Unsupervised Hyperbolic Representation Learning
via Message Passing Auto-Encoders

Jiwoong Park[*1]   Junho Cho[*1]   Hyung Jin Chang[2]   Jin Young Choi[1]

[1] ASRI, Dept. of ECE., Seoul National University   [2] School of Computer Science, University of Birmingham

{ptywoong,junhocho,jychoi}@snu.ac.kr, h.j.chang@bham.ac.uk

Abstract

Most of the existing literature regarding hyperbolic embedding concentrate upon supervised learning, whereas the use of unsupervised hyperbolic embedding is less well explored. In this paper, we analyze how unsupervised tasks can benefit from learned representations in hyperbolic space. To explore how well the hierarchical structure of unlabeled data can be represented in hyperbolic spaces, we design a novel hyperbolic message passing auto-encoder whose overall auto-encoding is performed in hyperbolic space. The proposed model conducts auto-encoding the networks via fully utilizing hyperbolic geometry in message passing. Through extensive quantitative and qualitative analyses, we validate the properties and benefits of the unsupervised hyperbolic representations. Codes are available at https://github.com/junhocho/HGCAE.

1. Introduction

A fundamental problem of machine learning is learning useful representations from high-dimensional data. There are many supervised representation learning methods that achieve good performances for downstream tasks [25, 22, 30, 60] on several data domains such as images and graphs. In recent years, with the success of deep learning, various large-scale real-world datasets have been collated [25, 24, 56, 49]. However, the larger these datasets and the closer they are to the real world, the expense and effort required to label the data increases proportionally. Thus, unsupervised representation learning is an increasingly viable approach to extract useful representation from real-world datasets.

Recently, many works [39, 40, 11, 16, 5, 1, 19] utilize hyperbolic geometry [23] to learn representations by understanding the underlying nature of the data domains. It is well known that complex networks contain latent hierarchies between large groups and the divided subgroups of nodes and

---

[*]equally contributed.

can be approximated as trees that grow exponentially with their depth [23]. Based on this fact, previous works which involve graphs [3, 38, 39, 40, 11, 36] showed the effectiveness of learning representation using hyperbolic spaces (a continuous version of trees) where distances increase exponentially when moving away from the origin. More recently, works [5, 27, 1] have been conducted which learn more powerful representations via conducting message passing (graph convolution) [12, 22, 54] in hyperbolic spaces.

In addition, it has been successfully shown that grafting hyperbolic geometry onto computer vision tasks is promising [19]. They observed a high degree of hyperbolicity [10] in the activations of image datasets obtained from pretrained convolutional networks. Also, it has been shown that the hyperbolic distance between learned embeddings and the origin of the Poincaré ball could be considered as a measurement of the model's confidence. Using these analyses, [19] added a single layer of hyperbolic neural networks [11] to deep convolutional networks and showed the benefits of hyperbolic embeddings on few-shot learning and person re-identification. Another work [28] also demonstrated the suitability of hyperbolic embeddings on zero-shot learning. However, most of the existing hyperbolic representation learning works [19, 28, 5, 27, 1] mainly focus on a supervised setting, and the effect of hyperbolic geometry on unsupervised representation learning has not been explored deeply so far [32, 15, 36].

In this paper, we explore the benefits of hyperbolic geometry to carry out unsupervised representation learning upon various data domains. Our motivation is to learn high-quality node embeddings of the graphs that are hierarchical and tree-like without supervision via considering the geometry of the embedding space. To do so, we present a novel hyperbolic graph convolutional auto-encoder (HGCAE) by combining hyperbolic geometry and message passing [12]. Every layer of HGCAE performs message passing in the hyperbolic space and its corresponding tangent space where curvature values can be trained. This is primarily in contrast to the Poincaré variational auto-encoder (P-VAE) [32] whose latent space is the Poincaré ball and conducts mes-

sage passing in Euclidean space. The HGCAE conducts auto-encoding the graphs from diverse data domains, such as images or social networks, in the hyperbolic space such as the Poincaré ball and hyperboloid. To fully utilize hyperbolic geometry for representation learning, we adopt a geometry-aware attention mechanism [16] when conducting message passing. Through extensive experiments and analyses using the learned representation in the hyperbolic latent spaces, we present the following observations on hierarchically structured data:

- The proposed auto-encoder, which combines message passing based on geometry-aware attention and hyperbolic spaces, can learn useful representations for downstream tasks. On various networks, the proposed method achieves state-of-the-art results on node clustering and link prediction tasks.

- Image clustering tasks can benefit from embeddings in hyperbolic latent spaces. We achieve comparable results to state-of-the-art image clustering results by learning representations from the activations of neural networks.

- Hyperbolic embeddings of images, the results of unsupervised learning, can recognize the underlying data structures such as a class hierarchy without any supervision of ground-truth class hierarchy.

- We show that the sample's hyperbolic distance from the origin in hyperbolic space can be utilized as a criterion to choose samples, therefore improving the generalization ability of a model for a given dataset.

2. Related Works

Hyperbolic embedding of images. Khrulkov et al. [19] validated hyperbolic embeddings of images via measuring the degree of hyperbolicity of image datasets. Many datasets such as CIFAR10/100 [24], CUB [56] and MiniImageNet [45] showed high degrees of hyperbolicity. In particular, the ImageNet dataset [47] is organized by following the hierarchical structure of WordNet [35]. These observations suggest that hyperbolic geometry can be beneficial in analyzing image manifolds by capturing not only semantic similarities but also hierarchical relationships between images. Furthermore, Khrulkov et al. [19] empirically showed that the distance between the origin and the image embeddings in the Poincaré ball can be regarded as the measure of the model's confidence. They observed that the samples which are easily classified are located near the boundary, while those more ambiguous samples lie near the origin of the hyperbolic space. Recent works of hyperbolic image embeddings [19, 28] add one or two layers of hyperbolic layers [11] after Euclidean convolutional networks.
Graph auto-encoding via hyperbolic geometry. Some recent works [15, 32, 51] attempted to auto-encode graphs

in hyperbolic space. Their models attempted to learn latent representations in the hyperbolic space via grafting hyperbolic geometry onto a variational auto-encoder model [20]. [15, 32] encoded the node representation via message passing [22] in Euclidean space, then the encoded representation was projected onto the hyperbolic space. Similar to these concurrent models, our auto-encoder framework learns latent node representations of the graph in hyperbolic latent spaces. Differing from these models, our work considers hyperbolic geometry throughout the auto-encoding process. Each encoder and decoder layer of the proposed model conducts message passing by utilizing geometry-aware attention in the hyperbolic space and its tangent space.

3. Hyperbolic Geometry

A real, smooth manifold $\mathcal{M}$ is a set of points x, that is locally similar to linear space. At each point $x \in \mathcal{M}$, the tangent space at x, $\mathcal{T}_x\mathcal{M}$, is a real vector space whose dimensionality is same as $\mathcal{M}$. A Riemannian manifold is defined as a tuple $(\mathcal{M}, g)$ that is possessing metric tensor $g_x : \mathcal{T}_x\mathcal{M} \times \mathcal{T}_x\mathcal{M} \to \mathbb{R}$ on the tangent space $\mathcal{T}_x\mathcal{M}$ at each point $x \in \mathcal{M}$ [43]. The metric tensor provides geometric notions such as geodesic, angle and volume. There exist mapping between the manifold and the tangent space: exponential map and logarithmic map. The exponential map $\exp_x : \mathcal{T}_x\mathcal{M} \to \mathcal{M}$ projects the vector on the tangent space $\mathcal{T}_x\mathcal{M}$ back to the manifold $\mathcal{M}$, while the logarithmic map $\log_x : \mathcal{M} \to \mathcal{T}_x\mathcal{M}$ is the inverse mapping of the exponential map as $\log_x(\exp_x(v)) = v$.

The hyperbolic space is a Riemannian manifold with constant negative sectional curvature equipped with hyperbolic geometry. This paper deals with two hyperbolic spaces; *'Poincaré ball'* and *'hyperboloid'*. The *Poincaré ball* $\mathbb{P}$ is highly effective for visualizing and analyzing the hyperbolic latent space. Meanwhile, the *hyperboloid* $\mathbb{H}$ can provide stable optimization since, unlike distance function of *Poincaré ball*, there is no division in the distance function [40]. A review of Riemannian geometry and details of the hyperboloid model are presented in the supplementary material.

Poincaré ball. The n-dimensional Poincaré ball with constant negative curvature $K(K < 0)$ $(\mathbb{P}_K^n, g_x^{\mathbb{P}_K})$ is defined:

$$\mathbb{P}_K^n = \{x \in \mathbb{R}^n : \|x\|^2 < -1/K\}, \tag{1}$$

where $\| \cdot \|$ denotes Euclidean norm. The metric tensor is $g_x^{\mathbb{P}_K} = (\lambda_x^K)^2 g_x^{\mathbb{E}}$, where $\lambda_x^K = \frac{2}{1+K\|x\|^2}$ is the conformal factor and $g_x^{\mathbb{E}} = \text{diag}([1, 1, \ldots 1])$ denotes Euclidean metric tensor. The origin of $\mathbb{P}_K^n$ is $\mathbf{o} = (0, \ldots, 0) \in \mathbb{R}^n$. The distance between two points $x, y \in \mathbb{P}_K^n$ is defined as

$$d_{\mathbb{P}_K^n}(x, y) = \frac{1}{\sqrt{-K}} \text{arcosh}\left(1 - \frac{2K\|x-y\|^2}{(1+K\|x\|^2)(1+K\|y\|^2)}\right). \tag{2}$$
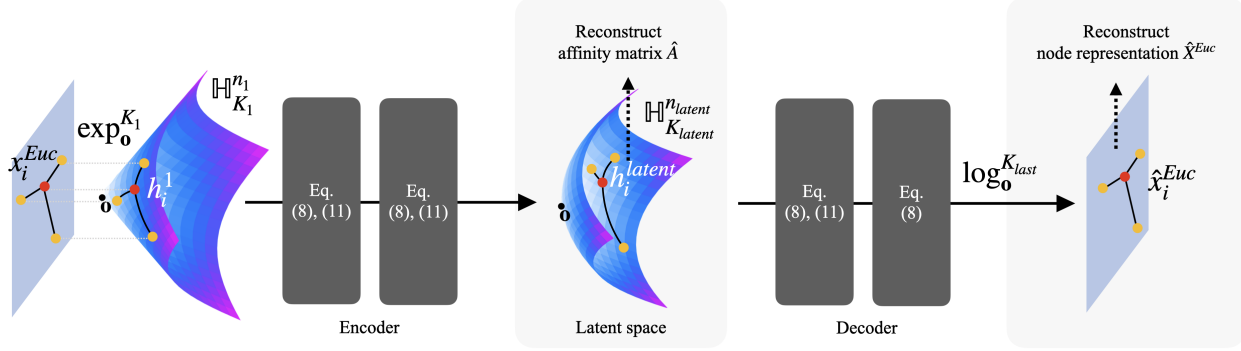
Figure 1: The overall architecture of HGCAE in a two-layer auto-encoder (*i.e.* the encoder and decoder have two layers each) whose hyperbolic space is hyperboloid. This figure describes three things: 1) how the node of the graph (red dot) conducts message passing (Eq. (8) and (11)) with its neighbors (yellow dot), 2) the process of embedding the output of encoder in hyperboloid latent space (blue-purple space), and 3) reconstruction of Euclidean node attributes at the end of the decoder.

For points $x \in \mathbb{P}^n_K$, tangent vector $v \in \mathcal{T}_x \mathbb{P}^n_K$, and $y \neq \mathbf{0}$, the exponential map $\exp_x : \mathcal{T}_x \mathbb{P}^n_K \rightarrow \mathbb{P}^n_K$ and the logarithmic map $\log_x : \mathbb{P}^n_K \rightarrow \mathcal{T}_x \mathbb{P}^n_K$ are defined as:

$$\exp_x^K(v) = x \oplus_K \left(\tanh(\frac{\sqrt{-K}\lambda_x^K \|v\|}{2}) \frac{v}{\sqrt{-K}\|v\|} \right), \quad (3)$$

$$\log_x^K(y) = \frac{2}{\sqrt{-K}\lambda_x^K} \operatorname{arctanh}\left(\sqrt{-K}\|u\|\right) \frac{u}{\|u\|}, \quad (4)$$

where $u = -x \oplus_K y$ and $\oplus_K$ denotes Möbius addition [53] for $x, y \in \mathbb{P}^n_K$ as

$$x \oplus_K y = \frac{(1 - 2K\langle x, y \rangle - K\|y\|^2)x + (1 + K\|x\|^2)y}{1 - 2K\langle x, y \rangle + K^2\|x\|^2\|y\|^2}. \quad (5)$$

Mapping between two models. Two hyperbolic models, Poincaré ball and hyperboloid, are equivalent and transformations between two models retain many geometric properties including isometry. There exist diffeomorphisms $p_{\mathbb{H} \rightarrow \mathbb{P}}$ and $p_{\mathbb{P} \rightarrow \mathbb{H}}$ between the two models, Poincaré ball $\mathbb{P}^n_K$ and hyperboloid $\mathbb{H}^n_K$ [27, 5], as follows:

$$p_{\mathbb{H} \rightarrow \mathbb{P}}(x_0, x_1, \ldots, x_n) = \frac{(x_1, \ldots, x_n)}{\sqrt{|K|}x_0 + 1}, \quad (6)$$

$$p_{\mathbb{P} \rightarrow \mathbb{H}}(x_1, \ldots, x_n) = \frac{(\frac{1}{\sqrt{|K|}}(1 - K\|x\|^2), 2x_1, \ldots, 2x_n)}{1 + K\|x\|^2}. \quad (7)$$

4. Methodology

HGCAE is designed to fully utilize hyperbolic geometry in the auto-encoding process along with leveraging the power of graph convolutions via a geometry-aware attention mechanism. Each layer conducts message passing in hyperbolic space whose curvature value is trainable. Before conducting message passing, we need to map the given input data points, x^{Euc}, defined in Euclidean space to the hyperbolic manifold. We map the Euclidean feature into hyperbolic manifold via $h_i^1 = \exp_{\mathbf{o}}^{K_1}(x_i^{Euc})$, where K_1 and h_i^1 denote a trainable curvature value and the i-th node's representation of the first layer respectively. When the hyperbolic

space is hyperboloid model, we use $(0, x^{Euc}) \in \mathbb{R}^{n+1}$ as an input of an exponential map as [5] did. The overall architecture of HGCAE is presented in Fig. 1.

4.1. Geometry-Aware Message Passing

Linear transformation. Message passing in the HGCAE consists of two steps: the linear transformation of a message and aggregating messages from neighbors. The i-th node's message passing result at the l-th layer z_i^l is as follows:

$$z_i^l = \exp_{\mathbf{o}}^{K_l} \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^l \left(W^l \log_{\mathbf{o}}^{K_l}(h_j^l) + b^l \right) \right), \quad (8)$$

where W^l, b^l, $\mathcal{N}(i)$, and α_{ij}^l denote a weight matrix, a bias term, the set of direct neighbors of node i including itself, and the relative importance (attention score) of the neighbor node j to the node i at the l-th layer respectively. Based on [11], we map the points in the hyperbolic manifold to the tangent space via the logarithmic map since the linear transformation cannot be performed directly in hyperbolic spaces. Then, the messages are linearly transformed on the tangent space of the origin in which inherits many properties of the ambient Euclidean space.

Aggregation. After performing linear transformation, we aggregate messages from neighbors via an attention mechanism. The majority of message passing algorithms that use attention mechanisms learn the relative importance of each node's neighbors based on node feature not only in Euclidean space [54] but also in hyperbolic space [5]. However, only considering node features for learning their relative importance does not take into account the geometry of the space, and this might result in an imprecise attention score. To make full use of the Riemannian metric of the hyperbolic manifolds, we adopt a geometry-aware attention mechanism [16] by utilizing the distance between linearly transformed node features on the hyperbolic space. Let $y_i^l = W^l \log_{\mathbf{o}}^{K_l}(h_i^l) + b^l$, then the attention score at the

l-th layer in Eq. (8) is:

$$\alpha_{ij}^l = \frac{\exp(-\beta^l d_{\mathcal{M}_{K_l}}^2(y_i^l, y_j^l) - \gamma^l)}{\sum_{p \in \mathcal{N}(i)} \exp(-\beta^l d_{\mathcal{M}_{K_l}}^2(y_i^l, y_p^l) - \gamma^l)}, \quad (9)$$

where $d_{\mathcal{M}_{K_l}}(\cdot, \cdot)$, β^l, and γ^l denote the distance on the hyperbolic space with curvature value K_l, and trainable parameters of the l-th layer respectively. After every step of message passing, we map the representation on the tangent space to the hyperbolic manifold via the exponential map.

4.2. Nonlinear Activation

The nonlinear activations, σ, such as ReLU can be directly applied to the points in the Poincaré ball, in contrast to the points on the hyperboloid [27]. Thus, when the hyperboloid model is used, we map the points to the Poincaré ball using Eq. (6) first. Next, we apply the nonlinear activation in the Poincaré ball and then return the result to the hyperboloid using the Eq. (7).

Since the curvature value of each layer in HGCAE is trainable, each layer can have different curvature values from other layers. Thus, a step for locating the result of the nonlinear activation in the hyperbolic space having a curvature value of the next layer is required. First, we map the results of the nonlinear activation to the tangent space of the current layer, $\mathcal{T}_{\mathbf{o}} \mathcal{M}_{K_l}$, using logarithmic map, $\log_{\mathbf{o}}^{K_l}$. Next, the points in the tangent space are mapped to the next layer's hyperbolic space via an exponential map of the next layer $\exp_{\mathbf{o}}^{K_{l+1}}$. The equations for performing such nonlinear activation and mapping to the hyperbolic space of the next layer in the cases of Poincaré ball and hyperboloid are as follows respectively:

$$h_i^{l+1} = \exp_{\mathbf{o}}^{K_{l+1}} \left(\log_{\mathbf{o}}^{K_l} \left(\sigma(z_i^l) \right) \right), \quad (10)$$

$$h_i^{l+1} = \exp_{\mathbf{o}}^{K_{l+1}} \left(\log_{\mathbf{o}}^{K_l} \left(p_{\mathbb{P} \to \mathbb{H}} (\sigma(p_{\mathbb{H} \to \mathbb{P}}(z_i^l))) \right) \right). \quad (11)$$

4.3. Loss Function

Our HGCAE reconstructs both the affinity matrix (graph structure) A and the Euclidean node attributes X^{Euc} at the end of the encoder and the decoder, respectively. To reconstruct the Euclidean node attributes $\hat{X}^{Euc}$, the aggregated representations in the hyperbolic space of the decoder's last layer are mapped to the tangent space of the origin $\mathcal{T}_{\mathbf{o}} \mathcal{M}$. Then, the loss of representations $\mathcal{L}_{REC-X}$ is defined as the mean square error between X^{Euc} and $\hat{X}^{Euc}$: $\frac{1}{N} \| X^{Euc} - \hat{X}^{Euc} \|^2$. For reconstructing the structure of the graph, the hyperbolic distance between the latent representations (the output of the encoder) of two nodes is utilized. To calculate the probability score of an edge which links between two nodes, we adopt the Fermi-Dirac distribution [23, 39], $\hat{A}_{ij} = [e^{(d_{\mathcal{M}_K}^2(h_i, h_j) - r)/t} + 1]^{-1}$, where h_i, $\hat{A}$, r, and t denote the latent representation of node i,

Table 1: Dataset statistics.

Dataset	Node	Edge	Attribute	Class
Phylogenetic [17, 48]	344	343	-	-
CS PhDs [9]	1,025	1,043	-	-
Diseases [13, 46]	516	1,188	-	-
Cora [49]	2,708	5,429	1,433	7
Citeseer [49]	3,312	4,552	3,703	6
Wiki [61]	2,405	17,981	4,973	17
Pubmed [49]	19,717	44,338	500	3
BlogCatalog [52]	5,196	171,743	8,189	6
Amazon Photo [33]	7,650	119,081	745	8
ImageNet-10 [7]	13,000	-	27,648	10
ImageNet-Dogs [7]	19,500	-	27,648	15
ImageNet-BNCR	11,700	-	27,648	9

the reconstructed affinity matrix, and hyperparameters respectively. The loss function for the affinity matrix is defined by the cross entropy loss with negative sampling: $\mathcal{L}_{REC-A} = \mathbb{E}_{q(H|X,A)}[\log p(\hat{A}|H)]$, where $q(H|X, A) = \prod_{i=1}^N q(h_i|X, A)$. The overall loss function of HGCAE is

$$\mathcal{L} = \mathcal{L}_{REC-A} + \lambda \mathcal{L}_{REC-X}, \quad (12)$$

where λ is a regularization parameter. λ serves to control the relative importance between the attributes and structure.

5. Experiments

This section explores the effectiveness of unsupervised hyperbolic embeddings on various data domains via quantitative and qualitative analyses. We use 9 real-world complex network datasets and 3 image datasets. The statistics of the datasets are summarized in Table 1. The details of the datasets, the compared methods, and the experimental details are described in the supplementary material. For node clustering and link prediction tasks on the 9 network datasets, we evaluate HGCAE-P and HGCAE-H, which denote HGCAE models whose latent spaces are Poincaré ball and hyperboloid, respectively. For the tasks of image clustering and visual data analysis, we use HGCAE-P because Poincaré ball is a powerful tool for visualizing and analyzing properties of hyperbolic visual embeddings.

5.1. Node Clustering and Link Prediction

Comparison to embeddings in Euclidean latent space. We evaluated the usefulness of hyperbolic representations by the performances of downstream tasks on citation [49, 61], social [52], and co-purchase [33] networks. We compared against the state-of-the-art unsupervised message passing models [21, 41, 57, 42, 66] which mainly conduct in Euclidean space. Similar to evaluation metrics used in [42], we used area under curve (AUC) and average precision (AP) to evaluate the performance of the link prediction task while using accuracy (ACC) and normalized mutual information (NMI) for evaluating the node clustering task.

The results of link prediction and node clustering are presented in Tables 2 and 3 respectively. From the results, we

Table 2: Link prediction performances.

	Cora		Citeseer		Wiki		Pubmed		BlogCatalog		Amazon Photo	
	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
GAE [21]	0.910	0.920	0.895	0.899	0.930	0.948	0.964	0.965	0.840	0.841	0.956	0.948
VGAE [21]	0.914	0.926	0.908	0.920	0.936	0.950	0.944	0.947	0.844	0.846	0.971	0.966
ARGA [41]	0.924	0.932	0.919	0.930	0.934	0.947	**0.968**	0.971	0.857	0.850	0.961	0.954
ARVGA [41]	0.924	0.926	0.924	0.930	0.947	0.948	0.965	0.968	0.837	0.828	0.927	0.909
GALA [42]	0.929	0.937	0.944	0.948	0.936	0.931	0.915	0.897	0.774	0.765	0.918	0.910
DBGAN [66]	0.945	0.951	0.945	0.958	-	-	**0.968**	0.973	-	-	-	-
HGCAE-P	0.948	0.947	0.960	0.963	**0.955**	**0.962**	0.962	0.960	**0.896**	**0.886**	**0.982**	**0.976**
HGCAE-H	**0.956**	**0.955**	**0.967**	**0.970**	0.952	0.958	0.962	0.960	0.857	0.850	0.972	0.966

Table 3: Node clustering performances.

	Cora		Citeseer		Wiki		Pubmed		BlogCatalog		Amazon Photo	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Kmeans [29]	0.492	0.321	0.540	0.305	0.417	0.440	0.595	0.315	0.180	0.007	0.267	0.122
GAE [21]	0.532	0.434	0.505	0.246	0.460	0.468	0.686	0.295	0.284	0.112	0.390	0.337
VGAE [21]	0.595	0.446	0.467	0.260	0.450	0.467	0.688	0.310	0.269	0.097	0.418	0.376
MGAE [57]	0.684	0.511	0.660	0.412	0.514	0.485	0.593	0.282	0.423	0.202	0.594	0.475
ARGA [41]	0.640	0.449	0.573	0.350	0.458	0.437	0.680	0.275	0.464	0.270	0.577	0.499
ARVGA [41]	0.638	0.450	0.544	0.261	0.386	0.338	0.513	0.116	0.450	0.250	0.455	0.395
GALA [42]	0.745	0.576	0.693	0.441	**0.544**	**0.503**	0.693	0.327	0.400	0.251	0.512	0.485
DBGAN [66]	0.748	0.560	0.670	0.407	-	-	0.694	0.324	-	-	-	-
HGCAE-P	0.746	0.572	0.693	0.422	0.459	0.467	**0.748**	**0.377**	0.550	0.325	0.781	0.696
HGCAE-H	**0.767**	**0.599**	**0.715**	**0.453**	0.530	0.435	0.711	0.347	**0.741**	**0.578**	**0.817**	**0.722**

Table 4: Link prediction task compared with P-VAE.

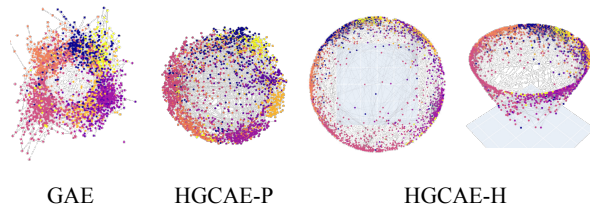|	Phylogenetic		CS PhDs		Diseases	
	AUC	AP	AUC	AP	AUC	AP
VGAE [21]	0.542	0.540	0.565	0.564	0.898	0.918
P-VAE [32]	0.590	0.555	0.598	0.567	0.923	**0.936**
HGCAE-P	**0.688**	**0.712**	**0.673**	**0.640**	**0.926**	0.914

GAE　　　　　HGCAE-P　　　　　HGCAE-H

Figure 2: 2-dimensional embeddings in Euclidean, Poincaré ball, and hyperboloid latent space on Cora dataset. Same color indicates same class. On hyperbolic latent spaces, most of the nodes are located on the boundary and well-clustered with the nodes in the same class.

can see that our HGCAE, with the representations of hyperbolic latent spaces, outperforms the existing methods, which use Euclidean latent spaces. Our superior results over their Euclidean counterparts support the fact that unsupervised learning with message passing benefit from the geometry of hyperbolic spaces. Due to space constraints, further analysis of the ablation study on the proposed architecture and the effectiveness of low-dimensional hyperbolic latent space are reported in the supplementary material.

Comparison to embeddings of hyperbolic graph auto-encoder. To validate the architecture of HGCAE, we compared its performance with the Poincaré variational auto-encoder (P-VAE) [32], whose latent space is the Poincaré ball and conducts its message passing in Euclidean space. Three networks, phylogenetic tree [17, 48], Ph.D. advisor-student relationships [9], and disease relationships [13, 46], were used for evaluating performance on link prediction. The latent space of both P-VAE and HGCAE-P is a 5-dimensional Poincaré ball. We report the results in Table 4. The proposed HGCAE-P outperforms P-VAE for most cases of the datasets since HGCAE-P considers hyperbolic geometry in the whole auto-encoding processes.

Visualization of citation network. We explored the latent representations of GAE [21] and our models on the Cora dataset [49] by constraining the latent space as a 2-dimensional hyperbolic or Euclidean space. The result is given in Fig. 2. On the results of HGCAE, most of the nodes are located on the boundary of hyperbolic spaces and well-clustered with the nodes in the same class. Further visualization of the network datasets is presented in the supplementary material.

5.2. Image Clustering

In this experiment, we illustrate that image clustering can benefit from hyperbolic geometry. The training sets of ImageNet-10 and ImageNet-Dogs [7], which are subsets of ImageNet [25], are used for evaluation. In the manner of the researches [11, 16, 19] which impose hyperbolic geometry on the activations of neural networks, we used the activations of PICA [18], one of the most recent models developed for deep image clustering. After obtaining activations from the pre-trained networks of PICA, we built the graph by mutual k nearest neighbors between activations. Then,
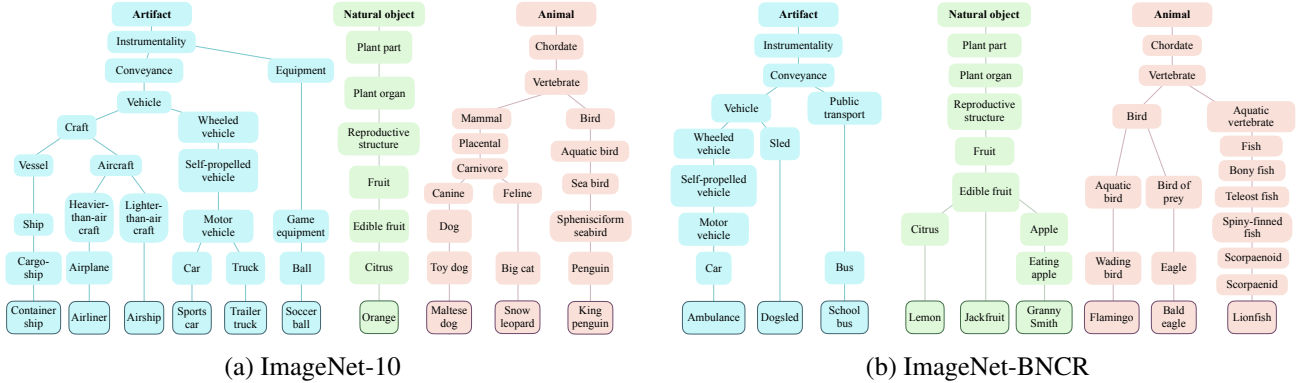
(a) ImageNet-10

(b) ImageNet-BNCR

Figure 3: Class hierarchy of ImageNet-10 and ImageNet-BNCR[1].

Table 5: Image clustering performances.

	ImageNet-10			ImageNet-Dogs		
	ACC	NMI	ARI	ACC	NMI	ARI
Kmeans [29]	0.241	0.119	0.057	0.105	0.055	0.020
SC [64]	0.274	0.151	0.076	0.111	0.038	0.013
AC [14]	0.242	0.138	0.067	0.139	0.037	0.021
NMF [4]	0.230	0.132	0.065	0.118	0.044	0.016
AE [2]	0.317	0.210	0.152	0.185	0.104	0.073
CAE [31]	0.253	0.134	0.068	0.134	0.059	0.022
SAE [37]	0.325	0.212	0.174	0.183	0.112	0.072
DAE [55]	0.304	0.206	0.138	0.190	0.104	0.078
DCGAN [44]	0.346	0.225	0.157	0.174	0.121	0.078
DeCNN [63]	0.313	0.186	0.142	0.175	0.098	0.073
SWWAE [65]	0.323	0.176	0.160	0.158	0.093	0.076
VAE [20]	0.334	0.193	0.168	0.179	0.107	0.079
JULE [62]	0.300	0.175	0.138	0.138	0.054	0.028
DEC [59]	0.381	0.282	0.203	0.195	0.122	0.079
DAC [7]	0.527	0.394	0.302	0.275	0.219	0.111
DDC [6]	0.577	0.433	0.345	-	-	-
DCCM [58]	0.710	0.608	0.555	0.383	0.321	0.182
PICA† [18]	0.850	0.782	0.733	0.324	0.336	0.179
PICA‡ [18]	0.828	0.763	0.692	0.352	0.353	0.201
PICA‡ [18]+HAE	0.821	0.759	0.686	0.338	0.347	0.200
PICA‡ [18]+GAE [21]	0.854	**0.792**	0.737	0.344	0.350	0.199
PICA‡ [18]+HGCAE-P	**0.855**	0.790	**0.741**	**0.387**	**0.360**	**0.226**

† Numbers from literature.
‡ Numbers from our experiments on the official pre-trained networks[2].

both the activations and the graph were used as inputs of HGCAE-P. Extensive baselines and state-of-the-art image clustering methods [29, 64, 14, 4, 2, 31, 37, 55, 44, 63, 65, 20, 62, 59, 7, 6, 58, 18] were compared. Furthermore, we also trained two auto-encoder models, GAE [21], and hyperbolic auto-encoder (HAE) whose layers are hyperbolic feed-forward layers [11]. The image clustering results are reported in Table 5. The metrics, ACC, NMI and Adjusted Rand Index (ARI), were used for evaluation. The results demonstrate that applying hyperbolic geometry along with using additional information of the approximated image manifold via nearest neighbor graphs can achieve better results than the Euclidean counterparts. We can also observe that HAE, the auto-encoder which naively applies hyperbolic geometry, does not work well, while our model per-

forms better via the message passing fully utilizing hyperbolic geometry.

5.3. Structure-Aware Unsupervised Embeddings

In this experiment, we observe the unsupervised hyperbolic image embeddings' ability to recognize the latent structure of visual datasets that have hierarchical structures. ImageNet [25] is constructed following the hierarchy of WordNet [35], therefore, its classes of ImageNet-10 [7] also have hierarchical structures. However, it is difficult to explore the effectiveness of hyperbolic embeddings since the classes of ImageNet-10 are biased to a certain root. Thus, we have constructed a new dataset, *ImageNet-BNCR*, that has a *B*alanced *N*umber of *C*lasses across *R*oots. For *ImageNet-BNCR*, we have chosen three roots, *Artifact, Natural objects*, and *Animal*, which have a large number of leaf classes. Each root contains balanced child nodes of {*Ambulance, Dogsled, School bus*}, {*Lemon, Jackfruit, Granny Smith*}, and {*Flamingo, Bald eagle, Lionfish*}, respectively. On the leaf classes of ImageNet-10, {*Container ship, Airliner, Airship, Sports car, Trailer truck, Soccer ball*}, {*Orange*}, and {*Maltese dog, Snow leopard, King penguin*} are the child nodes of the roots *Artifact, Natural objects*, and *Animal*, respectively. The class hierarchies of ImageNet-10 and ImageNet-BNCR are shown in Fig. 3.

We extracted 1000-dimensional features by training a convolutional auto-encoder (CAE) [31] on the ImageNet-10 and ImageNet-BNCR datasets. Then, after building the graph using mutual k nearest neighbors between extracted features, we trained three auto-encoder models (HGCAE-P, GAE [21], and HAE) whose latent space is 2-dimensional without the ground truth hierarchy structure of labels. The embedding results of the 1000-dimensional CAE features via UMAP [34] and three auto-encoders are presented in Fig. 4. We can observe that the embeddings of HGCAE-P are better clustered than others, according to the classes of each root in Fig. 3. On the ImageNet-10, in the same root *Artifact*, the embeddings of descendants of *Craft* and *Wheeled vehicle* are clustered respectively. The embeddings of the ImageNet-BNCR are clustered more distinctly ac-

---
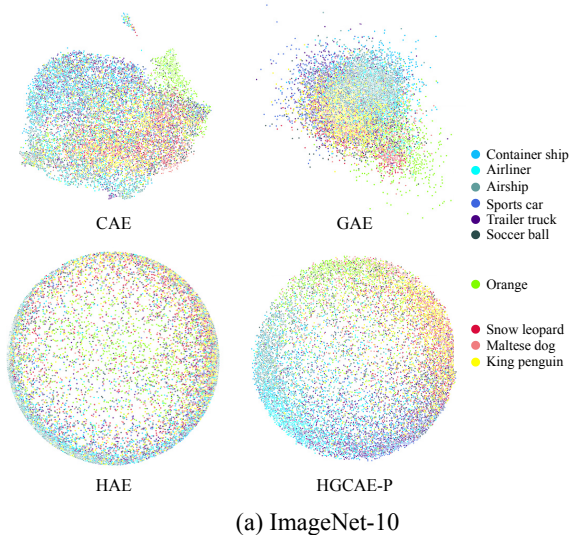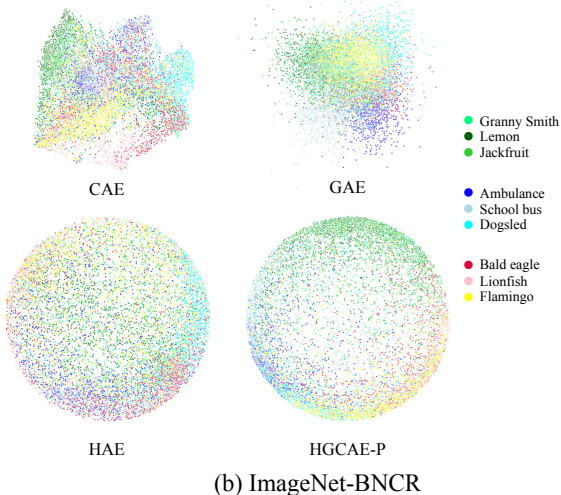
(a) ImageNet-10

(b) ImageNet-BNCR

Figure 4: 2-dimensional embeddings of CAE, GAE, HAE, and HGCAE-P on ImageNet-10 and ImageNet-BNCR. Hyperbolic representations belonging to the same root are close to each other near the boundary of the space.

cording to the root of class hierarchy than with ImageNet-10. On the other hand, the embeddings of the root *Natural objects*, {*Lemon, Jackfruit, Granny Smith*}, are located closer to each other since the geodesic distance between each leaf label is small. Our distinction from HAE implies that the additional information on image manifolds approximated by nearest neighbor graphs is helpful. In contrast to the representations of CAE and GAE, we can see that the hyperbolic representations belonging to the same root are located near the boundary of the space. In addition, to quantitatively validate the ability to recognize the latent hierarchical structure of the data without direct learning of label hierarchy, we cluster 2-dimensional embeddings of the three auto-encoders with three ground truth label settings according to the class hierarchy in Fig. 3: I. Root

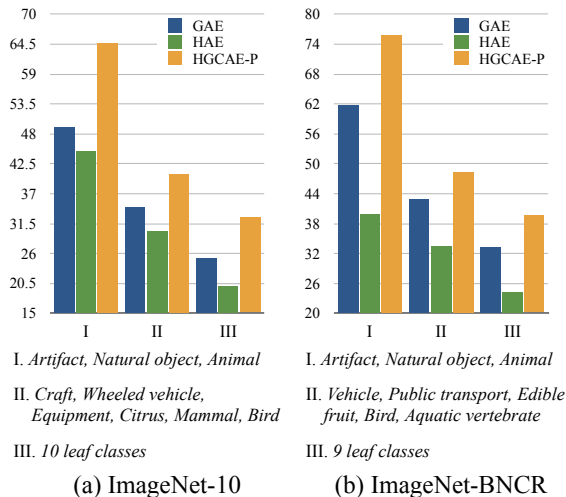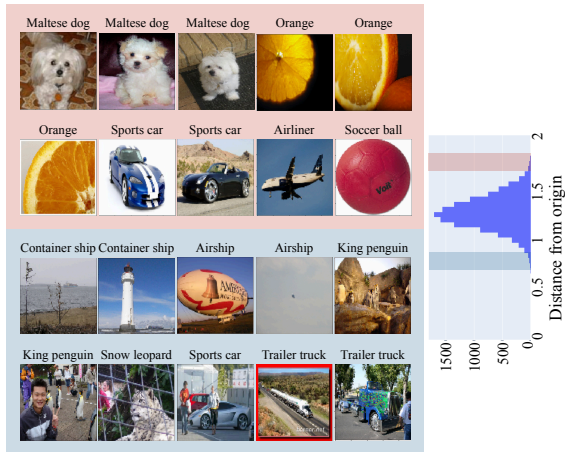(a) ImageNet-10   (b) ImageNet-BNCR

Figure 5: Clustering accuracy (%) according to the hierarchy of classes on ImageNet-10 and ImageNet-BNCR.

nodes, II. Internal nodes, and III. Leaf nodes. The quantitative results (clustering accuracy) on ImageNet-10 and ImageNet-BNCR are reported in Fig. 5. HGCAE-P outperforms GAE and HAE in every label hierarchy settings. This might be because the leaf classes whose parent is the same are closely embedded with each other. This analysis empirically demonstrates that unsupervised hyperbolic image embeddings can recognize the latent structure of the visual data that has a hierarchical structure.

5.4. Hyperbolic Distance to Filter Training Samples

In this experiment, we show that hyperbolic distance can help to choose training samples beneficial to the generalization ability of neural networks. To this end, we obtained the latent embeddings of ImageNet-10 [7] and ImageNet-BNCR via HGCAE-P model. Then, the hyperbolic distance (Eq. (2)) of each embedding from the origin was computed. Fig. 6 shows some samples near the boundary or near the origin in the histogram of the hyperbolic distance from embeddings to the origin. We can see that the samples near the boundary can be easily classified, whereas those near the origin are harder to classify. In general, the easy samples are not influential to learn an exact decision boundary. On the other hand, the hard samples make the decision boundary over-fitted, i.e., they work like noises located at the soft margin region near the decision boundary [8]. This illustration intuitively shows that the *H*yperbolic *D*istance from the *O*rigin (HDO) of a sample could give a clue which samples are influential or beneficial to learn the decision boundary crucial for the generalization ability of a classifier.

To verify this intuition, we conducted an experiment on the image classification task. On ImageNet-10 and ImageNet-BNCR, we trained the VGG-11 [50] classifier by adding further samples near the boundary/median of the dis-
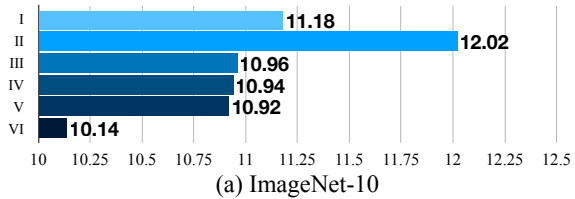
(a) ImageNet-10

(b) ImageNet-BNCR

Figure 6: Histogram and images according to the hyperbolic distance from the origin (HDO) on ImageNet-10 and ImageNet-BNCR. The feature of images inside red (blue) color box have high (low) HDO, so are located near the boundary (origin) of hyperbolic space.

tance histogram/origin to the original dataset in every training epoch and evaluated the network via each class' validation set in ImageNet [25]. We compared our results with six settings: I. Baseline: original data with cross-entropy loss, II. BaselineFL: original data with focal loss (FL) [26][3], III. Baseline + Random data adding, IV. Baseline + High HDO data adding, V. Baseline + Low HDO data adding, and VI. Baseline + Middle HDO data adding.

The image classification results are given in Fig. 7. As expected, the case V of adding low HDO data in the histogram show similar performances with the baseline. The case IV of adding high HDO data contributes the performance improvements, but the case VI of adding middle HDO data demonstrates the best performance among

---

[3]The focal loss tries to focus gradient updates on the samples that the classifier hard to classify.

(a) ImageNet-10
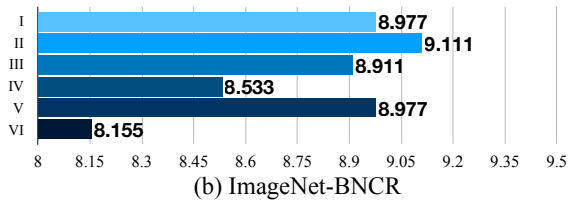
(b) ImageNet-BNCR

I. Baseline.
II. BaselineFL.
III. Baseline + Random data.
IV. Baseline + High HDO data.
V. Baseline + Low HDO data.
VI. Baseline + Middle HDO data.

Figure 7: Top-1 classification error (%) on ImageNet-10 and ImageNet-BNCR.

the compared settings. This result empirically verifies that the middle HDO samples are beneficial to learn a reasonable decision boundary which increases the generalization ability of a neural network. Since the supporting samples marginally apart from the decision boundary are crucial for the generalization performance [8], the HDO related with the generalization performance can be interpreted as a measure proportional to the distance of a sample from the decision boundary for a given classification task. In conclusion, we can utilize HDO as a criterion to choose samples for improving the generalization ability of a model for a given dataset.

6. Conclusion

In this paper, we explored the properties of unsupervised hyperbolic representations. We derived the representations from geometry-aware message passing autoencoders whose whole operations were conducted in hyperbolic spaces. Then, we conducted extensive experiments and analyses on the low-dimensional latent representations in hyperbolic spaces. The experimental results support the conclusion that taking advantage of hyperbolic geometry can improve the performances of unsupervised tasks; node clustering, link prediction, and image clustering. We observed that the proposed method could yield unsupervised hyperbolic image embeddings reflecting the latent structure of the visual datasets that have a hierarchical structure. Lastly, we demonstrated that the hyperbolic distance from origin for a sample could be utilized to determine the additional data crucial for a classifier's generalisation ability.

References

[1] Gregor Bachmann, Gary Bécigneul, and Octavian-Eugen Ganea. Constant curvature graph convolutional networks. *arXiv preprint arXiv:1911.05076*, 2019. 1

[2] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, pages 153–160, 2007. 6

[3] Guillaume Bouchard, Sameer Singh, and Theo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *2015 AAAI Spring Symposium Series*, 2015. 1

[4] Deng Cai, Xiaofei He, Xuanhui Wang, Hujun Bao, and Jiawei Han. Locality preserving nonnegative matrix factorization. In *IJCAI*, volume 9, pages 1010–1015, 2009. 6

[5] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 4869–4880, 2019. 1, 3

[6] Jianlong Chang, Yiwen Guo, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep discriminative clustering analysis. *arXiv preprint arXiv:1905.01681*, 2019. 6

[7] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5879–5887, 2017. 4, 5, 6, 7

[8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. 7, 8

[9] Wouter De Nooy, Andrej Mrvar, and Vladimir Batagelj. *Exploratory social network analysis with Pajek: Revised and expanded edition for updated software*, volume 46. Cambridge University Press, 2018. 4, 5

[10] 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. 1

[11] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*, pages 5345–5355, 2018. 1, 2, 3, 5, 6

[12] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272, 2017. 1

[13] Kwang-Il Goh, Michael E Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007. 4, 5

[14] K Chidananda Gowda and G Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2):105–112, 1978. 6

[15] Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Adversarial autoencoders with constant-curvature latent manifolds. *Applied Soft Computing*, 81:105511, 2019. 1, 2

[16] 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. 1, 2, 3, 5

[17] Wolfgang Karl Hofbauer, Laura Lowe Forrest, Peter M Hollingsworth, and Michelle L Hart. Preliminary insights from dna barcoding into the diversity of mosses colonising modern building surfaces. *Bryophyte Diversity and Evolution*, 38(1):1–22, 2016. 4, 5

[18] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8849–8858, 2020. 5, 6

[19] 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*, pages 6418–6428, 2020. 1, 2, 5

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 6

[21] Thomas N Kipf and Max Welling. Variational graph autoencoders. *NIPS Workshop on Bayesian Deep Learning*, 2016. 4, 5, 6

[22] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. 1, 2

[23] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010. 1, 4

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 2

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 1, 5, 6, 8

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 8

[27] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *Advances in Neural Information Processing Systems*, pages 8228–8239, 2019. 1, 3, 4

[28] 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*, pages 9273–9281, 2020. 1, 2

[29] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 5, 6

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1

[31] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hi-

erarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011. 6

[32] Emile Mathieu, Charline Le Lan, Chris J. Maddison, Ryota Tomioka, and Yee Whye Teh. Continuous hierarchical representations with poincaré variational auto-encoders. In *Advances in Neural Information Processing Systems*, 2019. 1, 2, 5

[33] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015. 4

[34] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 6

[35] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998. 2, 6

[36] Yoshihiro Nagano, Shoichiro Yamaguchi, Yasuhiro Fujita, and Masanori Koyama. A wrapped normal distribution on hyperbolic space for gradient-based learning. In *International Conference on Machine Learning*, pages 4693–4702, 2019. 1

[37] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011. 6

[38] Maximilian Nickel, Xueyan Jiang, and Volker Tresp. Reducing the rank in relational factorization models by including observable patterns. In *Advances in Neural Information Processing Systems*, pages 1179–1187, 2014. 1

[39] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pages 6338–6347, 2017. 1, 4

[40] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pages 3779–3788, 2018. 1, 2

[41] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2609–2615, 2018. 4, 5

[42] Jiwoong Park, Minsik Lee, Hyung Jin Chang, Kyuewang Lee, and Jin Young Choi. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6519–6528, 2019. 4, 5

[43] Peter Petersen, S Axler, and KA Ribet. *Riemannian geometry*, volume 171. Springer, 2006. 2

[44] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 6

[45] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016. 2

[46] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. 4, 5

[47] 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. 2

[48] MJ Sanderson, MJ Donoghue, W Piel, and T Eriksson. Treebase: a prototype database of phylogenetic analyses and an interactive tool for browsing the phylogeny of life. *American Journal of Botany*, 81(6):183, 1994. 4, 5

[49] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008. 1, 4, 5

[50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[51] Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. Mixed-curvature variational autoencoders. In *International Conference on Learning Representations*, 2019. 2

[52] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 817–826. ACM, 2009. 4

[53] Abraham Albert Ungar. A gyrovector space approach to hyperbolic geometry. *Synthesis Lectures on Mathematics and Statistics*, 1(1):1–194, 2008. 3

[54] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. 1, 3

[55] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(12), 2010. 6

[56] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 1, 2

[57] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. Mgae: Marginalized graph autoencoder for graph clustering. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 889–898. ACM, 2017. 4, 5

[58] Jianlong Wu, Keyu Long, Fei Wang, Chen Qian, Cheng Li, Zhouchen Lin, and Hongbin Zha. Deep comprehensive correlation mining for image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8150–8159, 2019. 6

[59] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487, 2016. 6

[60] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018. 1

[61] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Chang. Network representation learning with rich text

information. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 4

[62] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016. 6

[63] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2528–2535. IEEE, 2010. 6

[64] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2005. 6

[65] Junbo Zhao, Michael Mathieu, Ross Goroshin, and Yann Lecun. Stacked what-where auto-encoders. *arXiv preprint arXiv:1506.02351*, 2015. 6

[66] Shuai Zheng, Zhenfeng Zhu, Xingxing Zhang, Zhizhe Liu, Jian Cheng, and Yao Zhao. Distribution-induced bidirectional generative adversarial network for graph representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7224–7233, 2020. 4, 5