma_u = \frac{1}{\sqrt{1 - \|u\|^2}},$$

where γ_u is the Lorentz factor (Ungar, 2009).

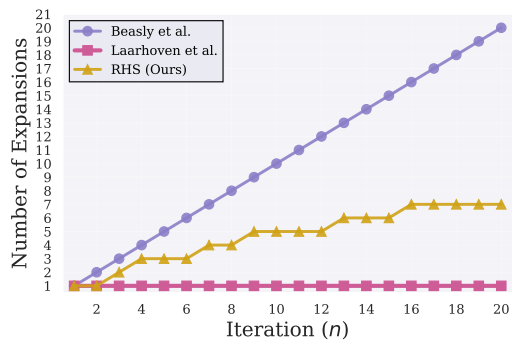


Figure 7: Comparison of expansion strategies.

At iteration n, we perform $\lfloor 2\sqrt{n} - 1 \rfloor$ expansion steps (see Algorithm 1). As discussed in Section 4, this schedule provides finer control of the search process and allows revisiting promising configurations. By contrast, Beasley and Goffinet (1994) perform exactly n expansions per iteration, which scales linearly and may lead to excessive growth in later stages (see Figure 7).

Detailed ablations regarding the insertion probabilities and the expansion scheduler can be found in Section E.5 and Section E.6.

A.2 Verification of Degree and Angle Conditions

Algorithms 2 and 3 enforce, respectively, the degree and angle conditions for candidate Steiner configurations.

Degree condition. Algorithm 2 locally verifies degree constraints by evaluating candidate full Steiner trees on three- and four-point neighborhoods. For the three-point case, the system of equations in (7) can be solved using standard numerical solvers. For the four-point case, we follow the iterative approximation as done in García-Castellanos et al. (2025). Moreover, Algorithm 2 leverages the fact that, given optimal Steiner points S, the resulting Steiner tree is the minimum spanning tree (MST) on $P \cup S$. Since in hyperbolic space the MST is always a subgraph of the DT (Preparata and Shamos, 2012), we can compute the MST via the DT in $\mathcal{O}(|P \cup S'| \log |P \cup S'|)$ time, avoiding redundant computation (Smith et al., 1981).

Angle condition. Algorithm 3 checks the 120° angle condition in constant time using the hyperbolic cosine rule. Specifically, the angle between edges (x_i, x_j) and (x_j, x_l) is given by

$$\theta = \arccos\left(\frac{\cosh(d_{\mathbb{K}}(x_j, x_i)) \cosh(d_{\mathbb{K}}(x_j, x_l)) - \cosh(d_{\mathbb{K}}(x_i, x_l))}{\sinh(d_{\mathbb{K}}(x_j, x_i)) \sinh(d_{\mathbb{K}}(x_j, x_l))} \right).$$

Algorithm 2 Reduction via Degree Condition Verification — Adapted from Beasley and Goffinet (1994)

```
Input: Terminal set P ⊂ 𝕂², Steiner set S ⊂ 𝕂²
S′ ← S
T ← MST(P ∪ S′)
repeat
  for each s ∈ S′ do
    d ← deg_T(s)
    if d ≤ 2 or d ≥ 5 then
      S′ ← S′ \ {s}
    else if d = 3 then
      N ← neighbors of s in T
      s* ← optimal Steiner point for triangle N
      if s* exists then
        Replace s with s* in S′
      else
        S′ ← S′ \ {s}
      end if
    end if
  end for
  T ← MST(P ∪ S′)
until no changes in S′
for each s ∈ S′ do
  if deg_T(s) = 4 then
    N ← neighbors of s in T
    Q, (s1, s2) ← optimal Full Steiner tree for 4-point set N, and Steiner points
    S′ ← (S′ \ {s}) ∪ {s1, s2}
    Substitute the subgraph of the neighborhood of s on T by Q
  end if
end for
Output: Reduced Steiner set S′, tree T on P ∪ S′
```

A.3 Global Refinement via Riemannian Gradient Descent

As discussed by Lezcano Casado (2019), evaluating the exponential map can be computationally expensive, motivating the use of retractions as efficient first-order approximations.

Definition 4 (Retraction)**.** *A differentiable map $r : T\mathcal{M} \to \mathcal{M}$ is a* retraction *if, for every $p \in \mathcal{M}$, the map $r_p : T_p\mathcal{M} \to \mathcal{M}$ satisfies:*

1. *$r_p(0) = p$ (identity at the origin), and*
2. *$(\mathrm{d}\, r_p)_0 = \mathrm{Id}$ (first-order equivalence to the exponential map).*

Retractions provide efficient approximations of geodesic flow near the origin. For our problem, the update rule becomes

$$S^{(t+1)} = r_{S^{(t)}}\left(-\eta \nabla L\left(S^{(t)}\right)\right),$$

where L is the tree length function and η is the learning rate. As shown by Boumal et al. (2016), gradient descent with retractions preserves convergence guarantees of exponential map-based methods while being significantly more efficient.

To further reduce runtime, we apply early stopping upon convergence (see Appendix E). While we use a fixed learning rate, more advanced line-search strategies (e.g., Armijo backtracking (Absil et al., 2008)) could further improve stability and convergence speed.

Algorithm 3 Expansion via Angle Condition Verification — Adapted from Thompson (1973)

Input: Terminal point set $P \subset \mathbb{K}^2$, Steiner point set $S \subset \mathbb{K}^2$, tree T on $P \cup S$

```
Let E be the set of edges in T.
Set T' ← T, S' ← S, and edge list ← ∅.
for i = 1, ..., |X ∪ S| do
  for j such that (x_i, x_j) ∈ E do
    for l such that (x_j, x_l) ∈ E do
      if (x_j, x_l) meets (x_i, x_j) at angle less than 120° then
        edge list ← edge list ∪ (x_j, x_l).
      end if
    end for
    if edge list ≠ ∅ then
      From edge list, select (x_j, x_k) which minimizes the angle with (x_i, x_j).
      s* ← optimal Steiner point for triangle △(x_i, x_j, x_k)
      if s* exists then
        Let S' ← S ∪ s*.
        Remove edges (x_i, x_j) and (x_j, x_k) from T'.
        Add edges (x_i, s*), (x_j, s*), and (x_k, s*) to T'.
      end if
      Set edge list ← ∅.
    end if
  end for
end for
```

Output: Expanded Steiner set S', tree T' on $P \cup S'$

A.4 Exploration Policy

Following Laarhoven and Ohlmann (2011), we set the maximum number of iterations to $N^* = \lfloor \sqrt{|P|} \rfloor$. This choice balances exploration and efficiency, enabling sufficient search depth to discover improved topologies while avoiding wasted computation. The algorithm maintains the best solution found (T^*) and continues exploring until either (i) a better solution is identified ($L(T) < L(T^*)$), triggering a restart from the new checkpoint, or (ii) the iteration budget N^* is exhausted. Figure 8 illustrates this policy.

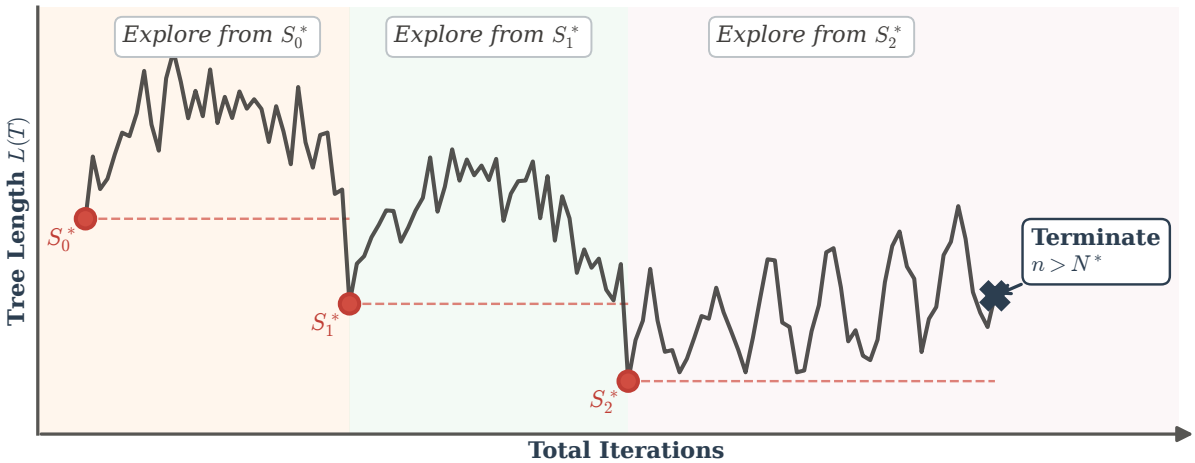


Figure 8: Exploration policy of RHS. The search restarts from a new checkpoint when a better configuration is discovered.

B Review on Original HyperSteiner

In this section, we review the original HyperSteiner algorithm presented in García-Castellanos et al. (2025). As mentioned in Section 2, Algorithm 4 is based on the Smith-Lee-Liebman (SLL) algorithm (Smith et al., 1981) with a minor modification from Zachariasen and Winter (1999).

Algorithm 4 Original HyperSteiner

Input: Terminal point set $P \subset \mathbb{K}^2$

▷ **Deterministic MST-Based Expansion**

1. Construct the Delaunay triangulation, $\mathrm{DT}(P)$.
2. Construct $\mathrm{MST}(P)$ (Kruskal algorithm) and simultaneously build a priority queue as follows:
   - 2.1. Mark all the triangles $\sigma \in \mathrm{DT}(P)$ containing two edges of $\mathrm{MST}(P)$ and admitting an FST.
   - 2.2. Place the FSTs of marked triangles σ in a queue Q prioritized on $\rho(\sigma)$ (smaller first).

▷ **Heuristic Construction**

3. Add the FST of the 4-terminal subsets:
   - 3.1. For each marked triangle σ, find its adjacent triangles σ' such that σ and σ' contain three edges of the $\mathrm{MST}(P)$.
   - 3.2. Compute the FST $\sigma \cup \sigma'$ for each of the two possible topologies and add the minimal one to Q.
4. Convert Q to an ordered list and append to it the edges of $\mathrm{MST}(P)$, sorted in non-decreasing order.
5. Let T be an empty tree. An FST in Q is added to T if it does not create a cycle (greedy concatenation).

Output: A heuristic SMT T

The SLL algorithm uses the MST as an initial approximation of the SMT. This approach relies on the Gilbert-Pollak conjecture that the Steiner ratio satisfies $\rho(P) \geq \sqrt{3}/2 \approx 0.866$ for any $P \subset \mathbb{R}^2$, which implies $L(\mathrm{MST}(P)) \leq 2/\sqrt{3} \cdot L(\mathrm{SMT}(P))$*. Similarly, the original HyperSteiner is based on the fact that, as described in Section 3, on hyperbolic space the Steiner ratio satisfies $\rho(P) \geq 1/2$ for any $P \subset \mathbb{K}^2$ (Innami and Kim, 2006).

The original HyperSteiner algorithm replaces MST edges connecting 3-terminal and 4-terminal subsets with their corresponding full Steiner trees (FSTs). Since the MST is a subgraph of the Delaunay triangulation (Preparata and Shamos, 2012), the algorithm restricts its search to vertices of Delaunay triangles.

This algorithm follows the standard three-phase DT-based framework from Section 2: expansion, heuristic construction, and decision policy. The expansion phase builds a priority queue of 3-terminal FSTs based on Steiner ratios. The construction phase adds 4-terminal FSTs from adjacent triangle pairs. The decision policy uses greedy concatenation, accepting all improvements that do not create cycles.

As shown in Section 5, the original HyperSteiner can be myopic due to its reliance on the MST as a guiding structure, leading to locally optimal but globally suboptimal configurations. However, this approach achieves fast computational complexity of $\mathcal{O}(|P| \log |P|)$.

*The Gilbert-Pollak conjecture remains open (Ivanov and Tuzhilin, 2012). The best known lower bound is ~ 0.824 (Chung and Graham, 2006).

C Uniqueness of Relatively Minimal Trees in Hyperbolic Space

Lemma C.1. *Let x, y be two distinct points strictly inside the Klein disk and let $z \neq z'$ be two distinct points. Parametrize the geodesic segment*

$$s(t) = (1-t)z + tz', \qquad t \in [0,1].$$

Then the entire segment $[z, z']$ lies in the $120°$–isoptic of (x, y) if and only if $z = z'$. In particular, if $z \neq z'$ the whole segment cannot be contained in the isoptic.

Proof. The automorphism group of the disk acts transitively on ordered pairs of distinct interior points and consists of hyperbolic isometries. Thus there exists a disk automorphism Φ with

$$\Phi(z) = (0,0), \qquad \Phi(z') = (r,0) \text{ with } r \in (0,1).$$

Because Φ is an isometry, the hyperbolic distance is preserved and

$$0 \neq d_{\mathbb{K}}(z, z') = d_{\mathbb{K}}((0,0),(r,0)) = 2\tanh^{-1}(r),$$

so $r = \tanh(\frac{1}{2}d_{\mathbb{K}}(z, z')) \in (0,1)$. In particular r is uniquely determined by the hyperbolic length of $[z, z']$. The isoptic property is preserved by Φ, so it suffices to prove the statement for the normalized configuration with $\hat{z} := \Phi(z) = (0,0)$ and $\hat{z}' := \Phi(z') = (r,0)$.

Write $\hat{x} := \Phi(x) = (a,b)$ and $\hat{y} := \Phi(y) = (c,d)$ with $a^2 + b^2 < 1,\ c^2 + d^2 < 1$. Parametrize the normalized segment by

$$s(t) = (rt, 0), \qquad t \in [0,1].$$

Let $\langle u, v\rangle = -1 + u_0v_0 + u_1v_1$ denote the Lorentzian inner product. Define

$$\begin{aligned} u(t) &= \langle \hat{x}, s(t)\rangle, & v(t) &= \langle \hat{y}, s(t)\rangle, & w(t) &= \langle s(t), s(t)\rangle, \\ L(t) &= u(t)v(t) - \langle \hat{x}, \hat{y}\rangle w(t), & p_1(t) &= u(t)^2 - \langle \hat{x}, \hat{x}\rangle w(t), & p_2(t) &= v(t)^2 - \langle \hat{y}, \hat{y}\rangle w(t). \end{aligned}$$

Hence, we can rewrite the isoptic condition of Proposition 4.2, as

$$\varphi(s(t)) = L(t) + \tfrac{1}{2}\sqrt{p_1(t)p_2(t)} = 0.$$

If $\varphi(s(t)) \equiv 0$ on the entire segment, then squaring yields the polynomial identity

$$Q(t) := 4L(t)^2 - p_1(t)p_2(t) \equiv 0. \tag{10}$$

Since $\deg Q \leq 4$, this means all coefficients of $Q(t)$ vanish.

Expanding Q with a Computer Algebra System (CAS), as done in Figure 9, one finds that the coefficient of t^4 is

$$E_4 = r^4\Big(3b^2d^2 + b^2 - 8bd + d^2 + 3\Big) = r^4\Big(3(bd-1)^2 + (b-d)^2\Big).$$

Because $0 < r < 1$ we have $r^4 > 0$, and the parenthesis is a sum of two squares, hence nonnegative. Equality $E_4 = 0$ would force simultaneously

$$(bd)r^2 = 1, \qquad b = dr^2,$$

which implies $d^2 = 1/r^4 > 1$. This contradicts the assumption that $\hat{y} = (c,d)$ lies strictly inside the unit disk, since $c^2 + d^2 < 1$ requires $|d| < 1$. Therefore $E_4 > 0$ for any two interior foci $\hat{x}, \hat{y}$.

Hence the quartic $Q(t)$ cannot vanish identically on the line. Thus the entire nontrivial segment $[z, z']$ cannot lie in the isoptic. The only case where the whole segment is contained is when $z = z'$, which is trivial. □

Proposition C.2 (Bridson and Haefliger (1999))**.** *Let $(\mathcal{H}, g)$ be a Hadamard manifold, i.e., a complete, simply connected Riemannian manifold with sectional curvature $K < 0$ (this includes hyperbolic spaces). For any two geodesics $\gamma, \eta : [0,1] \to \mathcal{H}$, the distance function $t \mapsto d(\gamma(t), \eta(t))$ is convex on $[0,1]$. That is,*

$$d(\gamma(t), \eta(t)) \leq (1-t)d(\gamma(0), \eta(0)) + td(\gamma(1), \eta(1)) \tag{11}$$

for all $t \in [0,1]$.

```python
import sympy as sp

a,b,c,d,r,t = sp.symbols('a b c d r t', real=True)
x = sp.Matrix([a,b]); y = sp.Matrix([c,d]); s = sp.Matrix([r*t,0])
inner = lambda u,v: -1 + u.dot(v)
u = inner(x,s); v = inner(y,s); w = inner(s,s)
L = sp.simplify(u*v - inner(x,y)*w)
p1 = sp.simplify(u**2 - inner(x,x)*w)
p2 = sp.simplify(v**2 - inner(y,y)*w)
Q = sp.expand(4*L**2 - p1*p2)
E4 = sp.Poly(Q, t).all_coeffs()[0]  # coeff of t^4
sp.factor(E4)
```

>>> $r^4(3b^2d^2 + b^2 - 8bd + d^2 + 3)$

Figure 9: Simple SymPy code snippet for computing the coefficients of the quartic presented in Equation 10.

Notation. Let $P_1, \ldots, P_n$ be fixed terminal points in the Klein disk model of hyperbolic space. Fix a tree topology with adjacency weights $a_{ij} \in \{0, 1\}$ and s Steiner points. A *configuration* of vertices is a tuple

$$V = (V_1, \ldots, V_{n+s}) \quad \text{with} \quad V_i = P_i \text{ for } 1 \le i \le n, \;\; V_k = S_k \text{ for } k > n.$$

The total tree length is

$$L(V) = \sum_{i<j} a_{ij}\, d_{\mathbb{K}}(V_i, V_j),$$

where $d_{\mathbb{K}}(\cdot, \cdot)$ is the Klein distance.

Given two configurations $V(0)$ and $V(1)$ with the same terminals but different Steiner points, define the interpolated configuration $V(t)$ for $t \in [0, 1]$ by

$$V_i(t) = \begin{cases} P_i, & i \le n \quad \text{(terminals fixed)}, \\ (1-t)S_i + tS_i', & i > n \quad \text{(Steiner interpolation)}. \end{cases}$$

Since geodesics in the Klein disk are Euclidean chords, each $t \mapsto V_i(t)$ is a hyperbolic geodesic (possibly constant if $i \le n$). The length of the interpolated tree is

$$L(t) = L(V(t)) = \sum_{i<j} a_{ij}\, d_{\mathbb{K}}(V_i(t), V_j(t)).$$

Theorem 4.3 (Restated). *Given a fixed tree topology and fixed terminals $P_1, \ldots, P_n$ in the Klein disk model, the minimizing configuration of Steiner points $S_{n+1}, \ldots, S_{n+s}$ is unique.*

Proof. This proof will closely follow the steps of the Euclidean counter part presented in Gilbert and Pollak (1968), but showing that can be adapted for the Klein disk.

Step 1. Convexity of $L(t)$. Fix a pair (i, j) with $a_{ij} = 1$. Both $t \mapsto V_i(t)$ and $t \mapsto V_j(t)$ are geodesics (constant if terminal, chordal if Steiner). By Proposition C.2, the function

$$t \;\mapsto\; d_{\mathbb{K}}\big(V_i(t), V_j(t)\big)$$

is convex. Explicitly,

$$d_{\mathbb{K}}\big(V_i(t), V_j(t)\big) \le (1-t) d_{\mathbb{K}}\big(V_i(0), V_j(0)\big) + t d_{\mathbb{K}}\big(V_i(1), V_j(1)\big). \tag{12}$$

(If one of the geodesics is constant — corresponding to a terminal — the statement still holds: the distance to a fixed point is convex along any geodesic.)

Multiplying (12) by $a_{ij} \geq 0$ and summing over all pairs $i < j$, we obtain

$$L(t) = \sum_{i<j} a_{ij}\, d_{\mathbb{K}}(V_i(t), V_j(t)) \;\leq\; (1-t)\sum_{i<j} a_{ij}\, d_{\mathbb{K}}(V_i(0), V_j(0)) + t\sum_{i<j} a_{ij}\, d_{\mathbb{K}}(V_i(1), V_j(1)).$$

That is,

$$L(t) \leq (1-t)L(0) + tL(1). \tag{13}$$

Hence $L(t)$ is convex in t.

Step 2. Equality at minima. Suppose two configurations $V(0)$ and $V(1)$ both minimize the tree length, i.e. $L(0) = L(1) = L_{\min}$. Then from (13),

$$L(t) \leq (1-t)L(0) + tL(1) = L_{\min}.$$

But since $L(t)$ is by definition the length of a feasible tree, we must also have $L(t) \geq L_{\min}$. Therefore $L(t) = L_{\min}$ for all $t \in [0, 1]$.

Step 3. Rigidity of Steiner points. Consider a Steiner point S in $V(0)$ and its counterpart S' in $V(1)$. At a minimizing configuration, incident edges meet at 120°. Along the interpolation $S(t)$, the Steiner point would have to remain in the 120°–isoptic of its neighbors. By Lemma C.1, this is only possible if $S = S'$. Thus all Steiner points coincide in the two minimizing configurations.

Step 4. Induction on the number of Steiner points. If $s = 0$, there are no Steiner points and the tree is uniquely determined by the terminals and topology. Assume uniqueness holds for $s - 1$ Steiner points. For s Steiner points, Step 3 shows that any two minimal configurations agree on at least one Steiner point. Treating this common Steiner point as an additional fixed terminal reduces the problem to $s - 1$ Steiner points, where uniqueness holds by the induction hypothesis. Thus the full configuration is unique.

□

Lemma C.3 (Model-independence)**.** *Let $\mathbb{H}^2$ denote 2-dimensional hyperbolic space, and let $\mathcal{M}_1, \mathcal{M}_2$ be two standard models of $\mathbb{H}^2$ (e.g. Klein disk, Poincaré disk, upper half–plane, hyperboloid). Suppose the Steiner problem with fixed terminals has a unique minimizing configuration of Steiner points in $\mathcal{M}_1$. Then the corresponding minimizing configuration in $\mathcal{M}_2$ is also unique.*

Proof. There exists a bijective isometry

$$\iota : \mathcal{M}_1 \to \mathcal{M}_2$$

between the two models. For any configuration V in $\mathcal{M}_1$, define $\iota(V)$ as the image configuration in $\mathcal{M}_2$. Since ι is an isometry, we have

$$d_{\mathcal{M}_2}\big(\iota(x), \iota(y)\big) \;=\; d_{\mathcal{M}_1}(x, y) \qquad \text{for all } x, y \in \mathcal{M}_1.$$

Therefore the tree length functional is preserved:

$$L_{\mathcal{M}_2}(\iota(V)) \;=\; L_{\mathcal{M}_1}(V).$$

In particular, ι maps minimizing configurations in $\mathcal{M}_1$ bijectively to minimizing configurations in $\mathcal{M}_2$. If the minimizer is unique in $\mathcal{M}_1$, its image is unique in $\mathcal{M}_2$. □

Since we have established in Theorem 4.3 that the minimizing configuration of Steiner points is unique in the Klein disk model, and since all standard models of hyperbolic space are mutually isometric, Lemma C.3 implies the following general statement:

In hyperbolic space, given a fixed terminal set and a fixed tree topology, the optimal locations of the Steiner points are unique, independent of the chosen model.

D Additional Experimental Details

This appendix provides implementation details for the three methods compared in our experiments: Randomized HyperSteiner (RHS), the original HyperSteiner (HS), and Neighbor Joining (NJ).

D.1 Hyperbolic SMT Heuristic Methods

Randomized HyperSteiner (RHS). We configure the global optimization procedure presented in Section 4.3, with the following hyperparameters, selected through tuning: a maximum of 10,000 epochs, learning rate $\eta = 1 \times 10^{-2}$, early stopping with patience of 100 epochs, and a convergence threshold of $\delta = 1 \times 10^{-6}$.

Although optimal learning rates are dataset-dependent, extensive hyperparameter tuning shows that $\eta = 1 \times 10^{-2}$ performs consistently well across all datasets. As noted in Appendix A.3, more advanced optimization schemes (e.g., Armijo backtracking (Absil et al., 2008) or Adam via trivializations (Lezcano Casado, 2019)) could further reduce this dependence.

For constructing full Steiner trees on four terminal points, we adopt the recursive decomposition approach of Halverson and March (2005) with three recursion levels as recommended by García-Castellanos et al. (2025). Moreover, for the three-point FSTs, we will solve the system of isoptic curves (7), using Powell's hybrid method (Powell, 1970). While García-Castellanos et al. (2025) demonstrated that Polynomial Homotopy Continuation methods (Chen and Li, 2015) can achieve better solutions, the computational overhead outweighs the minor performance gains for our application.

Original HyperSteiner (HS). For consistency, we configure HS identically to RHS: both use the recursive solver for four-point trees and Powell's method for three-point isoptic curves.

D.2 Neighbor Joining Method

Algorithm Overview. Neighbor Joining (NJ) (Saitou and Nei, 1987) is a greedy agglomerative algorithm for phylogenetic tree reconstruction from pairwise distances. The method maintains a distance matrix and iteratively merges pairs of nodes into internal nodes using the Q-criterion, which corrects for varying evolutionary rates. NJ runs in $\mathcal{O}(n^3)$ time and produces a unique topology without requiring a molecular clock assumption.

Implementation and Adaptation. NJ produces a tree topology together with edge lengths, yet the tree itself is not embedded in hyperbolic space. In line with Mimori and Hamada (2023), we therefore disregard the edge lengths and retain only the topology. We subsequently extend NJ by embedding this topology into hyperbolic space via Riemannian gradient descent as follows:

1. **Distance Matrix Construction.** Compute pairwise distances between all terminal points using the hyperbolic metric in the Klein disk.
2. **Topology Inference.** Apply standard NJ to the distance matrix, yielding a tree topology T_{NJ} with internal nodes (interpreted as Steiner points). We implement NJ via the Biotite Python library (Kunzmann and Hamacher, 2018).
3. **Geometric Refinement Embedding.** Fixing topology T_{NJ}, optimize Steiner point positions to minimize total tree length in hyperbolic space. We initialize positions by sampling from a pseudo-hyperbolic Gaussian $\mathcal{G}(0, 0.1)$ and optimize via Riemannian gradient descent with learning rate $\eta = 1$ for 10,000 epochs—see Section 4.3 for further details regarding the optimization problem given a fixed topology.

E Additional Results

E.1 Performance Across Data Distributions

To complement the analysis in Section 5.1, we provide additional visualizations and detailed numerical results.

Figure 10 shows the embedding of the *Planaria* single-cell RNA-sequencing data (Plass et al., 2018), while Tables 1 and 2 present the performance metrics for the synthetic experiments, including the percentage of reduction in tree-length over the Minimum Spanning Tree (RED) and computational efficiency measured by CPU wall time.

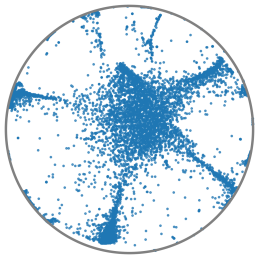


Figure 10: Planaria dataset.

	NJ		HS		RHS (ours)		vs HS (RED)		vs NJ (RED)	
$\lvert P\rvert$	RED (↑)	CPU	RED (↑)	CPU	RED (↑)	CPU	p_{Welch}	p_{TOST}	p_{Welch}	p_{TOST}
10	$1.46_{\pm 1.12}$	$5.50_{\pm 0.03}$	$2.72_{\pm 0.97}$	$0.03_{\pm 0.01}$	$3.17_{\pm 1.59}$	$4.60_{\pm 3.40}$	.457	.183	**.013**	.867
20	$-2.60_{\pm 3.00}$	$5.68_{\pm 0.04}$	$1.83_{\pm 0.70}$	$0.06_{\pm 0.01}$	$3.15_{\pm 0.56}$	$9.87_{\pm 7.00}$	**<.001**	.863	**<.001**	1.000
30	$-1.70_{\pm 2.31}$	$5.86_{\pm 0.03}$	$2.12_{\pm 0.60}$	$0.10_{\pm 0.02}$	$3.03_{\pm 1.09}$	$12.65_{\pm 6.94}$	**.036**	.411	**<.001**	1.000
40	$-2.62_{\pm 1.35}$	$5.95_{\pm 0.04}$	$2.17_{\pm 0.71}$	$0.13_{\pm 0.02}$	$3.30_{\pm 0.75}$	$17.59_{\pm 6.97}$	**.003**	.652	**<.001**	1.000
50	$-4.37_{\pm 1.87}$	$6.14_{\pm 0.04}$	$2.26_{\pm 0.43}$	$0.17_{\pm 0.02}$	$3.08_{\pm 0.34}$	$26.21_{\pm 8.67}$	**<.001**	.157	**<.001**	1.000
60	$-3.38_{\pm 1.09}$	$6.29_{\pm 0.03}$	$2.56_{\pm 0.57}$	$0.20_{\pm 0.02}$	$2.72_{\pm 0.48}$	$26.95_{\pm 8.40}$	.506	**.001**$^{\approx}$	**<.001**	1.000
70	$-3.28_{\pm 1.48}$	$6.51_{\pm 0.04}$	$2.45_{\pm 0.54}$	$0.22_{\pm 0.01}$	$2.68_{\pm 0.66}$	$31.09_{\pm 11.84}$	.405	**.005**$^{\approx}$	**<.001**	1.000
80	$-3.89_{\pm 1.79}$	$6.77_{\pm 0.03}$	$2.43_{\pm 0.53}$	$0.26_{\pm 0.02}$	$3.07_{\pm 0.39}$	$38.55_{\pm 8.45}$	**.007**	.051	**<.001**	1.000
90	$-3.46_{\pm 1.65}$	$7.01_{\pm 0.04}$	$2.72_{\pm 0.50}$	$0.30_{\pm 0.02}$	$2.87_{\pm 0.57}$	$66.07_{\pm 30.15}$	.540	**.001**$^{\approx}$	**<.001**	1.000
100	$-3.77_{\pm 1.75}$	$7.24_{\pm 0.03}$	$2.45_{\pm 0.35}$	$0.33_{\pm 0.02}$	$2.84_{\pm 0.47}$	$70.31_{\pm 14.92}$	.051	**.002**$^{\approx}$	**<.001**	1.000
150	$-4.17_{\pm 1.10}$	$8.46_{\pm 0.03}$	$2.27_{\pm 0.20}$	$0.48_{\pm 0.02}$	$2.48_{\pm 0.22}$	$139.87_{\pm 49.03}$	**.039**	**<.001**$^{\approx}$	**<.001**	1.000
200	$-5.01_{\pm 1.40}$	$11.09_{\pm 0.06}$	$2.43_{\pm 0.26}$	$0.66_{\pm 0.04}$	$2.64_{\pm 0.23}$	$341.42_{\pm 96.02}$	.072	**<.001**$^{\approx}$	**<.001**	1.000

Table 1: Sampling $|P|$ points from a centered hyperbolic Gaussian $\mathcal{G}(0, 0.5)$. RED: reduction over MST (%). CPU: total CPU time (sec.). Significance columns report Welch t-test and TOST p-values (RHS vs. the indicated baseline, RED metric, n=10, ε=1.0). **Bold** indicates p<0.05; $^{\approx}$ marks TOST equivalence (90 % CI $\subseteq (-\varepsilon, +\varepsilon)$ and p_{TOST}<0.05).

	NJ		HS		RHS (ours)		vs HS (RED)		vs NJ (RED)	
$\lvert P\rvert$	RED (↑)	CPU	RED (↑)	CPU	RED (↑)	CPU	p_{Welch}	p_{TOST}	p_{Welch}	p_{TOST}
10	$40.49_{\pm 0.03}$	$5.52_{\pm 0.04}$	$17.75_{\pm 4.57}$	$0.02_{\pm 0.01}$	$28.33_{\pm 18.54}$	$25.70_{\pm 13.30}$	.110	.928	.068	.955
20	$40.10_{\pm 0.06}$	$5.73_{\pm 0.05}$	$14.35_{\pm 4.61}$	$0.05_{\pm 0.02}$	$40.01_{\pm 0.09}$	$61.94_{\pm 16.77}$	**<.001**	1.000	**.018**	**<.001**$^{\approx}$
30	$39.93_{\pm 0.08}$	$5.91_{\pm 0.04}$	$14.74_{\pm 5.75}$	$0.08_{\pm 0.06}$	$39.84_{\pm 0.13}$	$95.05_{\pm 27.50}$	**<.001**	1.000	.082	**<.001**$^{\approx}$
40	$39.70_{\pm 0.09}$	$6.06_{\pm 0.03}$	$16.63_{\pm 4.46}$	$0.09_{\pm 0.03}$	$39.71_{\pm 0.12}$	$119.86_{\pm 53.15}$	**<.001**	1.000	.836	**<.001**$^{\approx}$
50	$39.53_{\pm 0.06}$	$6.00_{\pm 0.02}$	$17.88_{\pm 5.29}$	$0.10_{\pm 0.04}$	$39.52_{\pm 0.05}$	$116.38_{\pm 45.92}$	**<.001**	1.000	.690	**<.001**$^{\approx}$
60	$39.22_{\pm 0.10}$	$6.37_{\pm 0.04}$	$14.95_{\pm 4.60}$	$0.11_{\pm 0.02}$	$39.36_{\pm 0.11}$	$87.99_{\pm 9.33}$	**<.001**	1.000	**.008**	**<.001**$^{\approx}$
70	$39.01_{\pm 0.16}$	$6.55_{\pm 0.05}$	$14.42_{\pm 4.27}$	$0.11_{\pm 0.03}$	$39.26_{\pm 0.05}$	$118.26_{\pm 24.23}$	**<.001**	1.000	**<.001**	**<.001**$^{\approx}$
80	$38.55_{\pm 0.11}$	$6.77_{\pm 0.04}$	$14.47_{\pm 4.29}$	$0.12_{\pm 0.03}$	$39.11_{\pm 0.14}$	$111.56_{\pm 12.72}$	**<.001**	1.000	**<.001**	**<.001**$^{\approx}$
90	$38.18_{\pm 0.18}$	$6.96_{\pm 0.06}$	$15.85_{\pm 3.70}$	$0.14_{\pm 0.03}$	$39.03_{\pm 0.08}$	$171.84_{\pm 59.68}$	**<.001**	1.000	**<.001**	**.016**$^{\approx}$
100	$37.50_{\pm 0.28}$	$7.26_{\pm 0.07}$	$13.91_{\pm 4.99}$	$0.13_{\pm 0.04}$	$38.94_{\pm 0.12}$	$183.57_{\pm 60.30}$	**<.001**	1.000	**<.001**	1.000
150	$24.02_{\pm 1.24}$	$8.48_{\pm 0.05}$	$13.22_{\pm 6.21}$	$0.16_{\pm 0.02}$	$38.47_{\pm 0.10}$	$205.34_{\pm 52.63}$	**<.001**	1.000	**<.001**	1.000
200	$-8.08_{\pm 1.09}$	$10.51_{\pm 0.06}$	$13.62_{\pm 4.43}$	$0.18_{\pm 0.03}$	$38.17_{\pm 0.17}$	$265.46_{\pm 80.59}$	**<.001**	1.000	**<.001**	1.000

Table 2: Sampling $|P|$ points from a mixture of 10 hyperbolic Gaussians $\mathcal{G}(\mu_{10,k}(1-10^{-10}), 0.1)$, $k \in \{1, \ldots, 10\}$. RED: reduction over MST (%). CPU: total CPU time (sec.). Significance columns report Welch t-test and TOST p-values (RHS vs. the indicated baseline, RED metric, n=10, ε=1.0). **Bold** indicates p<0.05; $^{\approx}$ marks TOST equivalence (90 % CI $\subseteq (-\varepsilon, +\varepsilon)$ and p_{TOST}<0.05).

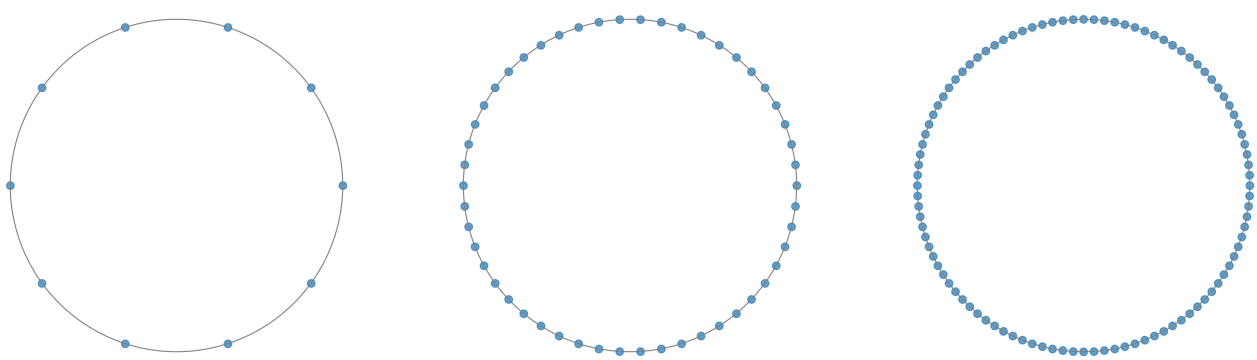
For each configuration, we assess statistical differences between RHS and each baseline via two complementary tests: the Welch–Satterthwaite two-sample t-test (Welch, 1938; Satterthwaite, 1946), which detects significant differences in means without assuming equal variances across groups, and the Two One-Sided Tests (TOST) procedure (Schuirmann, 1987), which declares *practical equivalence* when the 90% confidence interval of the mean difference lies within $[-\varepsilon, +\varepsilon]$ (in our case we choose ε=1.0). A significant Welch p-value with a non-significant TOST indicates a detectable difference; a significant TOST ($\approx$) with a non-significant Welch confirms practical equivalence; both significant implies a real but negligible gap; neither significant is inconclusive.

E.2 Near Boundary Performance Analysis

We present in Figure 11 some examples of the sampled data considered for the first experiment (*Approaching the Theoretical Limit*) of Section 5.2, and in Table 3 the corresponding results. Similarly, for the second experiment (*Characterizing the Transition Zone*), we illustrate in Figure 12 some datasets employed, and in Figure 13 the associated results.

The standard deviations reported in Table 3 reveal that RHS not only achieves higher reductions but also maintains low variance across random seeds; the exceptions are failures arising from numerical instability at the boundary, in which case the method does not return a valid solution. HS shows a different but related fragility: because it relies directly on Delaunay triangulations, it also becomes unstable near the boundary, but instead of terminating, it typically produces bad tree candidates. Stochastic expansion therefore contributes not only to solution quality but also to robustness, even though both heuristics remain sensitive to numerical issues in extreme boundary regimes. In contrast, NJ is less prompt to numerical instability in these regular polygon configurations near the boundary. Further discussion regarding the numerical instabilities at extreme boundary configurations can be found in Appendix E.3.

However, as highlighted in Figure 13, NJ performances degrade significatively when multiple clusters are introduced. For sufficiently high t (around 0.95 for RHS, 0.99 for HS, and already from 0.6 for NJ), and for a fixed number of clusters between 3 and 6, all three methods show improved RED as t increases. This trend also holds for RHS and HS when the number of clusters is higher. In contrast, NJ deteriorates sharply at the extreme boundary ($t = 1 - 10^{-10}$) when the number of clusters is more than 6. Moreover, for fixed high $t > 1 - 10^{-5}$, only RHS continues to benefit from an increasing number of clusters, while HS and NJ plateau or decline. In summary, RHS is the only method that consistently benefits both from being close to the boundary and from increasing the number of clusters. HS improves with boundary proximity, but saturates as the number of clusters grows. NJ initially improves when points are moderately close to the boundary, but its performance collapses in the extreme limit, showing that it cannot handle high curvature combined with clustering.



$|P| = 10$ $|P| = 50$ $|P| = 100$

Figure 11: Visualization of some of the datasets considered in the *Approaching the Theoretical Limit* experiment. Sampling $|P| \in \{10, 50, 100\}$ points from a mixture of hyperbolic Gaussians $\mathcal{G}(\mu_{|P|,k}(1 - 10^{-10}), 0.1)$ with $k \in \{1, \cdots, |P|\}$.

	NJ		HS		RHS (ours)	
$\lvert P\rvert$	RED (↑)	CPU	RED (↑)	CPU	RED (↑)	CPU
4	$31.31_{\pm 0.07}$	$5.54_{\pm 0.02}$	$31.34_{\pm 0.08}$	$0.01_{\pm 0.002}$	$31.39_{\pm 0.11}$	$1.59_{\pm 0.51}$
5	$34.86_{\pm 0.09}$	$5.58_{\pm 0.02}$	$23.26_{\pm 0.02}$	$0.01_{\pm 0.003}$	$34.94_{\pm 0.08}$	$10.26_{\pm 1.44}$
6	$36.95_{\pm 0.10}$	$5.60_{\pm 0.07}$	$18.28_{\pm 0.12}$	$0.01_{\pm 0.01}$	$37.01_{\pm 0.10}$	$15.46_{\pm 1.64}$
7	$38.30_{\pm 0.07}$	$5.59_{\pm 0.02}$	$17.61_{\pm 3.49}$	$0.01_{\pm 0.005}$	$38.33_{\pm 0.01}$	$13.09_{\pm 4.26}$
8	$39.26_{\pm 0.03}$	$5.56_{\pm 0.00}$	$10.81_{\pm 6.10}$	$0.01_{\pm 0.004}$	$39.28_{\pm 0.02}$	$24.27_{\pm 8.48}$
9	$39.95_{\pm 0.01}$	$5.59_{\pm 0.03}$	$13.05_{\pm 2.58}$	$0.01_{\pm 0.01}$	$39.96_{\pm 0.02}$	$52.79_{\pm 18.82}$
10	$40.48_{\pm 0.01}$	$5.59_{\pm 0.01}$	$11.60_{\pm 2.38}$	$0.02_{\pm 0.005}$	$40.51_{\pm 0.02}$	$35.03_{\pm 10.10}$
20	$42.48_{\pm 0.01}$	$5.76_{\pm 0.04}$	$20.86_{\pm 1.99}$	$0.17_{\pm 0.03}$	$42.59_{\pm 0.04}$	$92.29_{\pm 14.17}$
30	$42.93_{\pm 0.004}$	$5.84_{\pm 0.03}$	$21.04_{\pm 2.51}$	$0.31_{\pm 0.05}$	$43.03_{\pm 0.11}$	$89.08_{\pm 25.65}$
40	$43.10_{\pm 0.02}$	$6.01_{\pm 0.03}$	$17.56_{\pm 2.21}$	$0.33_{\pm 0.02}$	$43.29_{\pm 0.08}$	$159.69_{\pm 41.75}$
50	$43.16_{\pm 0.01}$	$6.21_{\pm 0.03}$	$20.07_{\pm 1.17}$	$0.50_{\pm 0.02}$	$43.26_{\pm 0.06}$	$217.99_{\pm 52.70}$
60	$43.14_{\pm 0.01}$	$6.40_{\pm 0.03}$	$18.80_{\pm 1.16}$	$0.58_{\pm 0.03}$	$43.26_{\pm 0.11}$	$214.50_{\pm 64.05}$
70	$43.12_{\pm 0.003}$	$6.63_{\pm 0.02}$	$18.84_{\pm 1.01}$	$0.63_{\pm 0.06}$	$43.39_{\pm 0.04}$	$287.02_{\pm 29.79}$
80	$43.09_{\pm 0.01}$	$6.81_{\pm 0.04}$	$17.09_{\pm 1.30}$	$0.75_{\pm 0.06}$	$43.09_{\pm 0.21}$	$239.36_{\pm 76.02}$
90	$43.05_{\pm 0.01}$	$7.03_{\pm 0.01}$	$18.65_{\pm 0.75}$	$0.83_{\pm 0.04}$	$43.15_{\pm 0.13}$	$304.83_{\pm 106.17}$
100	$43.02_{\pm 0.009}$	$7.20_{\pm 0.03}$	$18.00_{\pm 0.77}$	$0.80_{\pm 0.01}$	$42.99_{\pm 0.11}$	$310.98_{\pm 54.61}$

Table 3: Sampling $|P|$ points from a mixture of hyperbolic Gaussians $\mathcal{G}(\mu_{|P|,k}(1-10^{-10}), 0.1)$, $k \in \{1, \cdots, |P|\}$ near the boundary of the hyperbolic disk with only one point sampled per Gaussian. RED: reduction over MST (%). CPU: total CPU time (sec.).

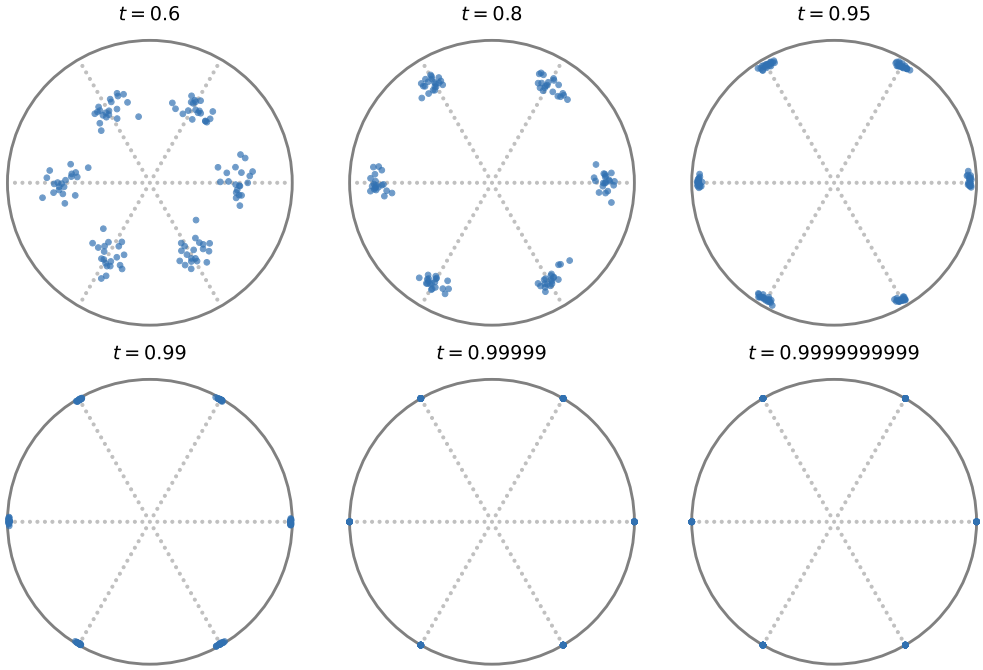


Figure 12: Visualization of the data sets considered in the *Characterizing the Transition Zone* experiment for the $d = 6$ case. Sampling 20 points from each hyperbolic Gaussian $\mathcal{G}(\mu_{6,k}(t), 0.1)$ with $k \in \{1, \cdots, 6\}$, and $t \in \{0.6, 0.8, 0.95, 0.99, 1-10^{-5}, 1-10^{-10}\}$.

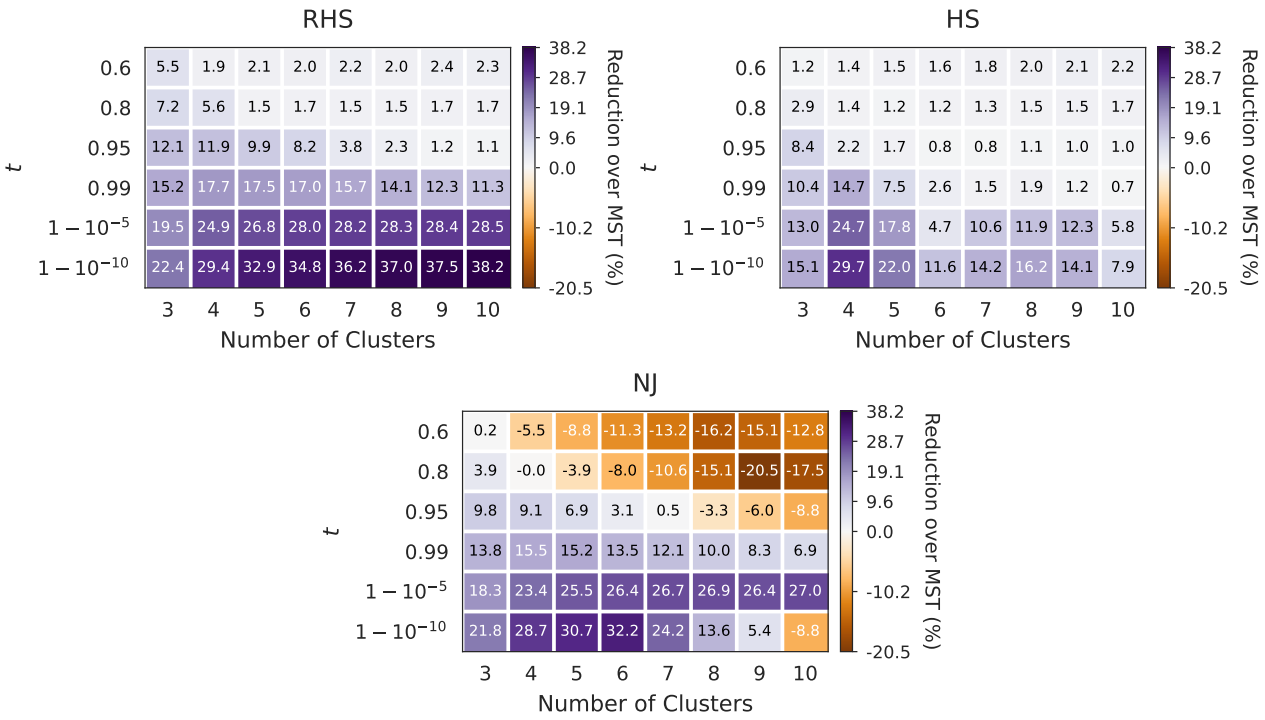


Figure 13: Convergence analysis for a mixture of $\mathcal{G}(\mu_{d,k}(t), 0.1)$, $k \in \{1, \cdots, d\}$ for $d \in \{3, \cdots, 10\}$ and with an increasing t, sampling 20 points per Gaussian. Values correspond to reduction over MST (%).

E.3 Discussion on Near Boundary Numerical Stability

As reported in Appendix E.2, numerical instabilities arise only in extreme near-boundary regimes, for example when $t = 1 - 10^{-10}$. This setting is included solely as a theoretical stress test for the upper-bound ratio and is not relevant in practical applications. For less extreme values, such as $t \geq 1 - 10^{-5}$, we do not observe instability.

Indeed, as also noted by Mishne et al. (2023), such numerical effects depend in part on the choice of hyperbolic model. We adopt the Klein–Beltrami model because it simplifies the computation of Hyperbolic Delaunay Triangulations and isoptic curves (García-Castellanos et al., 2025, Appendix A.2). When additional numerical robustness is required, the method could instead be implemented in the hyperboloid model. We leave this variation to future work.

E.4 Discussion on Scalability and Computational Complexity

Runtime comparison on synthetic experiments. As shown in Tables 1, 2, and 3, the proposed method consistently achieves superior performance compared to both baselines. While its computational cost exceeds that of NJ and HS, this is expected for non-deterministic heuristics designed to compute Steiner Minimal Trees. To contextualize our computational requirements, our method requires approximately 187 seconds for 100 points (averaged across all configurations). In comparison, other non-deterministic approaches report varied costs: among DT-based methods, Beasley and Goffinet (1994) requires 488.4 seconds for 100 points and Laarhoven and Ohlmann (2011) requires 40.2 seconds for 100 points, while Yang (2006) report that their evolutionary algorithm requires over 5,000 seconds for 100 points. Our method thus falls within the lower range of computational costs for non-deterministic heuristics in this domain.

Estimating the computational complexity. As stated in Appendix B and Appendix D.2, HyperSteiner has a computational complexity of $\mathcal{O}(|P| \log |P|)$ (García-Castellanos et al., 2025), while Neighbor Joining has a complexity of $\mathcal{O}(|P|^3)$ in its standard form (Studier and Keppler, 1988), and improved versions achieve $\mathcal{O}(|P|^2)$ or $\mathcal{O}(|P|^{3/2} \log |P|)$ (respectively (Elias and Lagergren, 2009; Price et al., 2009)). As mentioned above, due to its stochastic aspect, we cannot compute a theoretical complexity for Randomized HyperSteiner, similar to the

situation with its Euclidean stochastic analogs.

Following the methodology of Beasley and Goffinet (1994), we compute instead an empirical complexity by modeling the CPU runtime as a function of the number of points $|P|$, using a parametric form fitted via least-squares on log-transformed data. We focus on the results from Table 1, where the hyperbolic Gaussian sampling is locally Euclidean, allowing for rough comparison with Beasley & Goffinet's empirical finding of $\mathcal{O}(|P|^{2.19})$ for their Euclidean stochastic method. In our case, the average computational time of Randomized HyperSteiner is empirically estimated to be $\mathcal{O}(|P|^{2.1}\log|P|)$ or $\mathcal{O}(|P|^{2.28})$. This confirms that our method is computationally more expensive than the deterministic baselines, as expected for stochastic heuristics, while remaining in the same complexity range as Euclidean stochastic approaches.

Equation 7 computational complexity. Equation (7) is solved via MINPACK's *hybrd* and *hybrj* algorithms (implemented with scipy.optimize.fsolve in the Python library) which are modifications of Powell's hybrid method. The latter is a version of Newton's method with a trust-region approach, and its computational complexity depends on the accuracy requested and the number of iterations needed to converge, which itself depends on the quality of the initialization. For one iteration of these algorithms, since we have a system of two (nonlinear) equations, the number of arithmetic operations (according to MINPACK's documentation from Burton et al., 1980) is about $11.5 \times 2^2 + 1.3 \times 2^3 = 56.4$ in addition with the number of operations needed to evaluate the functions $\varphi_{x,y,\frac{2\pi}{3}}(s)$ and $\varphi_{y,z,\frac{2\pi}{3}}(s)$. Thus, the computational cost per iteration is constant time. A reasonable value for the maximal number of iterations (given also by the MINPACK documentation) is $100\times(2+1) = 300$. In summary, for a system of two equations in two dimensions, the computational complexity of solving Equation (7) is, essentially, $\mathcal{O}(1)$.

Compute-performance trade-off. These results present practitioners with a clear trade-off: RHS should be selected when maximizing solution quality is the priority, while the vanilla HyperSteiner remains suitable for applications requiring fast approximations. Notably, as demonstrated in Section 5.2, on configurations approaching the boundary, the proposed method achieves performance improvements of 20% to 40% over the MST, as well as significant gains over the other baselines. Under such configurations, these substantial improvements justify the additional computational cost.

E.5 Ablation Study: Insertion probabilities

As stated in Appendix A.1, for the insertion probability range $[l, u]$ of our *Expansion phase* in Algorithm 1, we chose the interval $[0.3, 0.6]$ following Laarhoven and Ohlmann (2011). We provide in Tables 4 and 5 an ablation study on the probability insertion p for a typical configuration in the hyperbolic space, that is when sampling $|P|$ points from a centered hyperbolic Gaussian. Taken together, these two tables suggest that choosing p between 0.3 and 0.6 provides a good trade-off between maximizing RED and minimizing computational cost.

$\|P\|$	$p=0$	$p=0.1$	$p=0.2$	$p=0.3$	$p=0.4$	$p=0.5$	$p=0.6$	$p=0.7$	$p=0.8$	$p=0.9$	$p=1$
10	$3.03_{\pm 1.42}$	$3.30_{\pm 1.32}$	$2.02_{\pm 0.97}$	$3.29_{\pm 1.72}$	$\mathbf{3.76_{\pm 0.94}}$	$3.53_{\pm 1.78}$	$3.07_{\pm 1.16}$	$3.45_{\pm 1.89}$	$1.68_{\pm 1.02}$	$\mathbf{3.95_{\pm 1.87}}$	$3.26_{\pm 1.28}$
20	$2.81_{\pm 1.14}$	$2.35_{\pm 0.84}$	$2.79_{\pm 0.68}$	$\mathbf{3.41_{\pm 1.26}}$	$3.18_{\pm 0.79}$	$3.34_{\pm 0.86}$	$\mathbf{3.55_{\pm 1.30}}$	$2.53_{\pm 0.85}$	$2.87_{\pm 0.45}$	$2.88_{\pm 0.68}$	$2.60_{\pm 0.64}$
30	$2.60_{\pm 0.75}$	$2.69_{\pm 0.80}$	$2.63_{\pm 0.90}$	$2.23_{\pm 0.52}$	$2.66_{\pm 0.80}$	$2.83_{\pm 0.58}$	$\mathbf{2.89_{\pm 0.73}}$	$\mathbf{2.94_{\pm 0.61}}$	$2.84_{\pm 0.81}$	$2.81_{\pm 0.84}$	$2.65_{\pm 0.85}$
40	$2.17_{\pm 0.56}$	$2.72_{\pm 0.89}$	$2.89_{\pm 0.74}$	$\mathbf{2.96_{\pm 0.70}}$	$2.50_{\pm 0.35}$	$2.81_{\pm 0.72}$	$\mathbf{2.98_{\pm 0.56}}$	$2.93_{\pm 0.66}$	$2.71_{\pm 0.50}$	$2.69_{\pm 0.51}$	$2.37_{\pm 0.67}$
50	$2.24_{\pm 0.59}$	$2.83_{\pm 0.65}$	$2.80_{\pm 0.71}$	$\mathbf{3.36_{\pm 0.46}}$	$2.59_{\pm 0.39}$	$3.10_{\pm 0.65}$	$3.08_{\pm 0.65}$	$2.74_{\pm 0.51}$	$2.90_{\pm 0.84}$	$\mathbf{3.29_{\pm 0.74}}$	$2.54_{\pm 0.57}$
60	$2.08_{\pm 0.53}$	$\mathbf{3.02_{\pm 0.53}}$	$2.94_{\pm 0.63}$	$2.72_{\pm 0.46}$	$2.70_{\pm 0.57}$	$2.90_{\pm 0.56}$	$2.77_{\pm 0.58}$	$2.77_{\pm 0.26}$	$\mathbf{3.00_{\pm 0.55}}$	$2.88_{\pm 0.56}$	$2.84_{\pm 0.55}$
70	$2.20_{\pm 0.52}$	$2.86_{\pm 0.62}$	$2.43_{\pm 0.58}$	$\mathbf{3.25_{\pm 0.60}}$	$2.82_{\pm 0.67}$	$2.97_{\pm 0.54}$	$2.43_{\pm 0.64}$	$2.78_{\pm 0.57}$	$\mathbf{3.01_{\pm 0.57}}$	$2.85_{\pm 0.57}$	$2.57_{\pm 0.64}$
80	$2.21_{\pm 0.59}$	$2.98_{\pm 0.49}$	$2.81_{\pm 0.69}$	$2.88_{\pm 0.63}$	$2.77_{\pm 0.69}$	$\mathbf{3.00_{\pm 0.51}}$	$2.81_{\pm 0.51}$	$2.73_{\pm 0.69}$	$\mathbf{3.17_{\pm 0.51}}$	$2.97_{\pm 0.63}$	$2.64_{\pm 0.62}$
90	$2.27_{\pm 0.50}$	$2.95_{\pm 0.51}$	$2.88_{\pm 0.56}$	$2.89_{\pm 0.52}$	$2.81_{\pm 0.51}$	$2.88_{\pm 0.58}$	$\mathbf{3.18_{\pm 0.57}}$	$2.88_{\pm 0.48}$	$2.86_{\pm 0.53}$	$\mathbf{2.95_{\pm 0.45}}$	$2.91_{\pm 0.44}$
100	$2.29_{\pm 0.45}$	$2.71_{\pm 0.59}$	$\mathbf{2.98_{\pm 0.46}}$	$2.79_{\pm 0.41}$	$2.82_{\pm 0.40}$	$2.66_{\pm 0.47}$	$2.65_{\pm 0.42}$	$2.91_{\pm 0.47}$	$2.82_{\pm 0.36}$	$\mathbf{2.95_{\pm 0.42}}$	$2.67_{\pm 0.52}$
Avg	$2.39_{\pm 0.70}$	$2.84_{\pm 0.72}$	$2.72_{\pm 0.69}$	$2.98_{\pm 0.73}$	$2.86_{\pm 0.61}$	$3.00_{\pm 0.72}$	$2.94_{\pm 0.71}$	$2.87_{\pm 0.70}$	$2.79_{\pm 0.61}$	$3.02_{\pm 0.73}$	$2.70_{\pm 0.68}$

Table 4: Randomized HyperSteiner RED results when sampling $|P|$ points from a Hyperbolic Centered Gaussian (mean $\pm$ standard deviation) for insertion probabilities p between 0 and 1. The two best results for each row are in bold.

$\|P\|$	$p=0$	$p=0.1$	$p=0.2$	$p=0.3$	$p=0.4$	$p=0.5$	$p=0.6$	$p=0.7$	$p=0.8$	$p=0.9$	$p=1$
10	$8.25_{\pm 4.89}$	$8.46_{\pm 6.28}$	$9.38_{\pm 6.38}$	$6.51_{\pm 5.42}$	$5.03_{\pm 2.93}$	$8.07_{\pm 5.13}$	$7.93_{\pm 5.67}$	$5.55_{\pm 4.69}$	$3.62_{\pm 2.36}$	$6.21_{\pm 4.38}$	$5.42_{\pm 4.68}$
20	$21.22_{\pm 18.91}$	$21.34_{\pm 17.15}$	$12.56_{\pm 11.00}$	$12.96_{\pm 5.43}$	$12.08_{\pm 4.61}$	$8.38_{\pm 3.88}$	$12.43_{\pm 7.75}$	$12.82_{\pm 8.40}$	$7.37_{\pm 2.10}$	$12.19_{\pm 8.40}$	$25.15_{\pm 46.83}$
30	$23.10_{\pm 21.33}$	$15.49_{\pm 7.27}$	$12.30_{\pm 6.21}$	$11.74_{\pm 6.01}$	$11.15_{\pm 6.21}$	$12.85_{\pm 5.29}$	$11.45_{\pm 6.04}$	$14.40_{\pm 7.32}$	$15.62_{\pm 4.68}$	$17.78_{\pm 8.55}$	$17.07_{\pm 7.11}$
40	$20.89_{\pm 18.82}$	$22.11_{\pm 7.50}$	$15.97_{\pm 7.03}$	$13.71_{\pm 5.23}$	$16.16_{\pm 8.94}$	$14.51_{\pm 6.36}$	$18.70_{\pm 7.59}$	$20.33_{\pm 4.27}$	$25.16_{\pm 11.66}$	$22.72_{\pm 7.91}$	$24.55_{\pm 7.50}$
50	$35.75_{\pm 32.54}$	$20.79_{\pm 6.46}$	$17.66_{\pm 8.74}$	$16.15_{\pm 5.77}$	$20.57_{\pm 10.73}$	$17.53_{\pm 5.78}$	$24.03_{\pm 11.60}$	$35.58_{\pm 10.01}$	$43.23_{\pm 11.14}$	$69.53_{\pm 15.59}$	$74.89_{\pm 28.67}$
60	$28.92_{\pm 26.24}$	$26.41_{\pm 8.73}$	$17.74_{\pm 5.68}$	$18.37_{\pm 7.40}$	$21.36_{\pm 4.92}$	$24.06_{\pm 9.25}$	$28.27_{\pm 6.51}$	$36.43_{\pm 13.48}$	$67.70_{\pm 31.42}$	$73.89_{\pm 22.90}$	$109.33_{\pm 33.30}$
70	$55.84_{\pm 71.78}$	$28.26_{\pm 8.94}$	$19.50_{\pm 7.85}$	$20.67_{\pm 6.88}$	$23.42_{\pm 5.67}$	$36.56_{\pm 20.86}$	$40.92_{\pm 9.56}$	$63.02_{\pm 20.79}$	$85.85_{\pm 23.53}$	$134.40_{\pm 39.08}$	$226.67_{\pm 82.49}$
80	$60.66_{\pm 109.63}$	$30.58_{\pm 12.82}$	$21.00_{\pm 5.69}$	$21.98_{\pm 7.64}$	$30.11_{\pm 8.40}$	$30.95_{\pm 9.33}$	$48.54_{\pm 17.12}$	$78.43_{\pm 18.81}$	$123.10_{\pm 41.94}$	$210.40_{\pm 67.90}$	$293.06_{\pm 75.58}$
90	$80.30_{\pm 120.06}$	$32.09_{\pm 13.54}$	$33.62_{\pm 12.13}$	$33.49_{\pm 6.74}$	$38.06_{\pm 11.01}$	$44.73_{\pm 12.22}$	$76.83_{\pm 18.01}$	$135.59_{\pm 38.33}$	$261.36_{\pm 70.28}$	$429.71_{\pm 87.88}$	$917.67_{\pm 349.84}$
100	$29.24_{\pm 13.20}$	$39.82_{\pm 8.82}$	$30.29_{\pm 10.15}$	$41.65_{\pm 17.64}$	$43.38_{\pm 7.20}$	$53.57_{\pm 10.77}$	$121.44_{\pm 38.52}$	$195.24_{\pm 63.01}$	$326.40_{\pm 102.91}$	$925.18_{\pm 211.66}$	$1936.48_{\pm 524.08}$
Avg	$36.42_{\pm 43.74}$	$24.54_{\pm 9.75}$	$19.00_{\pm 8.09}$	$19.72_{\pm 7.42}$	$22.13_{\pm 7.06}$	$25.12_{\pm 8.89}$	$39.05_{\pm 12.84}$	$59.74_{\pm 18.91}$	$95.94_{\pm 30.20}$	$190.20_{\pm 47.42}$	$363.03_{\pm 116.01}$

Table 5: Randomized HyperSteiner CPU time (seconds) when sampling $|P|$ points from a Hyperbolic Centered Gaussian (mean $\pm$ standard deviation) for insertion probabilities p between 0 and 1.

E.6 Ablation Study: Expansion schedule

As we show in Appendix A.1, there are several choices for how many times we expand the Delaunay triangulation at each step. In Figure 7, we compare three approaches: "linear" ($n \mapsto n$) (Beasley and Goffinet, 1994), "constant" ($n \mapsto 1$), and ours, which we call "sqrt" ($n \mapsto \lfloor 2\sqrt{n} - 1 \rfloor$). While we provide justifications for selecting the "sqrt" schedule in Appendix A.1, we will perform an ablation study in this section to further support this choice.

Hence, we provide in Table 6 a performance comparison on a mixture of d hyperbolic Gaussians, each one centered at a vertex of a regular d-gon, with radial parameter $t = 0.9$, and a standard deviation of 0.01. We sample 20 points per Gaussian, which leads to d clusters. This dataset resembles the ones of Section 5.2 for our *Near Boundary Performance Analysis*, making it a suitable benchmark for our purposes.

As shown in Table 6, the "linear" schedule can be more expressive than the "sqrt" one, but its computational cost increases significantly as the number of clusters grows. Therefore, in line with our justification in Appendix A.1, we adopt the "sqrt" schedule in our experiments as it provides the highest performance with a reasonable computational cost. Nevertheless, in practice, the "constant" schedule could be employed to improve computational efficiency without a significant performance drop.

	sqrt		constant		linear	
d	RED (↑)	CPU	RED (↑)	CPU	RED (↑)	CPU
4	$\mathbf{16.75}_{\pm 0.05}$	$14.80_{\pm 6.98}$	$16.62_{\pm 0.01}$	$5.91_{\pm 1.00}$	$16.68_{\pm 0.06}$	$23.05_{\pm 7.42}$
5	$\mathbf{13.68}_{\pm 0.07}$	$36.45_{\pm 15.14}$	$13.61_{\pm 0.12}$	$15.41_{\pm 3.82}$	$13.65_{\pm 0.11}$	$59.65_{\pm 30.82}$
6	$10.10_{\pm 0.12}$	$77.60_{\pm 5.44}$	$10.06_{\pm 0.15}$	$66.85_{\pm 20.40}$	$\mathbf{10.21}_{\pm 0.05}$	$118.69_{\pm 16.79}$
7	$\mathbf{6.97}_{\pm 0.02}$	$172.71_{\pm 66.49}$	$6.93_{\pm 0.19}$	$113.59_{\pm 28.29}$	$6.85_{\pm 0.07}$	$286.39_{\pm 66.62}$
8	$3.76_{\pm 0.68}$	$295.45_{\pm 79.41}$	$2.49_{\pm 0.31}$	$200.71_{\pm 22.70}$	$\mathbf{3.97}_{\pm 0.36}$	$397.98_{\pm 155.61}$
9	$1.41_{\pm 0.40}$	$243.90_{\pm 54.16}$	$1.11_{\pm 0.88}$	$171.49_{\pm 54.16}$	$\mathbf{1.51}_{\pm 0.40}$	$739.33_{\pm 486.84}$

Table 6: Ablation of the expansion schedule of the Randomized HyperSteiner, for a mixture of $\mathcal{G}(\mu_{d,k}(0.9), 0.01)$, with $k \in \{1, \cdots, d\}$ for $d \in \{4, \cdots, 10\}$, sampling 20 points per Gaussian.

F Additional Related Work on Hyperbolic Machine Learning

Hyperbolic geometry has emerged as a powerful tool in machine learning, particularly for modeling data with latent hierarchical structure. Early work focused on embedding techniques that leverage manifold learning principles (Nickel and Kiela, 2017; Chamberlain et al., 2017; Sala et al., 2018; Chami et al., 2021; Guo et al., 2022). Following the introduction of hyperbolic neural networks (Ganea et al., 2018), subsequent work explored hyperbolic generative modeling (Mathieu et al., 2019; Nagano et al., 2019; Bose et al., 2020; Huang et al., 2022). Recent developments have explored large-scale architectures, including hyperbolic large language models (He et al., 2025a,b) enabled by adapting the Transformer architecture (Vaswani et al., 2017) to hyperbolic geometry (Gulcehre et al., 2019; Chen et al., 2022; Shimizu et al., 2021; Yang et al., 2024).

These developments are motivated by hyperbolic geometry's representational advantages for hierarchical data, where exponential growth patterns can be embedded with arbitrarily low distortion compared to Euclidean space (Sarkar, 2011). However, while existing hyperbolic machine learning models effectively capture hierarchical structure in latent form, they do not inherently provide mechanisms for explicitly extracting the underlying hierarchy. Our work addresses this gap by building upon the García-Castellanos et al. (2025) framework to enable direct recovery of minimal trees from hyperbolic embeddings, supporting what may be broadly termed *hyperbolic geometric inference*. Related efforts, such as Medbouhi et al. (2024), also fall under this paradigm by enabling statistical data analysis in hyperbolic latent spaces.