HYPERBOLIC IMAGE EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Computer vision tasks such as image classification, image retrieval and few-shot learning are currently dominated by Euclidean and spherical embeddings, so that the final decisions about class belongings or the degree of similarity are made using linear hyperplanes, Euclidean distances, or spherical geodesic distances (cosine similarity). In this work, we demonstrate that in many practical scenarios hyperbolic embeddings provide a better alternative.

1 INTRODUCTION

High-dimensional embeddings are ubiquitous in modern computer vision. Many, perhaps most, modern computer vision systems learn non-linear mappings (in the form of deep convolutional networks) from the space of images or image fragments into high-dimensional spaces. The operations at the end of deep networks imply a certain type of geometry of the embedding spaces. For example, image classification networks (Krizhevsky et al., 2012; LeCun et al., 1989) use linear operators (matrix multiplication) to map embeddings in the penultimate layer to class logits. The class boundaries in the embedding space are thus piecewise-linear, and pairs of classes are separated by Euclidean hyperplanes. The embeddings learned by the model in the penultimate layer, therefore, live in the Euclidean space. The same can be said about systems where Euclidean distances are used to perform image retrieval (Oh Song et al., 2016; Sohn, 2016; Wu et al., 2017), face recognition (Parkhi et al., 2015; Wen et al., 2016) or one-shot learning (Snell et al., 2017).

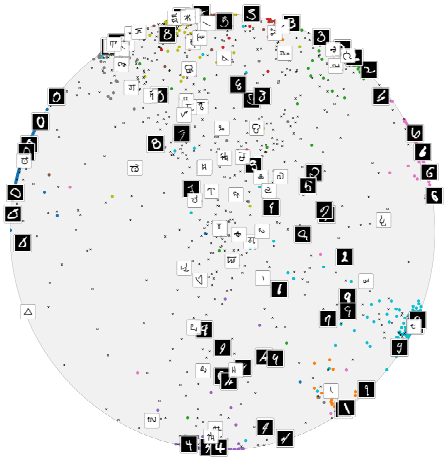

Figure 1: An example of two–dimensional Poincaré embeddings computed by a hyperbolic neural network trained on MNIST, and evaluated additionally on Omniglot. Ambiguous and unclear images from MNIST, as well as most of the images from Omniglot are embedded near the center, while samples with clear class labels (or characters from Omniglot similar to one of the digits) lie near the boundary.

Alternatively, some few-shot learning (Vinyals et al., 2016), face recognition (Schroff et al., 2015) and person re-identification methods (Ustinova & Lempitsky, 2016; Yi et al., 2014) learn spherical embeddings, so that sphere projection operator is applied at the end of a network that computes the embeddings. Cosine similarity (closely associated with sphere geodesic distance) is then used by such architectures to match images.

Euclidean spaces with their zero curvature and spherical spaces with their positive curvature have certain profound implications on the nature of embeddings that existing computer vision systems can learn. In this work, we argue that hyperbolic spaces with negative curvature might often be more appropriate for learning embedding of images. Towards this end, we add the recently-proposed hyperbolic network layers (Ganea et al., 2018) to the end of several computer vision networks, and present a number of experiments corresponding to image classification, one-shot, and few-shot

learning and person re-identification. We show that in many cases, the use of hyperbolic geometry improves the performance over Euclidean or spherical embeddings.

Motivation for hyperbolic image embeddings. The use of hyperbolic spaces in natural language processing (Nickel & Kiela, 2017; Tifrea et al., 2018; Dhingra et al., 2018) is motivated by their natural ability to embed hierarchies (e.g., tree graphs) with low distortion (Sarkar, 2011). Hierarchies are ubiquitous in natural language processing. First, there are natural hierarchies corresponding to, e.g., biological taxonomies and linguistic ontologies. Likewise, a more generic short phrase can have many plausible continuations and is therefore semantically-related to a multitude of long phrases that are not necessarily closely related to each other (in the semantic sense). The innate suitability of hyperbolic spaces to embedding hierarchies (Sala et al., 2018a; Sarkar, 2011) explains the success of such spaces in natural language processing (Nickel & Kiela, 2017).

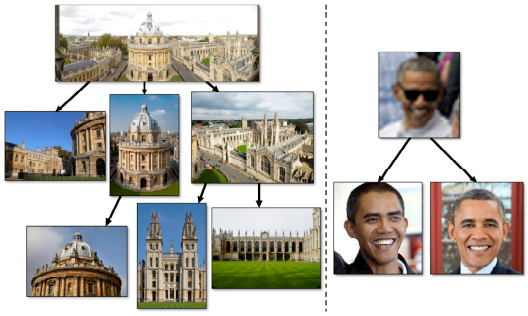

Figure 2: In many computer vision tasks, we want to learn image embeddings that obey the hierarchical constraints as shown above.

Here, we argue that similar hierarchical relations between images are common in computer vision tasks (Figure 2). One can observe the following example cases:

- In image retrieval, an overview photograph is related to many images that correspond to the close-ups of different distinct details. Likewise, for classification tasks in-the-wild, an image containing the representatives of multiple classes is related to images that contain representatives of the classes in isolation. Embedding a dataset that contains composite images into continuous space is therefore similar to embedding a hierarchy.
- In some tasks, more generic images may correspond to images that contain less information and are therefore more ambiguous. E.g., in face recognition, a blurry and/or low-resolution face image taken from afar can be related to many high-resolution images of faces that clearly belong to distinct people. Again natural embeddings for image datasets that have widely varying image quality/ambiguity calls for retaining such hierarchical structure.

In order to build deep learning models which operate on the embeddings to hyperbolic spaces, we capitalize on recent developments (Ganea et al., 2018), which construct the analogues of familiar layers (such as a feed–forward layer, or a multinomial regression layer) in hyperbolic spaces. We show that many standard architectures used for tasks of image classification, and in particular in the few–shot learning setting can be easily modified to operate on hyperbolic embeddings, which in many cases also leads to their improvement.

2 Poincaré ball model

Formally, n-dimensional hyperbolic space denoted as $\mathbb{H}^n$ is defined as the homogeneous, simply connected n-dimensional Riemannian manifold of constant negative sectional curvature. The property of constant negative curvature makes it analogous to the ordinary Euclidean sphere (which has constant positive curvature), however, the geometrical properties of the hyperbolic space are very different. It is known that hyperbolic space cannot be isometrically embedded into Euclidean space (Krioukov et al., 2010; Linial et al., 1998), but there exist several well–studied *models* of hyperbolic geometry. In every model a certain subset of Euclidean space is endowed with a *hyperbolic metric*,

however, all these models are isomorphic to each other and we may easily move from one to another base on where the formulas of interest are easier. We follow the majority of NLP works and use the *Poincaré ball* model. Investigating the alternative models that might provide better numerical stability remain future work (though already started in the NLP community (Nickel & Kiela, 2018; Sala et al., 2018b)). Here, we provide a very short summary of the model.

The Poincaré ball model $(\mathbb{D}^n, g^{\mathbb{D}})$ is defined by the manifold $\mathbb{D}^n = \{\mathbf{x} \in \mathbb{R}^n \colon \|\mathbf{x}\| < 1\}$ endowed with the Riemannian metric $g^{\mathbb{D}}(\mathbf{x}) = \lambda_{\mathbf{x}}^2 g^E$, where $\lambda_{\mathbf{x}} = \frac{2}{1-\|\mathbf{x}\|^2}$ is the *conformal factor* and g^E is the Euclidean metric tensor $g^E = \mathbf{I}^n$. In this model the *geodesic distance* between two points is given by the following expression:

$$d_{\mathbb{D}}(\mathbf{x}, \mathbf{y}) = \operatorname{arccosh}\left(1 + 2\frac{\|\mathbf{x} - \mathbf{y}\|^2}{(1 - \|\mathbf{x}\|^2)(1 - \|\mathbf{y}\|^2)}\right). \tag{1}$$

In order to define the *hyperbolic average*, we will make use of the *Klein model* of hyperbolic space. Similarly to the Poincaré model, it is defined on the set $\mathbb{K}^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < 1\}$, however, with a different metric, not relevant for further discussion. In Klein coordinates, the hyperbolic average (generalizing the usual Euclidean mean) takes the most simple form, and we present the necessary formulas in Section 4.

From the viewpoint of hyperbolic geometry, all points of Poincaré ball are equivalent. The models that we consider below are, however, hybrid in the sense that most layers use Euclidean operators, such as standard generalized convolutions, while only the final layers operate within the hyperbolic geometry framework. The hybrid nature of our setups makes the origin a special point, since from the Euclidean viewpoint the local volumes in Poincare ball expand exponentially from the origin to the boundary. This leads to the useful tendency of the learned embeddings to place more generic/ambiguous objects closer to the origin, while moving more specific objects towards the boundary. The distance to the origin in our models therefore provides a natural estimate of uncertainty, that can be used in several ways, as we show below.

3 Related work

Hyperbolic language embeddings Hyperbolic embeddings in the natural language processing field have recently been very successful (Nickel & Kiela, 2017; 2018). They are motivated by the innate ability of hyperbolic spaces to embed hierarchies (e.g., tree graphs) with low distortion (Sala et al., 2018b; Sarkar, 2011). The main result in this area states that any tree can be embedded into (two dimensional) hyperbolic space with arbitrarily low distortion. Another direction of research, more relevant to the present work is based on imposing hyperbolic structure on activations of neural networks (Ganea et al., 2018; Gulcehre et al., 2019).

Few–shot learning The task of few–shot learning, which has recently attracted a lot of attention, is concerned with the overall ability of the model to generalize to unseen data during training. A body of papers devoted to few–shot classification that focuses on metric learning methods includes Siamese Networks (Koch et al., 2015), Matching Networks (Vinyals et al., 2016), Prototypical Networks (Snell et al., 2017), Relation Networks (Sung et al., 2018). In contrast, other models apply meta-learning to few-shot learning: e.g., MAML by (Finn et al., 2017), Meta-Learner LSTM by (Ravi & Larochelle, 2016), SNAIL by (Mishra et al., 2018). While these methods employ either Euclidean or spherical geometries (like in (Vinyals et al., 2016)), there is no model extension to hyperbolic space.

Person re-identification The task of person re-identification is to match pedestrian images captured by possibly non-overlapping surveillance cameras. Papers (Ahmed et al., 2015; Guo & Cheung, 2018; Wang et al., 2018) adopt the pairwise models that accept pairs of images and output their similarity scores. The resulting similarity scores are used to classify the input pairs as being matching or non-matching. Another popular direction of work includes approaches that aim at learning a mapping of the pedestrian images to the Euclidean descriptor space. Several papers, e.g., (Suh et al., 2018; Yi et al., 2014) use verification loss functions based on the Euclidean distance or cosine similarity. A number of methods utilize a simple classification approach for training (Chang et al.,

2018; Su et al., 2017; Kalayeh et al., 2018; Zhao et al., 2017), and Euclidean distance is used in test time.

4 HYPERBOLIC NEURAL NETWORKS

In our work we strongly rely on the apparatus of hyperbolic neural networks developed in (Ganea et al., 2018). Hyperbolic networks are extensions of conventional neural networks in a sense that they generalize typical neural network operations to those in hyperbolic space using the formalism of Möbius gyrovector spaces. In this paper, the authors present the hyperbolic versions of feed-forward networks, multinomial logistic regression, and recurrent neural networks. In Appendix A we discuss the hyperbolic functions and layers used in hyperbolic neural networks. Similarly to the paper (Ganea et al., 2018), we use an additional hyperparameter c corresponding to the radius of the Poincaré ball, which is then defined in the following manner: $\mathbb{D}_c^n = \{\mathbf{x} \in \mathbb{R}^n : c\|\mathbf{x}\|^2 < 1, c \geq 0\}$. The corresponding conformal factor is then modified as $\lambda_{\mathbf{x}}^c = \frac{2}{1-c\|\mathbf{x}\|^2}$. In practice, the choice of c allows one to balance between hyperbolic and Euclidean geometries, which is made precise by noting that with $c \to 0$ all the formulas discussed below take their usual Euclidean form.

Hyperbolic averaging One important operation common in image processing is averaging of feature vectors, used, e.g., in prototypical networks for few–shot learning (Snell et al., 2017). In the Euclidean setting this operation takes the form $(\mathbf{x}_1, \ldots, \mathbf{x}_N) \to \frac{1}{N}\sum_i \mathbf{x}_i$. Extension of this operation to hyperbolic spaces is called the *Einstein midpoint* and takes the most simple form in *Klein* coordinates:

$$\mathrm{HypAve}(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \sum_{i=1}^{N} \gamma_i \mathbf{x}_i / \sum_{i=1}^{N} \gamma_i, \tag{2}$$

where $\gamma_i = \frac{1}{\sqrt{1-c\|\mathbf{x}_i\|^2}}$ are the Lorentz factors. Recall from the discussion in Section 2 that the Klein model is supported on the same space as the Poincaré ball, however the same point has different coordinate representations in these models. Let $\mathbf{x}_{\mathbb{D}}$ and $\mathbf{x}_{\mathbb{K}}$ denote the coordinates of the same point in the Poincaré and Klein models correspondingly. Then the following transition formulas hold.

$$\mathbf{x}_{\mathbb{D}} = \frac{\mathbf{x}_{\mathbb{K}}}{1 + \sqrt{1 - c\|\mathbf{x}_{\mathbb{K}}\|^2}}, \tag{3}$$

$$\mathbf{x}_{\mathbb{K}} = \frac{2\mathbf{x}_{\mathbb{D}}}{1 + c\|\mathbf{x}_{\mathbb{D}}\|^2}. \tag{4}$$

Thus, given points in the Poincaré ball we can first map them to the Klein model, compute the average using Equation (2), and then move it back to the Poincaré model.

Practical aspects of implementation While implementing most of the formulas described above is straightforward, we employ some tricks to make the training more stable.

- To ensure numerical stability we perform clipping by norm after applying the exponential map, which constrains the norm to not exceed $\frac{1}{\sqrt{c}}(1 - 10^{-3})$.
- Some of the parameters in the aforementioned layers are naturally elements of $\mathbb{D}_n^c$. While in principle it is possible to apply Riemannian optimization techniques to them (e.g., previously proposed Riemannian Adam optimizer (Becigneul & Ganea, 2019)), we did not observe any significant improvement. Instead, we parametrized them via ordinary Euclidean parameters which were mapped to their hyperbolic counterparts with the exponential map and used the standard Adam optimizer.

Gromov's δ-hyperbolicity A necessary parameter for embedding to Poincaré disk is its radius. In hyperbolic neural networks, one has a curvature parameter c, which is inversed squared disk radius: $r = \frac{1}{\sqrt{c}}$. For the Euclidean case, i.e., $c = 0$, the corresponding radius would be equal to infinity. The disk radius is closely related to the notion of Gromov's δ-hyperbolicity (Gromov, 1987), as we will show later in this section. Intuitively, this δ value shows 'how hyperbolic is a metric space'. For example, for graphs, δ represents how 'far' the graph is from a tree, which is known to be hyperbolic

(Fournier et al., 2015). Hence, we can compute the corresponding δ-hyperbolicity value to find the right Poincaré disk radius for an accurate embedding.

Formally, δ-hyperbolicity is defined as follows; we emphasize that this notion is defined for any metric space (X, d). First, we need to define *Gromov product* for points $x, y, z \in X$:

$$(y, z)_x = \frac{1}{2}(d(x, y) + d(x, z) - d(y, z)). \tag{5}$$

Then, the δ is the minimal value such that the following four-point condition holds for all points $x, y, z, w \in X$:

$$(x, z)_w \geq \min((x, y)_w, (y, z)_w) - \delta. \tag{6}$$

In practice, it suffice to find the δ for some fixed point w_0.

A more computational friendly way to define δ is presented in (Fournier et al., 2015). Having a set of points, we first compute the matrix A of pairwise Gromov products (5). After that, the δ value is simply the largest coefficient in the matrix $(A \otimes A) - A$, where $\otimes$ denotes the min-max matrix product

$$A \otimes B = \max_k \min\{A_{ik}, B_{kj}\}.$$

Relation between δ-hyperbolicity and Poincaré disk radius It is known (Tifrea et al., 2018) that the standard Poincaré ball is δ-hyperbolic with $\delta_P = \log(1 + \sqrt{2}) \sim 0.88$. Using this constant we can estimate the radius of Poincaré disk suitable for an embedding of a specific dataset. Suppose that for some dataset X we have found that its natural Gromov's δ is equal to δ_X. Then we can estimate $c(X)$ as follows.

$$c(X) = \left(\frac{\delta_P}{\delta_X}\right)^2. \tag{7}$$

Estimating hyperbolicity of a dataset In order to verify our hypothesis on hyperbolicity of visual datasets we compute the scale-invariant metric, defined as $\delta_{rel}(X) = \frac{2\delta(X)}{\text{diam}(X)}$, where $\text{diam}(X)$ denotes the set diameter (Borassi et al., 2015). By construction, $\delta_{rel}(X) \in [0, 1]$ and specifies how close is the dataset to a perfect hyperbolic space. For instance, trees which are discrete analogues of a hyperbolic space (under the natural shortest path metric) have δ_{rel} equal to 0. We computed δ_{rel} for various datasets we used for experiments. As a natural distance between images we used the standard Euclidean distance between the features extracted with VGG16 (Simonyan & Zisserman, 2014). Our results are summarized in Table 1. We observe that degree of hyperbolicity in image datasets is quite high, as the obtained δ_{rel} are significantly closer to 0 than to 1 (which corresponds to total non-hyperbolicity), which supports our hypothesis.

Table 1: The relative delta $2\delta(X)/\text{diam}(X)$ and curvature parameter values calculated for different datasets. For image datasets we measured the Euclidean distance between VGG16 features. S_2 and $S_2, z > 0$ denote the two–dimensional unit sphere and upper hemisphere correspondingly (1700 points were sampled from each one).

	Tree	Omniglot	CUB	*mini*ImageNet	S_2	$S_2, z > 0$
$2\delta(X)/\text{diam}(X)$	0	0.31	0.23	0.14	0.99	0.94
c	-	0.036	0.005	0.007	-	-

5 Experiments

Experimental setup We start with a toy experiment supporting our hypothesis that the distance to the center in Poincaré ball indicates a model uncertainty. To do so, we first train the MLR classifier in hyperbolic space on the MNIST dataset (LeCun et al., 1998) and evaluate it on the Omniglot dataset (Lake et al., 2013). We then investigate and compare the obtained distributions of distances to the origin of hyperbolic embeddings of the MNIST and Omniglot test sets.

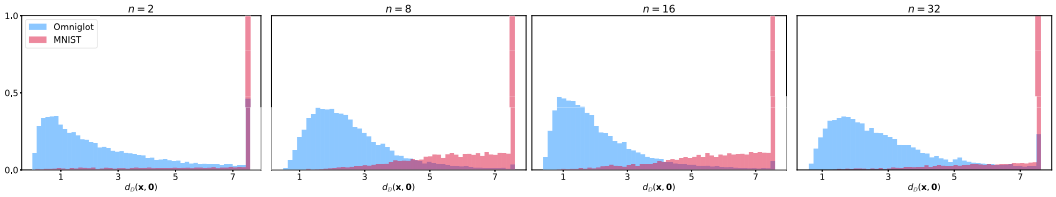

Figure 3: Distributions of the hyperbolic distance to the origin of the MNIST and Omniglot datasets embedded into the Poincaré ball. Embeddings are computed by a hyperbolic neural network trained for the MNIST classification task. We observe a significant difference between these distributions: embeddings of the Omniglot images are much closer to the origin. Table 2 provides the KS distances between the distributions.

In our further experiments, we concentrate on the few-shot classification and person re-identification tasks. The experiments on the Omniglot dataset serve as a starting point, and then we move towards more complex datasets. Afterwards, we consider two datasets, namely: *Mini*ImageNet (Ravi & Larochelle, 2016) and Caltech-UCSD Birds-200-2011 (CUB) (Wah et al., 2011). Here, for each dataset, we train four models: for one-shot five-way and five-shot five-way classification tasks both in the Euclidean and hyperbolic spaces. Finally, we provide the re-identification results for the two popular datasets: Market-1501 (Zheng et al., 2015) and DukeMTMD (Ristani et al., 2016; Zheng et al., 2017). Further in this section, we provide a thorough description of each experiment.

Our code is available at github[1].

5.1 Distance to the origin as the measure of uncertainty

In this subsection, we validate our hypothesis which claims that if one trains a hyperbolic classifier, then a distance of the Poincaré ball embedding of an image can serve as a good measure of confidence of a model. We start by training a simple hyperbolic convolutional neural network on the MNIST dataset. The output of the last hidden layer was mapped to the Poincaré ball using the exponential map (10) and was followed by the hyperbolic MLR layer. After training the model to $\sim 99\%$ test accuracy, we evaluate it on the Omniglot dataset (by resizing images to 28×28 and normalizing them to have the same background color as MNIST). We then evaluate the hyperbolic distance to the origin of embeddings produced by the network on both datasets. The closest Euclidean analogue to this approach would be comparing distributions of $p_{\max}$, maximum class probability predicted by the network. For the same range of dimensions we train ordinary Euclidean classifiers on MNIST, and compare these distributions for the same sets. Our findings are summarized in Figure 3 and Table 2. We observe that distances to the origin present a more statistically significant indicator of the dataset dissimilarity in 3 cases.

We have visualized the learned MNIST and Omniglot embeddings on Figure 1. We observe that more 'unclear' images are located near the center, while the images that are easy to classify are located closer to the boundary.

Table 2: Kolmogorov-Smirnov distances between the distributions of distance to the origin of the MNIST and Omniglot datasets embedded into the Poincaré ball with the hyperbolic classifier trained on MNIST, and between the distributions of $p_{\max}$ (maximum probablity predicted for a class) for the Euclidean classifier trained on MNIST and evaluated on the same sets. See further description in Subsection 5.1 and visualization on Figure 3. We observe that distance to the origin mostly presents a more statistically significant indicator of the dataset dissimilarity.

	$n = 2$	$n = 8$	$n = 16$	$n = 32$
$d_{\mathbb{D}}(\mathbf{x}, \mathbf{0})$	**0.868**	0.832	**0.853**	**0.859**
$p_{\max}(\mathbf{x})$	0.834	**0.835**	0.840	0.846

[1] https://github.com/hyperbolic-embeddings/hyperbolic-image-embeddings

5.2 OMNIGLOT FEW-SHOT CLASSIFICATION

We hypothesize that a certain class of problems – namely the few-shot classification task can benefit from hyperbolic embeddings. The starting point for our analysis is the experiments on the Omniglot dataset for few-shot classification. This dataset consists of the images of 1623 characters sampled from 50 different alphabets; each character is supported by 20 examples. We test several few-shot learning algorithms to see how hyperbolic embeddings affect them. In order to validate if hyperbolic embeddings can improve models performing on the state-of-the-art level, for the baseline architecture, we choose the prototype network (ProtoNet) introduced in the paper (Snell et al., 2017) with four convolutional blocks in a backbone. The specifics of the experimental setup can be found in B.

In ProtoNet, one uses a so-called *prototype representation* of a class, which is defined as a mean of the embedded support set of a class. Generalizing this concept to hyperbolic space, we substitute the Euclidean mean operation by HypAve, defined earlier in the Equation (2). Results are presented in Table 3. We can see that in some scenarios, in particular for one–shot learning, hyperbolic embeddings are more beneficial, while in other cases results are slightly worse. Relative simplicity of this dataset may explain why have not observed significant benefit of hyperbolic embeddings. We further test our approach on more advanced datasets.

Table 3: Few-shot classification accuracy values on Omniglot.

	ProtoNet	Hyperbolic ProtoNet
1-shot 5-way	98.2	**99.0**
5-shot 5-way	99.4	99.4
1-shot 20-way	95.8	**95.9**
5-shot 20-way	**98.6**	98.15

5.3 *MINI*IMAGENET FEW-SHOT CLASSIFICATION

*Mini*ImageNet dataset is the subset of ImageNet dataset (Russakovsky et al., 2015), which contains of 100 classes represented by 600 examples per class. We use the following split provided in the paper (Ravi & Larochelle, 2016): training dataset consists of 64 classes, validation dataset is represented by 16 classes, and the remaining 20 classes serve as a test dataset. As a baseline model, we again use prototype network (ProtoNet). We test the models on tasks for one-shot and five-shot classifications; the number of query points in each batch always equals to 15. All implementation details can be found in Appendix B.

Table 4: Experimental results on two datasets: *Mini*ImageNet and CUB averaged over $10,000$ test episodes and are reported with 95% confidence intervals.

Dataset	Model	c	1-shot 5-way	5-shot 5-way
*Mini*ImageNet	MatchNet (Vinyals et al., 2016)	-	43.56 ± 0.84	55.31 ± 0.73
	ProtoNet	-	48.29 ± 0.19	66.11 ± 0.16
	RelationNet (Sung et al., 2018)	-	50.44 ± 0.82	65.32 ± 0.70
	Hyperbolic ProtoNet	0.05	$\mathbf{51.57 \pm 0.2}$	66.27 ± 0.17
	Hyperbolic ProtoNet	0.007	47.97 ± 0.19	$\mathbf{68.92 \pm 0.16}$
CUB	ProtoNet	-	54.58 ± 0.24	68.04 ± 0.19
	Hyperbolic ProtoNet	0.05	$\mathbf{60.52 \pm 0.25}$	72.22 ± 0.19
	Hyperbolic ProtoNet	0.005	58.03 ± 0.24	$\mathbf{75.80 \pm 0.17}$

Table 4 illustrates the obtained results on *Mini*ImageNet dataset. For *Mini*ImageNet dataset, the results of the other models are available for the same classification tasks (i.e., for one-shot and five-shot learning). Therefore, we can compare our obtained results to those that were reported in the original papers. From these experimental results, we may observe a slight gain in model accuracy.

5.4 CALTECH-UCSD BIRDS FEW-SHOT CLASSIFICATION

The CUB dataset consists of $11,788$ images of 200 bird species and was designed for fine-grained classification. We use the split introduced in (Triantafillou et al., 2017): 100 classes out of 200 were used for training, 50 for validation and 50 for testing. Also, following (Triantafillou et al., 2017), we make the same pre-processing step by resizing each image to the size of 64×64. The implementation details can be found in B. Our findings on the experiments on the CUB dataset are summarized in Table 4. Interestingly, for this dataset, the hyperbolic version significantly outperforms its Euclidean counterpart.

5.5 PERSON RE-IDENTIFICATION

The DukeMTMC-reID dataset contains $16,522$ training images of 702 identities, 2228 query images of 702 identities and $17,661$ gallery images. Market1501 contains 12936 training images of 751 identities, 3368 queries of 750 identities and 15913 gallery images respectively. We report Rank1 of the Cumulative matching Characteristic Curve and Mean Average Precision for both datasets. We refer the reader to B for a more detailed description of the experimental setting. The results are reported after the 300 training epochs. As we can see in the Table 5, hyperbolic version generally performs better than the baseline, while the gap between the baseline and hyperbolic versions' results is decreasing for larger dimensionalities.

Table 5: Person re-identification results for Market-1501 and DukeMTMC-reID for the classification baseline (*bs*) and its hyperbolic counterpart (*hyp*). (See 5.5 for the details).

		Market-1501				DukeMTMC-reID			
		bs		hyp		bs		hyp	
dim	lr schedule	r1	mAP	r1	mAP	r1	mAP	r1	mAP
32	sch#1	71.4	49.7	69.8	45.9	56.1	35.6	56.5	34.9
	sch#2	68.0	43.4	**75.9**	**51.9**	57.2	35.7	**62.2**	**39.1**
64	sch#1	80.3	60.3	83.1	60.1	69.9	48.5	**70.8**	**48.6**
	sch#2	80.5	57.8	**84.4**	**62.7**	68.3	45.5	70.7	48.6
128	sch#1	86.0	67.3	**87.8**	**68.4**	74.1	53.3	**76.5**	**55.4**
	sch#2	86.5	68.5	86.4	66.2	71.5	51.5	74.0	52.2

6 DISCUSSION AND CONCLUSION

We have investigated the use of hyperbolic spaces for image embeddings. The models that we have considered use Euclidean operations in most layers, and use the exponential map to move from the Euclidean to hyperbolic spaces at the end of the network (akin to the normalization layers that are used to map from the Euclidean space to Euclidean spheres). The approach that we investigate here is thus compatible with existing backbone networks trained in Euclidean geometry.

At the same time, we have shown that across a number of tasks, in particular in the few-shot image classification, learning hyperbolic embeddings can result in a substantial boost in accuracy. We speculate that the negative curvature of the hyperbolic spaces allows for embeddings that are better conforming to the intrinsic geometry of at least some image manifolds with their hierarchical structure.

Future work may include several potential modifications of the approach. We have observed that the use of hyperbolic embeddings improves performance for some problems and datasets, while not helping others. A better understanding of when and why the use of hyperbolic geometry is justified is therefore needed. Also, we note that while all hyperbolic geometry models are equivalent in the continuous setting, fixed-precision arithmetic used in real computers breaks this equivalence. In practice, we observed that care should be taken about numeric precision effects (following (Ganea et al., 2018), we clip the embeddings to minimize numerical errors during learning). Using other models of hyperbolic geometry may result in more favourable floating point performance.

REFERENCES

Ejaz Ahmed, Michael J. Jones, and Tim K. Marks. An improved deep learning architecture for person re-identification. In *Conf. Computer Vision and Pattern Recognition, CVPR*, pp. 3908–3916, 2015. 3

Gary Becigneul and Octavian-Eugen Ganea. Riemannian Adaptive Optimization Methods. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=r1eiqi09K7. 4

Michele Borassi, David Coudert, Pierluigi Crescenzi, and Andrea Marino. On computing the hyperbolicity of real-world graphs. In *Algorithms-ESA 2015*, pp. 215–226. Springer, 2015. 5

Xiaobin Chang, Timothy M Hospedales, and Tao Xiang. Multi-level factorisation net for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2109–2118, 2018. 3

Bhuwan Dhingra, Christopher J Shallue, Mohammad Norouzi, Andrew M Dai, and George E Dahl. Embedding text in hyperbolic spaces. *arXiv preprint arXiv:1806.04313*, 2018. 2

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017. 3

Hervé Fournier, Anas Ismail, and Antoine Vigneron. Computing the Gromov hyperbolicity of a discrete metric space. *Information Processing Letters*, 115(6-8):576–579, 2015. 5

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*, pp. 5350–5360, 2018. 1, 2, 3, 4, 8, 13

Mikhael Gromov. Hyperbolic groups. In *Essays in group theory*, pp. 75–263. Springer, 1987. 4

Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. Hyperbolic attention networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJxHsjRqFQ. 3

Yiluan Guo and Ngai-Man Cheung. Efficient and deep person re-identification using multi-level similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2335–2344, 2018. 3

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 13

Mahdi M Kalayeh, Emrah Basaran, Muhittin Gökmen, Mustafa E Kamasak, and Mubarak Shah. Human semantic parsing for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1062–1071, 2018. 4

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015. 3

Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010. 2

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012. 1

Brenden M Lake, Ruslan R Salakhutdinov, and Josh Tenenbaum. One-shot learning by inverting a compositional causal process. In *Advances in Neural Information Processing Systems*, pp. 2526–2534, 2013. 5

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

Yann LeCun et al. Generalization and network design strategies. In *Connectionism in perspective*, volume 19. Citeseer, 1989. 1

Nathan Linial, Avner Magen, and Michael E Saks. Low distortion Euclidean embeddings of trees. *Israel Journal of Mathematics*, 106(1):339–348, 1998. 2

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1DmUzWAW. 3

Maximilian Nickel and Douwe Kiela. Learning continuous hierarchies in the Lorentz model of Hyperbolic geometry. In *Proc. ICML*, pp. 3776–3785, 2018. 3

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pp. 6338–6347, 2017. 2, 3

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4004–4012, 2016. 1

O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. 1

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016. 3, 6, 7

Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016. 6

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 7

Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International Conference on Machine Learning*, pp. 4457–4466, 2018a. 2

Frederic Sala, Christopher De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *Proc. ICML*, volume 80 of *JMLR Workshop and Conference Proceedings*, pp. 4457–4466. JMLR.org, 2018b. 3

Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pp. 355–366. Springer, 2011. 2, 3

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015. 1

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017. 1, 3, 4, 7

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pp. 1857–1865, 2016. 1

Chi Su, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Pose-driven deep convolutional model for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3960–3969, 2017. 4

Yumin Suh, Jingdong Wang, Siyu Tang, Tao Mei, and Kyoung Mu Lee. Part-aligned bilinear representations for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 402–419, 2018. 3

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018. 3, 7

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincaré GloVe: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018. 2, 5

Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. In *Advances in Neural Information Processing Systems*, pp. 2255–2265, 2017. 8

Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. In *Advances in Neural Information Processing Systems*, pp. 4170–4178, 2016. 1

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016. 1, 3, 7

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. 2011. 6

Yicheng Wang, Zhenzhong Chen, Feng Wu, and Gang Wang. Person re-identification with cascaded pairwise convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1470–1478, 2018. 3

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pp. 499–515. Springer, 2016. 1

Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2840–2848, 2017. 1

Dong Yi, Zhen Lei, and Stan Z Li. Deep metric learning for practical person re-identification. *arXiv prepzrint arXiv:1407.4979*, 2014. 1, 3

Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1077–1085, 2017. 4

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Computer Vision, IEEE International Conference on*, 2015. 6

Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 6

A HYPERBOLIC NEURAL NETWORKS

Möbius addition For a pair $\mathbf{x}, \mathbf{y} \in \mathbb{D}_c^n$, the Möbius addition is defined as follows:

$$\mathbf{x} \oplus_c \mathbf{y} \coloneqq \frac{(1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c\|\mathbf{y}\|^2)\mathbf{x} + (1 - c\|\mathbf{x}\|^2)\mathbf{y}}{1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}. \tag{8}$$

Distance The induced distance function is defined as

$$d_c(\mathbf{x}, \mathbf{y}) \coloneqq \frac{2}{\sqrt{c}} \operatorname{arctanh}(\sqrt{c}\| - \mathbf{x} \oplus_c \mathbf{y}\|). \tag{9}$$

Note that with $c = 1$ one recovers the geodesic distance (1), while with $c \to 0$ we obtain the Euclidean distance $\lim_{c \to 0} d_c(\mathbf{x}, \mathbf{y}) = 2\|\mathbf{x} - \mathbf{y}\|$.

Exponential and logarithmic maps To perform operations in the hyperbolic space, one first needs to define a bijective map from $\mathbb{R}^n$ to $\mathbb{D}_c^n$ in order to map Euclidean vectors to the hyperbolic space, and vice versa. The so–called exponential and (inverse to it) logarithmic map serve as such a bijection.

The *exponential* map $\exp_{\mathbf{x}}^c$ is a function from $T_{\mathbf{x}}\mathbb{D}_c^n \cong \mathbb{R}^n$ to $\mathbb{D}_c^n$, which is given by

$$\exp_{\mathbf{x}}^c(\mathbf{v}) \coloneqq \mathbf{x} \oplus_c \left(\tanh\left(\sqrt{c} \frac{\lambda_{\mathbf{x}}^c \|\mathbf{v}\|}{2} \right) \frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|} \right). \tag{10}$$

The inverse *logarithmic* map is defined as

$$\log_{\mathbf{x}}^c(\mathbf{y}) \coloneqq \frac{2}{\sqrt{c}\lambda_{\mathbf{x}}^c} \operatorname{arctanh}(\sqrt{c}\| - \mathbf{x} \oplus_c \mathbf{y}\|) \frac{-\mathbf{x} \oplus_c \mathbf{y}}{\| - \mathbf{x} \oplus_c \mathbf{y}\|}. \tag{11}$$

In practice, we use the maps $\exp_{\mathbf{0}}^c$ and $\log_{\mathbf{0}}^c$ for transition between the Euclidean and Poincaré ball representations of a vector.

Linear layer Assume we have a standard (Euclidean) linear layer $\mathbf{x} \to \mathrm{M}\mathbf{x} + \mathbf{b}$. In order to generalize it, one needs to define the Möbius matrix by vector product:

$$\mathrm{M}^{\otimes_c}(\mathbf{x}) \coloneqq \frac{1}{\sqrt{c}} \tanh\left(\frac{\|\mathrm{M}\mathbf{x}\|}{\|\mathbf{x}\|} \operatorname{arctanh}(\sqrt{c}\|\mathbf{x}\|) \right) \frac{\mathrm{M}\mathbf{x}}{\|\mathrm{M}\mathbf{x}\|}, \tag{12}$$

if $\mathrm{M}\mathbf{x} \neq \mathbf{0}$, and $\mathrm{M}^{\otimes_c}(\mathbf{x}) \coloneqq \mathbf{0}$ otherwise. Finally, for a bias vector $\mathbf{b} \in \mathbb{D}_c^n$ the operation underlying the hyperbolic linear layer is then given by $\mathrm{M}^{\otimes_c}(\mathbf{x}) \oplus_c \mathbf{b}$.

Concatenation of input vectors In several architectures (e.g., in siamese networks), it is needed to concatenate two vectors; such operation is obvious in Euclidean space. However, straightforward concatenation of two vectors from hyperbolic space does not necessarily remain in hyperbolic space. Thus, we have to use a generalized version of the concatenation operation, which is then defined in the following manner. For $\mathbf{x} \in \mathbb{D}_c^{n_1}$, $\mathbf{y} \in \mathbb{D}_c^{n_2}$ we define the mapping $\mathrm{Concat} : \mathbb{D}_c^{n_1} \times \mathbb{D}_c^{n_2} \to \mathbb{D}_c^{n_3}$ as follows.

$$\mathrm{Concat}(\mathbf{x}, \mathbf{y}) = \mathrm{M}_1^{\otimes_c}\mathbf{x} \oplus_c \mathrm{M}_2^{\otimes_c}\mathbf{y}, \tag{13}$$

where M_1 and M_2 are trainable matrices of sizes $n_3 \times n_1$ and $n_3 \times n_2$ correspondingly. The motivation for this definition is simple: usually, the Euclidean concatenation layer is followed by a linear map, which when written explicitly takes the (Euclidean) form of Equation (13).

Multiclass logistic regression (MLR) In our experiments, to perform the multiclass classification, we take advantage of the generalization of multiclass logistic regression to hyperbolic spaces. The idea of this generalization is based on the observation that in Euclidean space logits can be represented as the distances to certain *hyperplanes*, where each hyperplane can be specified with a point of origin and a normal vector. The same construction can be used in the Poincaré ball after a suitable analogue for hyperplanes is introduced. Given $\mathbf{p} \in \mathbb{D}_c^n$ and $\mathbf{a} \in T_{\mathbf{p}}\mathbb{D}_c^n \setminus \{\mathbf{0}\}$, such an analogue would be the union of all geodesics passing through $\mathbf{p}$ and orthogonal to $\mathbf{a}$.

The resulting formula for hyperbolic MLR for K classes is written below; here $\mathbf{p}_k \in \mathbb{D}_c^n$ and $\mathbf{a}_k \in T_{\mathbf{p}_k}\mathbb{D}_c^n \setminus \{\mathbf{0}\}$ are learnable parameters.

$$p(y=k|\mathbf{x}) \propto \exp\left(\frac{\lambda_{\mathbf{p}_k}^c \|\mathbf{a}_k\|}{\sqrt{c}} \operatorname{arcsinh}\left(\frac{2\sqrt{c}\langle -\mathbf{p}_k \oplus_c \mathbf{x}, \mathbf{a}_k\rangle}{(1-c\| -\mathbf{p}_k \oplus_c \mathbf{x}\|^2)\|\mathbf{a}_k\|}\right)\right).$$

For a more thorough discussion of hyperbolic neural networks, we refer the reader to the paper (Ganea et al., 2018).

B Implementation details

Omniglot As a baseline model, we consider the prototype network (ProtoNet). Each convolutional block consists of 3×3 convolutional layer followed by batch normalization, ReLU nonlinearity and 2×2 max-pooling layer. The number of filters in the last convolutional layer corresponds to the value of the embedding dimension, for which we choose 64. The hyperbolic model differs from the baseline in the following aspects. First, the output of the last convolutional block is embedded into the Poincaré ball of dimension 64 using the exponential map. The initial value of learning rate equals to 10^{-3} and is multiplied by 0.5 every 20 epochs out of total 60 epochs.

***mini*ImageNet** For this task we again considered ProtoNet as a baseline model. Similarly, number of filters the last convolutional layer corresponds to the varying value of the embedding dimension. In our experiments we set this value to 1024. We test the models on tasks for one-shot and five-shot classifications; the number of query points in each batch always equals to 15. We consider the following learning rate decay scheme: the initial learning rate equals to 10^{-3} and is further multiplied by 0.2 every 10 epochs (out of total 200 epochs).

The hyperbolic model differs from the baseline in the following aspects. First, the output of the last convolutional block is embedded into Poincaré ball of dimension 1024 using the exponential map defined in Equation (10). In ProtoNet, one uses a so-called *prototype representation* of a class, which is defined as a mean of the embedded support set of a class. Generalizing this concept to hyperbolic space, we substitute the Euclidean mean operation by HypAve, defined earlier in the Equation (2). The initial learning rate equals to 10^{-3} and is further multiplied by 0.2 every 10 epochs (out of total 200 epochs).

Caltech-UCSD Birds Likewise, we use ProtoNet mentioned above with the following modifications. Here, we fix the embedding dimension to 512 and use a slightly different setup for learning rate scheduler: the initial learning rate of value 10^{-3} is multiplied by 0.7 every 20 epochs out of total 100 epochs. Remaining architecture and parameters both in baseline and hyperbolic models are identical to those in the experiments on the *Mini*ImageNet dataset.

Person re-identification We use ResNet-50 (He et al., 2016) architecture with one fully connected embedding layer following the global average pooling. Three embedding dimensionalities are used in our experiments: 32, 64 and 128. For the baseline experiments, we add the additional classification linear layer, followed by the cross-entropy loss. For the hyperbolic version of the experiments, we map the descriptors to the Poincaré ball and apply multiclass logistic regression as described in Section 4. We found that in both cases the results are very sensitive to the learning rate schedules. We tried four schedules for learning 32-dimensional descriptors for both baseline and hyperbolic versions. Two best performing schedules were applied for the 64 and 128-dimensional descriptors. In these experiments, we also found that smaller c values give better results. We finally set c to 10^{-5}. Therefore, based on the discussion in 4, our hyperbolic setting is quite close to Euclidean. The results are compiled in Table 5. We set starting learning rates to $3 \cdot 10^{-4}$ and $6 \cdot 10^{-4}$ for $sch\#1$ and $sch\#2$ correspondingly and multiply them by 0.1 after each of the epochs 200 and 270.