Hyperbolic Random Forests

Anonymous authors
Paper under double-blind review

Abstract

Hyperbolic space is becoming a popular choice for representing data due to the hierarchical structure — whether implicit or explicit — of many real-world datasets. Along with it comes a need for algorithms capable of solving fundamental tasks, such as classification, in hyperbolic space. Recently, multiple papers have investigated hyperbolic alternatives to hyperplane-based classifiers, such as logistic regression and SVMs. While effective, these approaches struggle with more complex hierarchical data. We, therefore, propose to generalize the well-known random forests to hyperbolic space. We do this by redefining the notion of a split using horospheres. Since finding the globally optimal split is computationally intractable, we find candidate horospheres through a large-margin classifier. To make hyperbolic random forests work on multi-class data and imbalanced experiments, we furthermore outline new methods for combining classes based on the lowest common ancestor and class-balanced large-margin losses. Experiments on standard and new benchmarks show that our approach outperforms both conventional random forest algorithms and recent hyperbolic classifiers.

1 Introduction

Machine learning in hyperbolic space is gaining more and more attention, and hyperbolic representations of data have already found success in numerous domains, such as natural language processing (Nickel & Kiela, 2017; Tai et al., 2022) computer vision (Ahmad & Lecue, 2022; Khrulkov et al., 2020; Ghadimi Atigh et al., 2021), graphs (Chami et al., 2019; Liu et al., 2019; Sun et al., 2021b), recommender systems (Sun et al., 2021a), and more. Hyperbolic space is a natural choice for data with a hierarchical structure due to the fact that the available space grows exponentially when moving away from the origin. Therefore, it can be seen as a continuous version of a graph-theoretical tree (Nickel & Kiela, 2018).

With the rise of datasets embedded in hyperbolic space comes a need for algorithms that can successfully operate on them (Cho et al., 2019). As a result, many methods specifically designed for hyperbolic space have been proposed that tackle a variety of machine learning tasks, such as clustering (Monath et al., 2019), regression (Marconi et al., 2020), and classification (Ganea et al., 2018b; Chien et al., 2021; Cho et al., 2019; Pan et al., 2023; Weber et al., 2020).

Current hyperbolic classification algorithms, such as hyperbolic support vector machines (Fan et al., 2023) or hyperbolic logistic regression (Ganea et al., 2018b), have shown promising results, but still struggle with complex datasets. In Euclidean space, there is a long history of success of tree-based random forest algorithms in this regard, and they remain among the most popular learning algorithms for many data types (Caruana & Niculescu-Mizil, 2006; Fernández-Delgado et al., 2014). We argue that decision trees and, by extension, random forests are well-suited for hyperbolic space due to their shared hierarchical structure. Figure 1 illustrates the need for a classifier that fits the underlying geometry and the benefits of a hyperbolic tree-based classifier. Euclidean hyperplane splits (Figure 1(a/b)) are ineffective at capturing the structure of the data. In contrast, our hyperbolic splits (Figure 1(c)) are more appropriate, where the nested splits clearly show how the smallest green class is a subclass of the larger orange one.

Tree-based algorithms are built by recursively applying a splitting function; hence, to generalize the concept of a tree-based classifier to hyperbolic space, we need to define a hyperbolic splitting function. For this

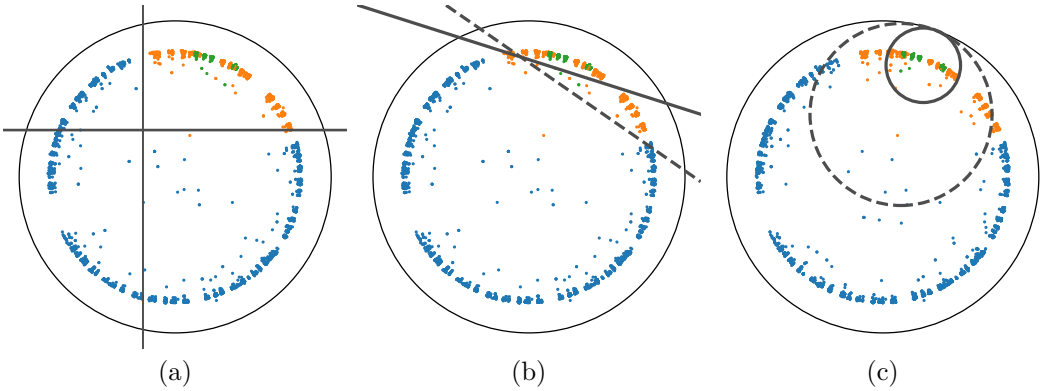


Figure 1: **Motivation for hyperbolic random forests.** Two splits of (a) an axis-aligned Euclidean, (b) an oblique Euclidean, and (b) a hyperbolic decision tree of depth two. The data is a continuous embedding of a tree split into three nested classes. The inductive biases of linear decision boundaries are inappropriate for efficiently capturing the underlying geometry, and more splits are needed to be effective.

purpose, we propose to use *horospheres*, which share several desirable properties with the hyperplanes used in Euclidean trees. Due to a combinatorial explosion, finding the optimal horosphere by enumeration is computationally intractable. We propose to tackle data splitting as a binary horosphere classification task. To deal with multiple classes in a hierarchically consistent manner, we extend our splitting operator using hyperclasses. We show how to obtain hyperclasses through clustering in hyperbolic space with the lowest common ancestor. We also outline a class-balancing large-margin loss for dealing with long-tailed data.

We extensively evaluate our approach on canonical benchmarks from previous works on hyperbolic classification. Additionally, we introduce three new multi-class classification experiments. We show that our method is superior to both competing hyperbolic classifiers as well as their Euclidean counterparts in these settings. Summarized, our contributions are (1) a generalization of random forests to hyperbolic space using horospheres, dubbed HoroRF, (2) two extensions to enable effective learning in multi-class and imbalanced settings, and (3) a thorough evaluation of HoroRF, showing its advantage over other methods, both Euclidean and hyperbolic.

2 Related Work

2.1 Hyperbolic machine learning

Machine learning in hyperbolic space has gained traction due to its inherent hierarchical and compact nature. Foundational work showed that hyperbolic space is superior to Euclidean space for embedding hierarchies in a continuous space, allowing for embeddings with minimal distortion (Nickel & Kiela, 2017; Ganea et al., 2018a). Empowered by these results, learning with hyperbolic embeddings has recently been successfully used for various problems. Chami et al. (2019) and Liu et al. (2019) showed how to generalize graph networks to hyperbolic space, while Tifrea et al. (2019) showed the potential of hyperbolic word embeddings. In the visual domain, hyperbolic embeddings have been shown to improve image segmentation (Ghadimi Atigh et al., 2022), zero-shot recognition (Liu et al., 2020), image-text representation learning (Desai et al., 2023), and more. Hyperbolic embeddings have also been effective for biology (Klimovskaia et al., 2020) and in recommender systems (Sun et al., 2021a). We refer to recent surveys for a more complete overview (Peng et al., 2021; Yang et al., 2022; Mettes et al., 2023).

For classification specifically, various traditional classifiers such as logistic regression (Ganea et al., 2018b), neural networks (Ganea et al., 2018b; Shimizu et al., 2021; van Spengler et al., 2023), and support vector machines (SVM) (Cho et al., 2019; Fan et al., 2023; Fang et al., 2021; Weber et al., 2020) now have counterparts in hyperbolic space. In this work, we strive to go beyond single hyperplane/gyroplane decision boundaries per class and bring random forests to hyperbolic space.

Our method is built on the concept of horospheres, which are the level sets of the Busemann function to ideal points. Ideal points have recently succeeded in multiple areas, such as supervised classification (Ghadimi Atigh et al., 2021) and self-supervised learning (Durrant & Leontidis, 2023). Horospheres, in particular, have found applications in dimensionality reduction (Chami et al., 2021), as well as generalizing SVMs (Fan et al., 2023) and neural networks (Sonoda et al., 2022) to hyperbolic space, but we are the first to use them as building blocks for random forests. Concurrent with our work, Chlenski et al. (2023) also propose a hyperbolic random forest model where they rely on geodesics rather than horospheres and work in the hyperboloid model of hyperbolic space instead of the Poincaré disk model.

2.2 Random Forests

Random Forests remain a popular choice of classifier to this day due to their high performance, speed, and insensitivity to hyperparameters (Grinsztajn et al., 2022). In their original paper (Breiman, 2001), two versions of random forests were proposed: one with axis-aligned splits based on a single feature and one with oblique splits based on linear combinations of features. While the optimal axis-aligned split can be found by exhaustive search, finding the optimal oblique split is NP-complete (Heath et al., 1993). As a result, many works have developed heuristics to find good oblique splits in a reasonable time.

One line of work makes use of meta-heuristics such as hill climbing (Murthy et al., 1993), simulated annealing (Heath et al., 1993), or genetic algorithms (Cantú-Paz & Kamath, 2003). Other approaches train one or more binary classifiers at every node, choosing among the resulting hyperplanes to split the data. Examples include using linear discriminant analysis, ridge regression (Menze et al., 2011), or support vector machines (Do et al., 2010). For multi-class cases, typically, heuristics are employed to partition them into two hyperclasses (Katuwal & Suganthan, 2018). We follow the binary classifier approach for our hyperbolic random forests and introduce a hyperclass heuristic specifically designed for hyperbolic space.

3 Hyperbolic Random Forests

3.1 The Poincaré ball model

We follow the convention from previous works using horospheres (Chami et al., 2021; Fan et al., 2023) and hyperbolic machine learning more broadly (Ganea et al., 2018b; Khrulkov et al., 2020; Chien et al., 2021; Shimizu et al., 2021) and make use of the Poincaré ball model of hyperbolic space. The Poincaré ball model is defined by the metric space $(\mathbb{B}_c^n, g_c^{\mathbb{B}})$ for a given negative curvature c, which we set to 1 throughout this work, where

$$\mathbb{B}_1^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < 1\} \tag{1}$$

with $\|\cdot\|$ the standard Euclidean norm, and

$$g_1^{\mathbb{B}}(\mathbf{x}) = \frac{2}{1 - \|\mathbf{x}\|^2}\mathbf{I}_n. \tag{2}$$

From here on, we will omit the curvature subscript for clarity. The distance between two points is given by

$$d_{\mathbb{B}}(\mathbf{a}, \mathbf{b}) = \operatorname{arcosh}\left(1 + 2\frac{\|\mathbf{a} - \mathbf{b}\|^2}{(1 - \|\mathbf{a}\|^2)(1 - \|\mathbf{b}\|^2)}\right), \tag{3}$$

which is the length of the geodesic arc connecting them. Extending geodesics to infinity in one direction leads to a point on the boundary of the Poincaré ball. These points on the boundary are known as *ideal points*. The set of all ideal points thus lies on the hypersphere $\mathbb{S}^{n-1}$, and they can be interpreted as directions in hyperbolic space (Chami et al., 2021).

3.2 HoroRF

To generalize random forests to hyperbolic space, we require a way to split data points recursively for any number of classes, regardless of the class frequency distribution, into two partitions with low impurity. In Euclidean space, this splitting function outputs hyperplanes to split the data. For hyperbolic random forests, we propose to use horospheres.

3.2.1 Formalization

Horospheres are the level sets of the Busemann function (Busemann, 1955), which calculates the normalized distance to infinity in a given direction. It can be expressed in closed form in the Poincaré model:

$$B_{\mathbf{w}}(\mathbf{x}) = \log \frac{\|\mathbf{w} - \mathbf{x}\|^2}{1 - \|\mathbf{x}\|^2}. \tag{4}$$

As a result, horospheres $\pi_{\mathbf{w},b}$ are parameterized by an ideal point $\mathbf{w} \in \mathbb{S}^{n-1}$, and a distance to that point $b \in \mathbb{R}$, which is the radius of the horosphere. Our goal is to find the optimal horosphere $\pi'_{\mathbf{w},b}$ that minimizes the impurity of the resulting partitions or, equivalently, maximizes the information gain. Consider a node S with data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with $\mathbf{x} \in \mathbb{B}^n$ and $y \in Y$. The optimal horosphere is given by

$$\pi'_{\mathbf{w},b} = \underset{\pi_{\mathbf{w},b}}{\arg\max}(H(S) - I_{in} - I_{out}), \tag{5}$$

where H is an impurity measure, and I_{in} and I_{out} give the impurity of the set of points inside and outside of the horosphere, respectively:

$$\begin{aligned} I_{in} &= \frac{N_{in}}{N} H(\{\mathbf{x} \in S | B_{\mathbf{w}}(\mathbf{x}) < b\}), \\ I_{out} &= \frac{N_{out}}{N} H(\{\mathbf{x} \in S | B_{\mathbf{w}}(\mathbf{x}) \geq b\}). \end{aligned} \tag{6}$$

Here, N_{in} and N_{out} denote the number of samples inside and outside the horosphere. Similar to oblique decision trees in Euclidean space, finding the globally optimal horosphere is computationally infeasible. As such, we need to find approximate solutions.

3.2.2 Finding approximate solutions

As is conventional in (oblique) decision tree literature (*e.g.*, Do et al. (2010); Menze et al. (2011); Katuwal et al. (2020)), we employ a binary classifier at every node to find candidate solutions. We need a hyperbolic classifier that will allow us to find splits that fit the underlying geometry. Large margin classifiers based on horospheres guarantee a globally optimal solution, bypassing hyperbolic hyperplane-based methods that fail to converge (Weber et al., 2020) or are limited to two dimensions (Chien et al., 2021). Moreover, the state-of-the-art hyperbolic SVM, HoroSVM, is based on horospheres and outperforms hyperbolic hyperplane-based methods (Fan et al., 2023). For this reason, we build upon HoroSVM. For a labeled dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with $\mathbf{x}_i \in \mathbb{B}^n$ and $y_i \in \{-1, 1\}$, it optimizes the convex loss function

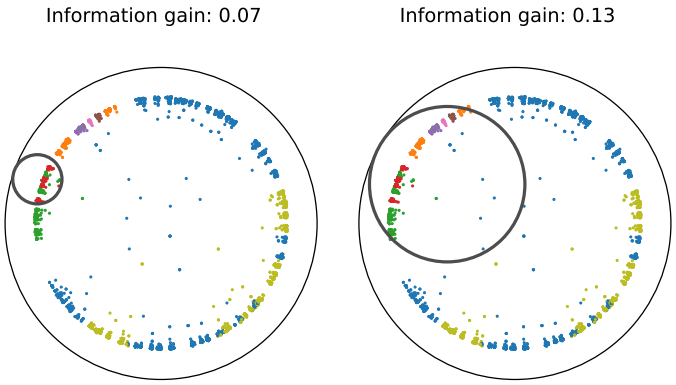
$$\boldsymbol{\ell}(\mu, \mathbf{w}, o; D) = \frac{1}{2}\mu^2 + C \sum_{i=1}^{N} \max\left(0, 1 - y_i(\mu B_{\mathbf{w}}^{-1}(\mathbf{x}_i) - o)\right), \tag{7}$$

where

$$B_{\mathbf{w}}^{-1}(\mathbf{x}) = \log \frac{1 - \|\mathbf{x}\|^2}{\|\mathbf{w} - \mathbf{x}\|^2}, \tag{8}$$

with $\mu, o \in \mathbb{R}^+$ and $\mathbf{w} \in \mathbb{S}^{n-1}$. The slack hyperparameter C controls the trade-off between misclassification tolerance and margin size. Solving for HoroSVM thus results in the three parameters $\mu, \mathbf{w}$, and o. In order to transform these results into a horosphere $\pi_{\mathbf{w},b}$, we set $b = -\frac{o}{\mu}$ and use $\mathbf{w}$ directly.

We repeat the classification K times in a one-versus-rest setting, with K the number of classes, and compute the information gain for all resulting horospheres. We select the horosphere with the highest information gain to split the data. We refer to this splitting procedure as the `HoroSplitter`. While this base version of the `HoroSplitter` already achieves competitive results, it naturally has limited capacity to deal with imbalanced data, and the one-versus-rest set-up limits its effectiveness in multi-class settings. Therefore, we introduce two additional components to the `HoroSplitter` that make it more appropriate specifically as a building block for hyperbolic decision trees.



(a) (b)

Figure 2: **Information gain from hierarchical splits.** The best (a) one-versus-rest split and (b) split found with our hyperclass heuristic. We find splits with a higher information gain by considering splits where more than one class is seen as positive.

3.2.3 Combining classes

The default `HoroSplitter` uses a one-versus-rest approach to deal with multi-class data. As a result, it can only find one-versus-rest splits. Nonetheless, a horosphere split with a high information gain could have more than one class in either partition. To find good splits with multiple classes while avoiding searching over all possible combinations, only the most promising combinations should be evaluated. For this, we design a heuristic that transforms the multi-class problem into a hierarchical set of binary problems by iteratively grouping classes into two hyperclasses based on their lowest common ancestor (LCA). Then, these binary combinations are added to the pool of one-versus-rest experiments and evaluated for their information gain.

Recall the connection of hyperbolic space to graph-theoretical trees. In a tree, similar leaf nodes lie in a small subtree and have their LCA at a high depth. In contrast, dissimilar nodes will have their LCA close to the root of the tree. We exploit the analog of this property in hyperbolic space to cluster together the most similar classes first, as similar classes are likely to be capturable by a single horosphere.

We represent each cluster by its hyperbolic mean. As computing the average in the Poincaré model involves the computationally expensive Fréchet mean (Lou et al., 2020), we use the Einstein midpoint after transforming the data to Klein coordinates instead. Specifically, data is transformed into Klein space $\mathbb{K}$ with

$$\mathbf{x}_{\mathbb{K}} = \frac{2\mathbf{x}_{\mathbb{B}}}{1 + \|\mathbf{x}_{\mathbb{B}}\|^2}, \tag{9}$$

and we compute the average $\boldsymbol{\phi}_y$ for class y using

$$\boldsymbol{\phi}_y = \frac{\sum_{i=1}^{N} \mathbb{1}_{[y_i=y]} \gamma_i \mathbf{x}_i}{\sum_{i=1}^{N} \gamma_i}, \tag{10}$$

where $\gamma_i = \frac{1}{\sqrt{1-\|\mathbf{x}_i\|^2}}$. Afterward, the mean is transformed back to the Poincaré ball with

$$\mathbf{x}_{\mathbb{B}} = \frac{\mathbf{x}_{\mathbb{K}}}{1 + \sqrt{1 - \|\mathbf{x}_{\mathbb{K}}\|^2}}. \tag{11}$$

We compute all pairwise similarities and iteratively group the two most similar classes together, where the similarity is based on the LCA of the means of the two classes. The LCA for two points in hyperbolic space is the point that is closest to the origin on the geodesic between them (Chami et al., 2020). As a result, the distance of the LCA from the origin can be seen as a similarity measure. Given two means $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$, we can calculate their similarity by

$$\text{sim}(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = 2 \tanh^{-1}\left(\sqrt{R^2 + 1} - R\right) \tag{12}$$

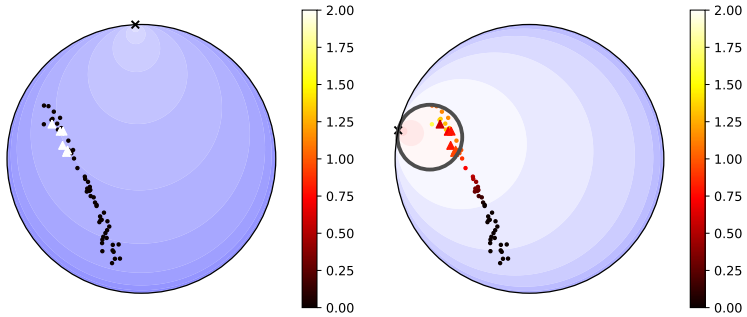


Figure 3: **Importance of class balancing.** The optimal solution on a binary classification problem found (left) without and (right) with class-balancing. The left horosphere has a large negative $\mathbf{b}$ and does not split the data. In contrast, the right horosphere has a higher loss but a positive information gain. Class labels are given by marker shape. The points and color bar are colored by loss; the background is colored by distance to the horosphere.

with

$$R = \sqrt{\left(\frac{\|\boldsymbol{\phi}_i\|^2 + 1}{2\|\boldsymbol{\phi}_i\| \cos(\alpha)}\right)^2 - 1} \tag{13}$$

and

$$\alpha = \tan^{-1}\left(\frac{1}{\sin(\theta)}\left(\frac{\|\boldsymbol{\phi}_i\|(\|\boldsymbol{\phi}_j\|^2 + 1)}{\|\boldsymbol{\phi}_j\|(\|\boldsymbol{\phi}_i\|^2 + 1)} - \cos(\theta)\right)\right), \tag{14}$$

where θ is the angle between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$. We repeat this until only two hyperclasses are left, for $K - 2$ iterations total. As a result, we only evaluate $K + K - 2$ horospheres per node. We show an example where evaluating hyperclasses is beneficial in Fig. 2.

3.2.4 Imbalanced data

SVM-based methods are known to behave poorly on imbalanced data, and `HoroSplitter` is no exception. We address this by proposing a class-balanced version of the `HoroSplitter`. Specifically, the problem comes from equation 7 finding an optimal solution that does not split the data and has an information gain of zero as a result. As an example, consider the binary problem in Figure 3. The optimal horosphere has its ideal point on the top of the Poincaré ball, a μ very close to 0 and o very close to 1; a result of the loss, in the case of a large imbalance, almost reducing to the degenerate form

$$\boldsymbol{\ell}_{\text{imbalanced}}(\mu, \mathbf{w}, o; D) \approxeq C \sum_{i=1}^{N} \max(0, 1 + y_i). \tag{15}$$

We remedy this limitation by incorporating class-balancing into `HoroSplitter`'s loss function (Cui et al., 2019). We rewrite the loss function in equation 7 as

$$\boldsymbol{\ell}(\mu, \mathbf{w}, o; D) = \frac{1}{2}\mu^2 + C \sum_{i=1}^{N} \frac{1 - \beta}{1 - \beta^{n_y}} \max(0, 1 - y_i(\mu B_{\mathbf{w}}^{-1}(\mathbf{x}_i) - o)), \tag{16}$$

where n_y gives the number of training samples belonging to class y and β is a hyperparameter. By adding more emphasis on the loss for samples of the minority class, we can find horospheres that split the data even in imbalanced cases.

Training and inference We build horospherical decision trees (HoroDT) by repeatedly applying the `HoroSplitter`. This process is repeated until all nodes are pure, *i.e.* contain one class, or a stopping criterion is met. Example stopping criteria include stopping when a node reaches a certain number of

samples or no split with an information gain higher than a threshold can be found. By combining multiple HoroDTs into an ensemble, we form a horospherical Random Forest (HoroRF). Each tree is trained on a randomly sampled (with replacement) subset of the data, and a subsample of the features is considered at every split. Predictions are made by a majority vote among the HoroDTs.

4 Experiments

4.1 Datasets

We evaluate HoroRF on two canonical hyperbolic classification benchmarks: WordNet subtree and network node classification (Cho et al., 2019; Fan et al., 2023). Additionally, we introduce three new multi-class WordNet experiments, which we will detail first.

Multi-class WordNet We propose three new multi-class WordNet experiments aimed at evaluating three distinct qualities. In the first experiment, the task is to classify samples into one of multiple subtrees with the same parent. In the second experiment, we pick nested subtrees, and samples must be classified into the smallest subtree they belong to. The third experiment combines the previous experiments with nested classes of multiple subtrees. Unlike previous benchmarks, these are difficult multi-class experiments on large, imbalanced datasets. We provide the full details of the experiments in the appendix. The datasets are split into train, validation, and test sets with a ratio of 60:20:20. A hyperparameter search is done on the validation set, and the macro-f1 score over three trials on the test set is reported.

WordNet We run binary classification experiments on the nouns of WordNet (Fellbaum, 2010), embedded into hyperbolic space using hyperbolic entailment cones (Ganea et al., 2018a). The goal is to identify whether a sample belongs to a semantic category (*i.e.*, a subtree of the hyperbolic embeddings). Following (Fan et al., 2023), we split the data into 80% for training and 20% for testing and report the best results from a grid search over three runs in AUPR. We find that three of the larger subtrees typically used for these experiments (`animal.n.01`, `group.n.01`, `mammal.n.01`) can be solved near-perfectly with random forests, both Euclidean and hyperbolic. Therefore, we add experiments on two additional subtrees, `occupation.n.01` and `rodent.n.01`, which are smaller and more difficult subtrees for classification purposes.

Networks We run node classification experiments on four network datasets embedded into hyperbolic space. These networks are (1) *karate* (Zachary, 1977), a network of 34 members of a karate club split into two factions, (2) *polblogs* (Adamic & Glance, 2005), a network of 1224 hyperlinks in political blogs split into two communities, (3) football (Girvan & Newman, 2002), a network of 115 colleges linked by football games split into 12 leagues, and (4) *polbooks*[1], a network of 105 books split into three affiliations.

We use the public embeddings provided by Cho et al. (2019). Each dataset has five sets of embeddings obtained using the method of Chamberlain et al. (2017). Following previous work (Fan et al., 2023), we run a grid search with 5-fold stratified cross-validation and report the best average micro-f1 score over five seeds, using a different embedding for each seed.

4.2 Evaluation

Baselines We evaluate HoroRF against six baselines, split into two categories. The first set covers hyperbolic classifier baselines, namely hyperbolic multiple logistic regression (HypMLR; Ganea et al. (2018b)) and the state-of-the-art hyperbolic SVM, HoroSVM (Fan et al., 2023). The second set of baselines comprises Euclidean counterparts of the hyperbolic classifiers, namely a linear SVM (LinSVM), an SVM with an RBF kernel (RBFSVM; (Cortes & Vapnik, 1995)), an axis-aligned random forest (RF) (Breiman, 2001), and an oblique random forest (OblRF) with a similar set-up to HoroRF. The OblRF uses a cost-balanced linear SVM to find optimal splits (see *e.g.* (Do et al., 2010)). If the SVM fails to converge, it defaults to using the best axis-aligned split. It finds hyperclasses by iteratively merging the two classes with the closest means.

[1] http://www-personal.umich.edu/ mejn/netdata/

	Animal	Group	Worker	Mammal	Tree	Solid	Occupation	Rodent
Euclidean								
LinSVM	2.5±0.0	6.0±0.6	0.7±0.0	0.7±0.0	0.6±0.0	0.8±0.0	0.2±0.0	0.1±0.0
RBFSVM	2.5±0.0	6.0±0.6	0.7±0.0	0.7±0.0	0.6±0.0	0.8±0.0	0.2±0.0	0.1±0.0
RF	**98.6**±0.2	96.6±0.4	70.0±3.1	**99.3**±0.3	75.8±2.0	92.5± 1.2	59.8±1.1	34.5±3.0
OblRF	98.4±0.1	**96.7** ±0.3	73.3±1.3	98.8±0.3	76.2±3.2	92.9±1.7	65.0±2.4	38.2±0.9
Hyperbolic								
HypMLR	3.2 ±0.1	6.5 ±0.0	1.2 ±0.2	0.9 ±0.0	1.3±0.8	1.0 ±0.2	0.9±0.9	0.4 ±0.4
HoroSVM	92.4±1.3	71.0±0.5	39.9±3.6	89.8±0.9	41.5±1.0	65.7±0.9	13.0±1.5	13.6±2.0
HoroRF	98.5±0.2	96.6±0.4	**73.4**± 1.7	**99.3**±0.4	**76.6**± 3.8	**93.5**± 1.3	**65.6**± 1.5	**39.0**± 2.4

Table 1: **Comparative evaluation on binary WordNet experiments.** We follow the experimental protocol of Fan et al. (2023) and report the mean and standard deviation of the AUPR over three runs. Underlined gives the best hyperbolic method, **Bold** denotes the best method overall. On aggregate across the experiments, HoroRF obtains the best performance.

As is common practice (Cho et al., 2019; Fan et al., 2023), the Euclidean baselines are run on the same embeddings as the hyperbolic methods, which we also find to perform better compared to mapping the embeddings back to Euclidean space first. Moreover, as the SVMs, HypMLR, and HoroSVM baselines are negatively affected by the class imbalance in the WordNet experiments, we report cost-balanced results on those benchmarks, where the loss is weighted inversely proportional to the class frequency.

Implementation Details We use 100 trees for all tree-based methods. We use the Gini impurity to calculate the information gain. At every node, we set C to 2^n where n is randomly sampled from $\{-3, -2, \ldots, 5\}$. We stop splitting when a node reaches a certain number of samples m. In the case of ties in information gain between possible splits, a random split is selected. If the `HoroSplitter` fails to converge, we choose the best horosphere from a small number of random ideal points. We set m and the class-balancing hyperparameter β via grid search. Further details are given in the appendix.

4.3 Results

WordNet The results for the WordNet subtree classification experiments are shown in Tab. 1. The Euclidean SVMs are unable to handle the large imbalance in the datasets despite the loss re-weighting. The random forest approaches are better equipped to deal with this type of data. The results show that HoroRF outperforms the Euclidean tree-based methods in most cases. The differences are most pronounced in the hardest experiments, *occupation* and *rodent*, while we find the tree-based methods to perform similarly on the saturated experiments with close to perfect scores, i.e., *animal*, *group*, and *mammal*. Overall, HoroRF performs better than the well-known Euclidean SVMs and random forest approaches.

Compared to the hyperbolic classifiers, we find that both the hyperbolic MLR and SVM baselines struggle to deal with the imbalanced data, despite the loss re-weighting. Neither are able to match the performance of HoroRF. We argue this is the case as real-world data cannot be distinguished with a single horosphere, but requires recursive horospherical separation as done in HoroRF. We conclude from the WordNet experiment that HorRF is a competitive classifier even in the binary setting.

Multi-class WordNet The results for the multi-class WordNet experiments are shown in Table 2, which paint a similar picture as the previous experiments. For the Euclidean methods, the tree-based methods outperform the SVMs, with OblRF, on average, having a slight edge over the axis-aligned random forest. HoroRF outperforms the SVMs and axis-aligned RF in all three cases, and despite reaching an equal score in the first experiment, it outperforms OblRF in the remaining two. Compared to the other hyperbolic classifiers, HoroRF reaches far higher performance on these complex imbalanced multi-class experiments, on which both HypMLR and HoroSVM struggle greatly.

	Same level	Nested	Both
Euclidean			
LinSVM	48.8 ±0.9	59.7 ±0.5	35.6 ±0.2
RBFSVM	80.0 ±1.8	89.8 ±1.1	70.8 ±0.6
RF	89.7±0.7	91.7±0.3	81.5±1.5
OblRF	**91.3**±0.6	93.0±0.2	81.3±1.4
Hyperbolic			
HypMLR	29.0 ±4.5	57.6 ±0.5	35.9 ±6.7
HoroSVM	50.2 ±2.2	56.0 ±0.7	35.2 ±0.5
HoroRF	**91.3**±0.3	**93.3**±1.1	**81.9**±1.5

Table 2: **Comparative evaluation on multi-class WordNet.** We run a grid search to determine the optimal configuration on the validation set and report results over three trials in macro-f1 score on the test set. Underlined gives the best hyperbolic method, **Bold** denotes the best method overall. HoroRF performs best in all settings.

	Binary		Multi-class	
	Karate	Polbooks	Football	Polblogs
Euclidean				
LinSVM	**95.4**±2.3	92.4±0.3	33.2±5.1	85.5±0.9
RBFSVM	**95.4**±2.3	92.4±0.3	35.5±4.7	84.4±1.9
RF	94.3±3.1	92.1±0.3	36.2±4.9	85.1±2.1
OblRF	94.8 ±2.2	92.1 ±0.3	36.7 ±2.7	84.4 ±1.3
Hyperbolic				
HypMLR	93.1±3.4	90.9±0.7	**40.2**±2.5	81.5±1.5
HoroSVM	**95.4**±2.3	92.4±0.2	34.3±1.8	85.3±0.8
HoroRF	**95.4**±2.3	**92.5**±0.3	38.3±1.8	**86.1**±1.0

Table 3: **Comparative evaluation on network datasets.** We follow the experimental protocol of Fan et al. (2023) and report the mean and standard deviation of the micro-f1 score over five trials of 5-fold stratified cross-validation. Underlined gives the best hyperbolic method, **Bold** denotes the best method overall. HoroRF obtains the best performance on aggregate over the datasets.

Networks The results on the network datasets are shown in Table 3. Our HoroRF outperforms the Euclidean methods in all cases, besides matching the SVMs on *karate*, showing that the choice of horospheres over hyperplanes results in better performance on data lying in hyperbolic space.

As for the hyperbolic methods, HypMLR is outclassed by both HoroRF and HoroSVM in all cases except *football*. HoroSVM and HoroRF perform similarly in the binary experiments, with a slight edge for HoroRF on *polblogs*. In contrast, for the multi-class experiments *football* and *polbooks*, HoroRF outclasses HoroSVM by 4.0 and 0.6 macro f-1, respectively, confirming the benefit of HoroRF over HoroSVM in more complex cases. Overall, we find that HoroRF is the most consistent and best-performing classifier across all benchmarks.

4.4 Ablations and Visualizations

Hierarchical Classification Hyperbolic methods have been proven successful when evaluated on a wide range of hierarchical tasks Cao et al. (2020); Ghadimi Atigh et al. (2021); Surís et al. (2021); Tifrea et al. (2019). Here, we aim to verify our hypothesis that random forests are well-suited for hyperbolic space by comparing HoroRF with other SOTA hyperbolic classifiers on hierarchical classification, with metrics explicitly designed to evaluate their success in capturing the hierarchy.

	CIFAR10		STL10	
	Mis. Sev. (↓)	HD@1 (↓)	Mis. Sev. (↓)	HD@1 (↓)
HypMLR	60.8±1.7	10.2±0.0	54.2±2.8	22.4±0.3
HoroSVM	59.9±0.1	12.5±0.0	54.4±1.0	26.0±0.3
HoroRF	**58.2**±0.3	**9.9**±0.2	**53.4**±0.2	**22.3**±0.1

Table 4: **Evaluating hyperbolic classifiers for hierarchical classification.** We run a grid search to determine the optimal configuration on the validation set and report the mean and standard deviation of the mistake severity and hierarchical distance@1 on the test set over three seeds. **Bold** denotes the best method. HoroRF performs best.

Ideal Points	Hyper-classes	Class-balanced	f1
Axis-aligned	✗	✗	9.9
`HoroSplitter`	✗	✗	35.7
`HoroSplitter`	✗	✓	36.5
`HoroSplitter`	✓	✗	37.4
`HoroSplitter`	✓	✓	**38.3**

Table 5: **Ablating the `HoroSplitter`** in HoroRF on the *football* dataset. We follow the experimental protocol of Fan et al. (2023) and report the mean and standard deviation of the micro-f1 score over five trials of 5-fold stratified cross-validation. Our splitting function, hyperclass-based aggregation, and class balancing are all important for effective hyperbolic random forests.

We evaluate the methods using CIFAR10 Krizhevsky et al. (2009) and STL10 Coates et al. (2011) using the CIFAR10 hierarchy from Sebastian & Sebastian (2023). Note that STL10 has the same classes as CIFAR10, with the only difference being the substitution of `frog` for `monkey`. As such, we maintain the hierarchy from Sebastian & Sebastian (2023) for STL10 but modify it by removing `frog` and incorporating `monkey` into the mammal subtree.

We embed the images into 9-dimensional space by training a hyperbolic prototype ResNet-18 with prototypes uniformly distributed over the hyperbole Kasarla et al. (2022) and train the classifiers on these 9-dimensional image features. We obtain hyperbolic embeddings for 3600 training, 900 validation, and 1000 test samples. To compare the classifiers, we run a grid search on the training set and select the optimal configuration based on validation performance. Then, we report the mistake severity and hierarchical distance@1 on the test set Garg et al. (2022). These metrics both use the LCA distance, computed by dividing the height of the LCA of the predicted and ground-truth class by the tree height. The mistake severity is the average of this distance for all misclassified samples, whereas the HD@1 computes the average distance for all the samples Karthik et al. (2021).

From the results in Tab. 4, HoroRF is the best hyperbolic method, confirming its ability to model the hierarchy effectively and the benefits of our generalization of random forests to hyperbolic space.

Ablating hyperclasses and balancing. We perform an ablation study to validate our choices on the *football* dataset. From Tab. 5, we find a large benefit in using our `HoroSplitter` to find splits instead of enumerating all possible horospheres at axis-aligned ideal points. Furthermore, we show how both our hyperclass and class-balancing additions improve upon the base `HoroSplitter`, and their combination enables us to reach the best performance.

Visualization & synthetic evaluation We build a synthetic dataset following Ganea et al. (2018a) by creating a synthetic tree of depth six with a branching factor of four and embedding it into hyperbolic space with hyperbolic entailment cones. We then partition the nodes into five classes from two depths. Using this

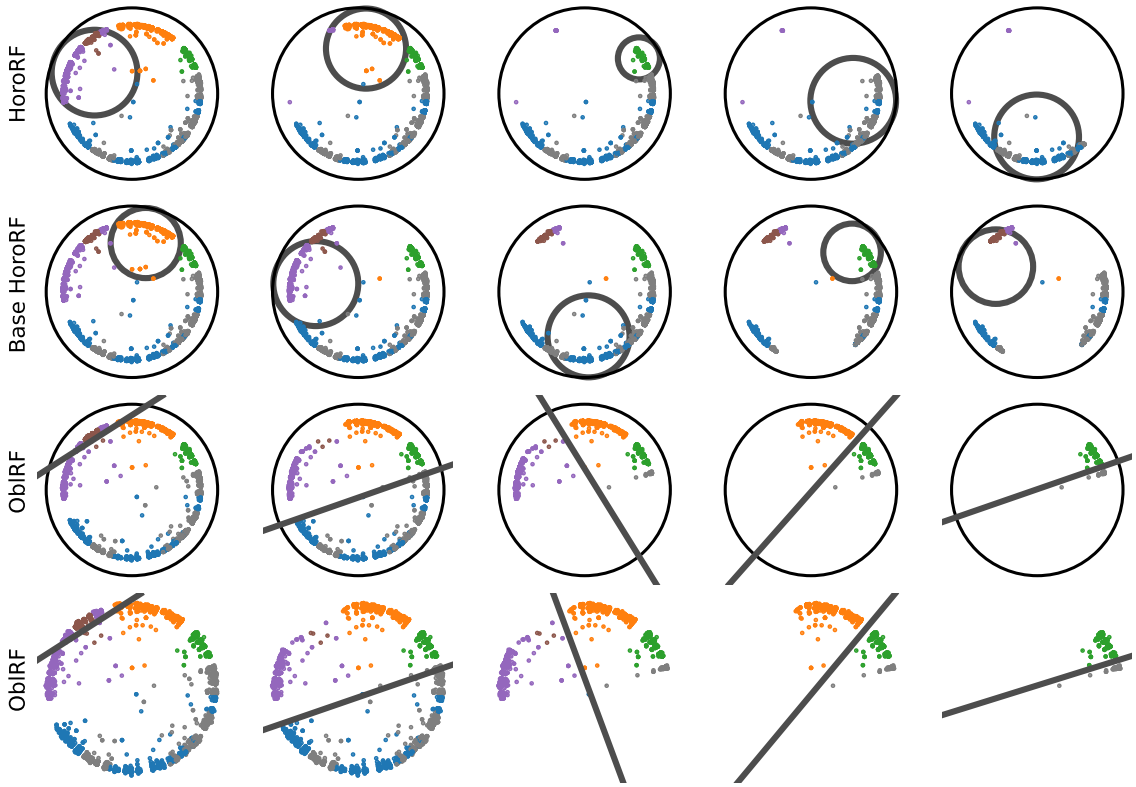


Figure 4: **Visualizing HoroRF & OblRF splits.** We show five splits for HoroRF, a basic version of HoroRF without class-balancing and hyperclasses, OblRF on the original hyperbolic embeddings, and OblRF after mapping the data to Euclidean space. The split with the highest impurity is chosen at every level to visualize the next level. Splitting in hyperbolic space with balanced and multi-level horospheres obtains good splits for hyperbolic classification.

Depth	1	2	3	10	No limit
OblRF Euc	34.5±0.1	**68.8**±2.5	83.3±0.0	**99.3**±0.0	98.9±0.0
OblRF Hyp	33.6±0.7	65.6±1.3	83.2±0.3	99.1±0.0	98.9±0.0
Base HoroRF	60.8±0.0	67.8±4.9	83.6±1.6	99.1±0.0	**99.1**±0.3
HoroRF	**63.1**±3.2	67.4±5.1	**85.5**±1.8	**99.3**±0.0	98.9±0.3

Table 6: **Effect of tree depth on performance on the synthetic dataset.** We run a grid search to determine the optimal configuration on the validation set and report results over three trials in micro-f1 score on the test set. **Bold** denotes the best method, underlined the second best. Splitting in hyperbolic space with balanced and multi-level horospheres consistently reaches high performance.

dataset, we visualize the splits of HoroRF, HoroRF without hyperclasses or class-balancing, OblRF on the hyperbolic embeddings, and OblRF on the Euclidean embeddings in Fig. 4. We additionally verify which method is the most successful at various depths in Tab. 6, where we find that HoroRF consistently reaches high performance and is especially effective with a single split.

Reducing runtime A limitation of HoroRF is its runtime on large datasets. While the `HoroSplitter` scales linearly with the number of samples as HoroSVM has a linear time complexity, it is applied multiple times per node and repeated for every node. We designed and conducted two experiments showing how to mitigate the computational time needed to obtain good results: reducing the number of trees and lowering

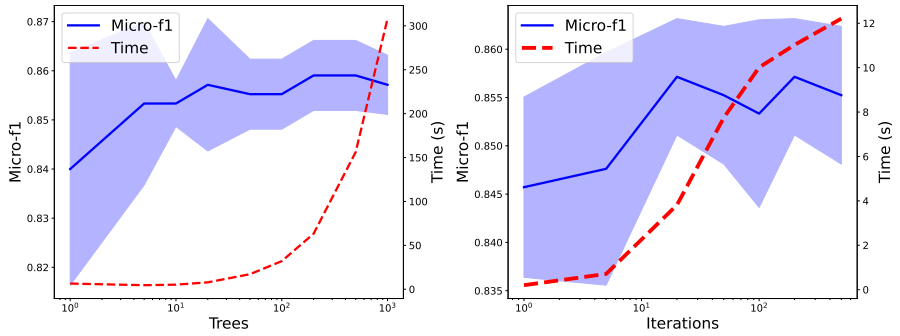


Figure 5: **Analyzing ways to reduce the computational time of HoroRF on *polbooks*.** In (a), we show that the optimal performance is already reached with around 20 trees. In (b), we show that by reducing the number of optimization iterations of HoroSVM, we can further reduce computational time while maintaining high performance.

the number of HoroSVM optimization iterations. From the results in Fig. 5, we find that both the number of trees and HoroSVM optimization iterations can be greatly reduced without sacrificing performance. Lowering the number of HoroSVM iterations is a logical way to reduce computational time, as we are not interested in finding the exact optimal horosphere; a solution close to it suffices for our purpose and might even benefit HoroRF due to the increased variation in splits.

5 Conclusion

We present HoroRF, a random forest algorithm in hyperbolic space. HoroRF constructs its trees by repeated application of the `HoroSplitter`, which finds horosphere-based splits with high information gain. We show the benefits of additional components aimed at improving performance on imbalanced and multi-class datasets. Extensive experiments on two established and one newly introduced benchmark show its effectiveness over both existing hyperbolic classifiers as well as their Euclidean counterparts.

Further advances in hyperbolic classification can easily be incorporated into the framework, as HoroRF is in no way restricted to a single type of classifier. In future work, we aim to investigate the benefits of combining multiple types of splits, for example by including hyperbolic logistic regression to provide additional split options to include next to the horosphere splits. We furthermore expect that any improvement in optimization of HoroSVM as splitting operation, akin to hyper-optimized libraries for Euclidean SVMs (Chang & Lin, 2011), will have direct benefits for HoroRF, especially when dealing with many classes.

References

Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pp. 36–43, 2005.

Ola Ahmad and Freddy Lecue. Fisheyehdk: Hyperbolic deformable kernel learning for ultra-wide field-of-view image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 5968–5975, 2022.

Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

Herbert Busemann. *The geometry of geodesics.* Academic Press, 1955.

Erick Cantú-Paz and Chandrika Kamath. Inducing oblique decision trees with evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(1):54–68, 2003.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. Hypercore: Hyperbolic and co-graph representation for automatic icd coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3105–3114, 2020.

Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pp. 161–168, 2006.

Benjamin Paul Chamberlain, James Clough, and Marc Peter Deisenroth. Neural embeddings of graphs in hyperbolic space. *arXiv preprint arXiv:1705.10359*, 2017.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 2019.

Ines Chami, Albert Gu, Vaggos Chatziafratis, and Christopher Ré. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. *Advances in Neural Information Processing Systems*, 33: 15065–15076, 2020.

Ines Chami, Albert Gu, Dat P Nguyen, and Christopher Ré. Horopca: Hyperbolic dimensionality reduction via horospherical projections. In *International Conference on Machine Learning*, pp. 1419–1429. PMLR, 2021.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Eli Chien, Chao Pan, Puoya Tabaghi, and Olgica Milenkovic. Highly scalable and provably accurate classification in poincaré balls. In *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 61–70. IEEE, 2021.

Philippe Chlenski, Ethan Turok, Antonio Moretti, and Itsik Pe'er. Fast hyperboloid decision tree algorithms. *arXiv preprint arXiv:2310.13841*, 2023.

Hyunghoon Cho, Benjamin DeMeo, Jian Peng, and Bonnie Berger. Large-margin classification in hyperbolic space. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1832–1840. PMLR, 2019.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Shanmukha Ramakrishna Vedantam. Hyperbolic image-text representations. In *International Conference on Machine Learning*, 2023.

Thanh-Nghi Do, Philippe Lenca, Stéphane Lallich, and Nguyen-Khang Pham. Classifying very-high-dimensional data with random forests of oblique decision trees. *Advances in knowledge discovery and management*, pp. 39–55, 2010.

Aiden Durrant and Georgios Leontidis. Hmsn: Hyperbolic self-supervised learning by clustering with ideal prototypes. *arXiv preprint arXiv:2305.10926*, 2023.

Xiran Fan, Chun-Hao Yang, and Baba C Vemuri. Horocycle decision boundaries for large margin classification in hyperbolic space. *arXiv preprint arXiv:2302.06807*, 2023.

Pengfei Fang, Mehrtash Harandi, and Lars Petersson. Kernel methods in hyperbolic spaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10665–10674, 2021.

Christiane Fellbaum. Princeton university: About wordnet, 2010.

Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1): 3133–3181, 2014.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *International Conference on Machine Learning*, pp. 1646–1655. PMLR, 2018a.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018b.

Ashima Garg, Depanshu Sani, and Saket Anand. Learning hierarchy aware features for reducing mistake severity. In *European Conference on Computer Vision*, pp. 252–267. Springer, 2022.

Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic busemann learning with ideal prototypes. *Advances in Neural Information Processing Systems*, 34:103–115, 2021.

Mina Ghadimi Atigh, Julian Schoep, Erman Acar, Nanne Van Noord, and Pascal Mettes. Hyperbolic image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35:507–520, 2022.

David Heath, Simon Kasif, and Steven Salzberg. Induction of oblique decision trees. In *IJCAI*, volume 1993, pp. 1002–1007. Citeseer, 1993.

Shyamgopal Karthik, Ameya Prabhu, Puneet K Dokania, and Vineet Gandhi. No cost likelihood manipulation at test time for making better mistakes in deep networks. *International Conference on Learning Representations*, 2021.

Tejaswi Kasarla, Gertjan Burghouts, Max van Spengler, Elise van der Pol, Rita Cucchiara, and Pascal Mettes. Maximum class separation as inductive bias in one matrix. *Advances in Neural Information Processing Systems*, 35:19553–19566, 2022.

Rakesh Katuwal and Ponnuthurai N Suganthan. Enhancing multi-class classification of random forest using random vector functional neural network and oblique decision surfaces. In *2018 International Joint Conference on Neural Networks*, pp. 1–8. IEEE, 2018.

Rakesh Katuwal, Ponnuthurai Nagaratnam Suganthan, and Le Zhang. Heterogeneous oblique random forest. *Pattern Recognition*, 99:107078, 2020.

Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6418–6428, 2020.

Anna Klimovskaia, David Lopez-Paz, Léon Bottou, and Maximilian Nickel. Poincaré maps for analyzing complex hierarchies in single-cell data. *Nature communications*, 11(1):2966, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in neural information processing systems*, 2019.

Shaoteng Liu, Jingjing Chen, Liangming Pan, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang. Hyperbolic visual embedding learning for zero-shot recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

Aaron Lou, Isay Katsman, Qingxuan Jiang, Serge Belongie, Ser-Nam Lim, and Christopher De Sa. Differentiating through the fréchet mean. In *International Conference on Machine Learning*, 2020.

Gian Marconi, Carlo Ciliberto, and Lorenzo Rosasco. Hyperbolic manifold regression. In *International Conference on Artificial Intelligence and Statistics*, pp. 2570–2580. PMLR, 2020.

Bjoern H Menze, B Michael Kelm, Daniel N Splitthoff, Ullrich Koethe, and Fred A Hamprecht. On oblique random forests. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II 22*, pp. 453–469. Springer, 2011.

Pascal Mettes, Mina Ghadimi Atigh, Martin Keller-Ressel, Jeffrey Gu, and Serena Yeung. Hyperbolic deep learning in computer vision: A survey. *arXiv preprint arXiv:2305.06611*, 2023.

Nicholas Monath, Manzil Zaheer, Daniel Silva, Andrew McCallum, and Amr Ahmed. Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 714–722, 2019.

Sreerama K Murthy, Simon Kasif, Steven Salzberg, and Richard Beigel. Oc1: A randomized algorithm for building oblique decision trees. In *Proceedings of AAAI*, volume 93, pp. 322–327. Citeseer, 1993.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, volume 30, 2017.

Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International conference on machine learning*, 2018.

Chao Pan, Eli Chien, Puoya Tabaghi, Jianhao Peng, and Olgica Milenkovic. Provably accurate and scalable linear classifiers in hyperbolic spaces. *Knowledge and Information Systems*, pp. 1–34, 2023.

Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 44(12):10023–10044, 2021.

Rinu Ann Sebastian and Anu Maria Sebastian. Leveraging semantic similarity to mitigate the severity of misclassification for safety critical applications. *Multimedia Tools and Applications*, 82(8):12615–12633, 2023.

Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. In *International Conference on Learning Representations*, 2021.

Sho Sonoda, Isao Ishikawa, and Masahiro Ikeda. Fully-connected network on noncompact symmetric space and ridgelet transform based on helgason-fourier analysis. In *International Conference on Machine Learning*, pp. 20405–20422. PMLR, 2022.

Jianing Sun, Zhaoyue Cheng, Saba Zuberi, Felipe Pérez, and Maksims Volkovs. Hgcf: Hyperbolic graph convolution networks for collaborative filtering. In *Proceedings of the Web Conference 2021*, 2021a.

Li Sun, Zhongbao Zhang, Jiawei Zhang, Feiyang Wang, Hao Peng, Sen Su, and S Yu Philip. Hyperbolic variational graph neural network for modeling dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4375–4383, 2021b.

Dídac Surís, Ruoshi Liu, and Carl Vondrick. Learning the predictability of the future. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12607–12617, 2021.

Chang-Yu Tai, Ming-Yao Li, and Lun-Wei Ku. Hyperbolic disentangled representation for fine-grained aspect extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11358–11366, 2022.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincaré glove: Hyperbolic word embeddings. *International Conference on Learning Representations*, 2019.

Max van Spengler, Erwin Berkhout, and Pascal Mettes. Poincaré resnet. In *International Conference on Computer Vision*, 2023.

Melanie Weber, Manzil Zaheer, Ankit Singh Rawat, Aditya K Menon, and Sanjiv Kumar. Robust large-margin learning in hyperbolic space. *Advances in Neural Information Processing Systems*, 33:17863–17873, 2020.

Menglin Yang, Min Zhou, Zhihao Li, Jiahong Liu, Lujia Pan, Hui Xiong, and Irwin King. Hyperbolic graph neural networks: a review of methods and applications. *arXiv preprint arXiv:2202.13852*, 2022.

Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.