Supplementary material for "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

In this supplemental material, we present the reviews of Riemannian geometry and hyperboloid model firstly. Then, we explain the details of the datasets, compared methods, and experimental details. Finally, further experiments on network datasets and further discussions are presented.

1. Riemannian Geometry

1.1. A Review of Riemannian Geometry

A manifold $\mathcal{M}$ of n-dimension is a topological space that each point $x \in \mathcal{M}$ has a neighborhood that is homeomorphic to n-dimensional Euclidean space $\mathbb{R}^n$. For each point $x \in \mathcal{M}$, a real vector space $\mathcal{T}_x\mathcal{M}$ whose dimensionality is the same as $\mathcal{M}$ exists and is called a tangent space. The tangent space $\mathcal{T}_x\mathcal{M}$ is the set of all the possible directions and speeds of the curves on $\mathcal{M}$ across $x \in \mathcal{M}$. A Riemannian manifold is a tuple $(\mathcal{M}, g)$ that is possessing Riemannian metric $g_x : \mathcal{T}_x\mathcal{M} \times \mathcal{T}_x\mathcal{M} \to \mathbb{R}$ on the tangent space $\mathcal{T}_x\mathcal{M}$ at each point $x \in \mathcal{M}$ such that $\langle y, z\rangle_x = g_x(y, z) = y^T G(x) z$, where $G(x)$ is a matrix representation of Riemannian metric [27]. The metric tensor provides geometric notions such as the length of curve, angle and volume. The length of curve $\gamma : t \mapsto \gamma(t) \in \mathcal{M}$ is $L(\gamma) = \int_0^1 \|\gamma'(t)\|^{1/2}_{\gamma(t)}\, dt$. The geodesic, the generalization of straight line on Euclidean space, is the constant speed curves giving the shortest path between the pair of points $x, y \in \mathcal{M}$: $\gamma^* = \arg\min_\gamma L(\gamma)$ where $\gamma(0) = x$, $\gamma(1) = y$ and $\|\gamma'(t)\|_{\gamma(t)} = 1$. The global distance between two points $x, y \in \mathcal{M}$ is defined as $d_{\mathcal{M}}(x, y) = \inf_\gamma L(\gamma)$. For a tangent vector $v \in \mathcal{T}_x\mathcal{M}$ of $x \in \mathcal{M}$, there exists a unique unit speed geodesic γ such that $\gamma(0) = x$ and $\gamma'(0) = v$. Then, the corresponding exponential map is defined as $\exp_x(v) = \gamma(1)$. The inverse mapping of exponential map, the logarithmic map, is defined as $\log_x : \mathcal{M} \to \mathcal{T}_x\mathcal{M}$. Refer the website of footnote for good introduction of hyperbolic geometry[1].

*equally contributed.

[1]http://hyperbolicdeeplearning.com/simple-geometry-initiation/

1.2. Hyperboloid Model

The hyperbolic space is a Riemannian manifold with constant negative sectional curvature equipped with hyperbolic geometry, and the hyperboloid model is one of the multiple equivalent hyperbolic models. For $x, y \in \mathbb{R}^{n+1}$, the Lorentz inner product $\langle\cdot,\cdot\rangle_{\mathcal{L}}$ is defined as $\langle x, y\rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^n x_i y_i$. The n-dimensional hyperboloid with constant negative curvature $K (K < 0)$ is defined as $(\mathbb{H}^n_K, g^{\mathbb{H}_K}_x)$:

$$\mathbb{H}^n_K = \{x \in \mathbb{R}^{n+1} : \langle x, x\rangle_{\mathcal{L}} = 1/K, x_0 > 0\}. \quad (1)$$

The metric tensor is $g^{\mathbb{H}_K}_x = \mathrm{diag}([-1, 1, \ldots 1])$, and the origin of the hyperboloid model is $\mathbf{o} = (1/\sqrt{|K|}, 0, \ldots, 0) \in \mathbb{R}^{n+1}$. The distance between two points $x, y \in \mathbb{H}^n_K$ is defined as

$$d_{\mathbb{H}^n_K}(x, y) = \frac{1}{\sqrt{-K}} \mathrm{arcosh}(K\langle x, y\rangle_{\mathcal{L}}). \quad (2)$$

For points $x \in \mathbb{H}^n_K$, tangent vector $v \in \mathcal{T}_x\mathbb{H}^n_K$, and $y \neq \mathbf{0}$, $\exp_x : \mathcal{T}_x\mathbb{H}^n_K \to \mathbb{H}^n_K$ and $\log_x : \mathbb{H}^n_K \to \mathcal{T}_x\mathbb{H}^n_K$ are defined as

$$\exp^K_x(v) = \cosh(s)x + \sinh(s)\frac{v}{s}, \quad (3)$$

$$\log^K_x(y) = \frac{\mathrm{arcosh}(K\langle x, y\rangle_{\mathcal{L}})}{\sqrt{K^2\langle x, y\rangle^2_{\mathcal{L}} - 1}}(y - K\langle x, y\rangle_{\mathcal{L}} x), \quad (4)$$

where $s = \sqrt{-K}\|v\|_{\mathcal{L}}$ and $\|x\|_{\mathcal{L}} = \sqrt{\langle x, x\rangle_{\mathcal{L}}}$.

2. Datasets

2.1. Network Datasets

Phylogenetic tree [14, 32] models the generic heritage. CS PhDs [10] represents the relationship between Ph.D. candidates and their advisors in computer science fields. Diseases [12, 30] is a biological network expressing the relationship between diseases. Cora [33], Citeseer [33], Pubmed [33], and Wiki [41] are citation networks whose nodes are scientific papers or web pages and edges represent citation relationships between any two papers or links

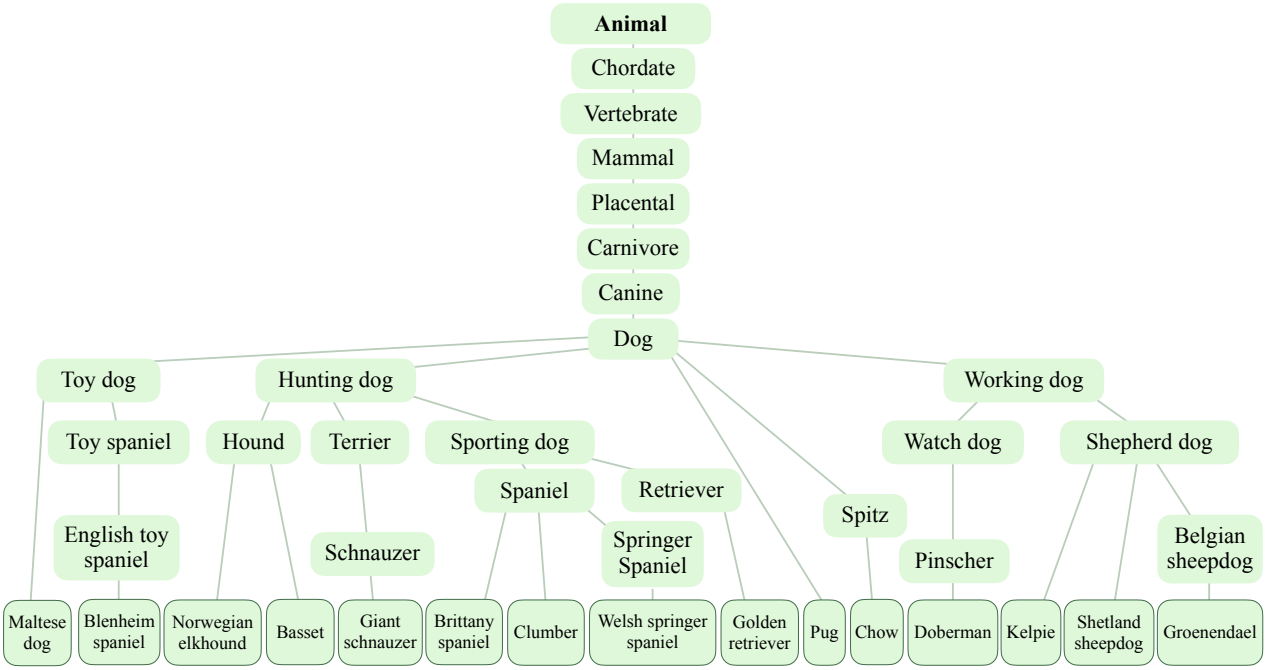

Figure 1: Class hierarchy of ImageNet-Dogs[2].

between any two web pages. BlogCatalog [35] models a social network among bloggers in the online community. Attribute and label of a node represent the description of each blog and the interest of a blogger, respectively. Amazon Photo [23] is a part of Amazon co-purchase networks whose nodes are goods and edges represent purchase correlations between any two goods. A node attribute indicates the bag-of-words for goods' reviews and its label denotes a product category.

2.2. Image Datasets

ImageNet-10 [7] and ImageNet-Dogs [7] are subsets of the ImageNet dataset [19]. ImageNet-10 consists of $13,000$ images from 10 randomly selected subjects. ImageNet-Dogs are $19,500$ images from 15 randomly selected dog breeds. The class hierarchy of ImageNet-Dogs is illustrated in Fig. 1. We have constructed a new dataset, ImageNet-BNCR, via randomly choosing 3 leaf classes per root. We chose three roots, *Artifacts*, *Natural objects*, and *Animal*. Thus, there exist 9 leaf classes, and each leaf class contains $1,300$ images in ImageNet-BNCR dataset. For every dataset used for the image clustering task, we used only the training set without the validation set, and images were resized to $96 \times 96 \times 3$.

[2] http://image-net.org/index

3. Compared Methods

3.1. Node Clustering and Link Prediction

We compared HGCAE with seven state-of-the-art unsupervised message passing models which mainly conduct in Euclidean space.

- **GAE [18], VGAE [18], ARGA [25],** and **ARVGA [25]** are graph auto-encoders that reconstruct only the affinity matrix using a non-parametric decoder which is not learnable.
- **MGAE [38]** is a stacked one-layer graph auto-encoder that reconstructs only the node attributes via a linear activation function.
- **GALA [26]** is a graph auto-encoder that reconstructs only the node attributes through learnable parametric encoder and decoder.
- **DBGAN [47]** is a distribution-induced bidirectional generative adversarial network that estimates the structure-aware prior distribution of the representations.

GAE [18], VGAE [18], ARGA [25], ARVGA [25], and GALA [26] are constrained to have two-layer auto-encoder models, since they report that two-layer structures show the best performances. In the case of MGAE [38] which is a stacked one-layer auto-encoder model, we have stacked the layer up to three and reported the best performances. For

DBGAN [47], we followed the number of layers in the literature. For every compared method, we followed the hyperparameters in the literature.

3.2. Image Clustering

Extensive baselines and state-of-the-art image clustering methods were compared. Several traditional methods including k-means clustering (Kmeans) [21], spectral clustering (SC) [45], agglomerative clustering (AC) [13], and nonnegative matrix factorization (NMF) [4] were also compared. For the representation-based clustering methods, AE [2], CAE [22], SAE [24], DAE [37], DCGAN [28], DeCNN [44], SWWAE [46], and VAE [17] were adopted. Besides, the state-of-the-art image clustering methods including JULE [42], DEC [40], DAC [7], DDC [6], DCCM [39], and PICA [15] were employed. For every compared method, we followed the experimental details in the literature.

4. Experimental Details

For every experiment and analysis, HGCAE has two encoder layers and two decoder layers. The dimension of each layer for HGCAE was set to one of $\{2^3, 2^4, ..., 2^{11}\}$. We optimized HGCAE using Adam [16] with learning rate 0.01. As reported in [5], we observe that Euclidean optimization [16] is much more stable than Riemannian optimization [1]. Because of exponential and logarithmic maps, the parameters of our model can be optimized using Euclidean optimization. We experimented with HGCAE for two cases, fixing the curvature of all layers or learning the curvature of each layer, then we reported the best performances. In the case of fixing the curvature of all layers, the curvature K was set to one of $\{-0.1, -0.5, -1, -2\}$. The regularization parameter λ of Eq. (12) in the manuscript was set to one of $\{10^{-6}, 10^{-5}, ..., 10^3\}$. The initial values of trainable parameters β and γ in Eq. (9) in the manuscript were set to 0. We searched the best hyperparameters which suited well to each dataset by random search. For visual datasets, we construct the mutual k nearest neighbors graph, A, as follows:

$$A_{ij} = \begin{cases} 1 & \text{if } x_i \in \mathrm{NN}_k(x_j) \wedge x_j \in \mathrm{NN}_k(x_i) \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where x_i and $\mathrm{NN}_k(x_i)$ denote the feature and k Euclidean nearest neighbor set of the i-th image respectively. We set $k = 20$ and $k = 10$ for ImageNet-10 and ImageNet-Dogs, respectively.

4.1. Details of Node Clustering and Link Prediction

For the link prediction task, we divided the edges into training edges, validation edges, and test edges as 85%, 5%, and 10%, then we used validation edges for the model convergence. During training for the link prediction task, we only reconstructed training edges in $\mathcal{L}_{REC-A} = \mathbb{E}_{q(H|X,A)}[\log p(\hat{A}|H)]$. For the node clustering task, every edge is reconstructed by the output of the encoder during training. The performance of node clustering was obtained by running k-means clustering [21] on the latent representations (output of the encoder) in the tangent space of the last layer of the encoder.

4.2. Details of Image Clustering

The performance of HGCAE on the image clustering task was obtained by running k-means clustering [21] on the latent representations (output of the encoder) in the tangent space of the last layer of the encoder.

4.3. Details of Convolutional Auto-Encoder

We extracted 1000-dimensional features by training a convolutional auto-encoder (CAE) [22] on the ImageNet-10 [7] and ImageNet-BNCR datasets on the experiment of Section 5.3 in the manuscript. We used the encoder part and decoder part as VGG-16 network [34] and five deconvolution layers [44] respectively. We optimized CAE using Adam [16] with learning rate 0.0001 and obtained the feature after 100 epochs.

4.4. Details of Image Classification

We obtained the latent representation of ImageNet-10 [7] and ImageNet-BNCR by training CAE on the experiments of Section 5.4 in the manuscript. For the image classification task, we trained the VGG-11 [34] classifier. We trained the classifier using stochastic gradient descent [3] and used the learning rate scheduler as in [43]. When adding further samples in every training epoch, high, middle, and low HDO samples were chosen by $n\%$ of the original data closest to the boundary, $n\%$ of the original data closest to the median of distance histogram, and $n\%$ of the original data closest to the origin, respectively. We set n for ImageNet-10 and ImageNet-BNCR to 30 and 50 respectively. The learning rates of ImageNet-10 and ImageNet-BCNR were set to 0.01 and 0.0005 respectively. When training BaselineFL, we tried $\{0.5, 1.0, 2.0\}$ for γ in focal loss [20] and reported the best performances. There has been recent research on manipulating the gradient updates based on the prediction difficulty, anchor loss (AL) [31], and we have tried to report the classification performance of AL as well as FL. However, due to the several NaN issues of official AL implementation[3], we could not report the performance of AL.

5. Further Experiments

5.1. Effectiveness of The Proposed Components

Through link prediction experiments, we validated the effectiveness of two components: learning in the hyperbolic

[3] https://github.com/slryou41/AnchorLoss

Table 1: Ablation studies on link prediction task: The baseline model is GAE which conducts graph convolution in Euclidean space, does not use an attention mechanism and reconstructs only the graph structure A.

	Reconstruct both A and X	Geometry aware attention	in hyperbolic space fixing K	in hyperbolic spaces learning K	Cora AUC	Cora AP	Citeseer AUC	Citeseer AP
Baseline: GAE [18]	×	×	×	×	91.0	92.0	89.5	89.9
Ablation I	√	×	×	×	92.7	92.1	94.0	94.8
Ablation II	√	×	√	×	94.6	94.4	95.9	96.3
Ablation III	×	√	√	×	94.5	94.8	96.1	96.4
Proposed I: HGCAE	√	√	√	×	95.4	**95.5**	**96.7**	**97.0**
Proposed II: HGCAE	√	√	×	√	**95.6**	**95.5**	96.5	96.8

Table 2: Clustering performances in low-dimensional space.

	Pubmed ACC	Pubmed NMI	BlogCatalog ACC	BlogCatalog NMI	Amazon Photo ACC	Amazon Photo NMI
GAE [18]	51.3	7.7	27.6	11.4	37.1	27.3
VGAE [18]	40.6	0.1	23.3	5.9	36.3	27.7
ARGA [25]	40.0	0.5	29.8	14.6	41.0	37.0
ARVGA [25]	38.5	0.1	27.2	9.7	40.8	27.8
GALA [26]	36.1	0.4	25.2	7.1	24.2	5.8
HGCAE	**68.1**	**28.2**	**74.1**	**57.8**	**76.3**	**64.0**

spaces and reconstructing both the graph structure and the node attributes. The experiment was conducted on two citation networks, Cora [33] and Citeseer [33], then the results for link prediction task are presented in Table 1. The baseline model is GAE [18], which conducts graph convolution in Euclidean space, does not use an attention mechanism, and reconstructs only the affinity matrix A. In Ablation I, reconstructing both the node attribute X^{Euc} and the graph structure A (Eq. (12) in the manuscript) are added to the baseline settings. In Ablation II, operating in hyperbolic space with fixed curvature K is added to Ablation I. In Ablation III, operating in hyperbolic space with fixed curvature K and the geometry-aware attention mechanism (Eq. (9) in the manuscript) are added to baseline settings. The results between Ablation I and Ablation II show that the message passing in the hyperbolic space is more effective than that in Euclidean space. Also, the performance gap between Ablation III and Proposed I shows that it is helpful to learn a representation that reflects both the structure of the network and the attributes of each node in hyperbolic space. This component is also valid in Euclidean space, as shown in the gap between Baseline and Ablation I. As shown in the gap between Proposed I and II, the fixed K and the trainable K show similar performance to each other for some datasets, but training K gives an efficient training scheme without multiple learning for searching the best K.

5.2. Learning in Low-Dimensional Space

One of the strengths of hyperbolic space compared to Euclidean space is that hyperbolic model can learn latent representation of data whose structure is hierarchical without the need for infeasible high-dimensional space [11]. To show this point, we obtained the latent representations of network datasets in the very low-dimensional latent space for node clustering task. Every compared graph autoencoder and HGCAE were constrained to have two layers whose each dimension was 4 and 2 respectively. Note that the performance of MGAE [38] cannot be reported since MGAE cannot manipulate the latent dimension. The experiments were conducted on Pubmed [33], BlogCatalog [35], and Amazon Photo [23] datasets. The results are presented in Table 2. Although the dimension of latent space is extremely low, HGCAE still significantly outperforms the

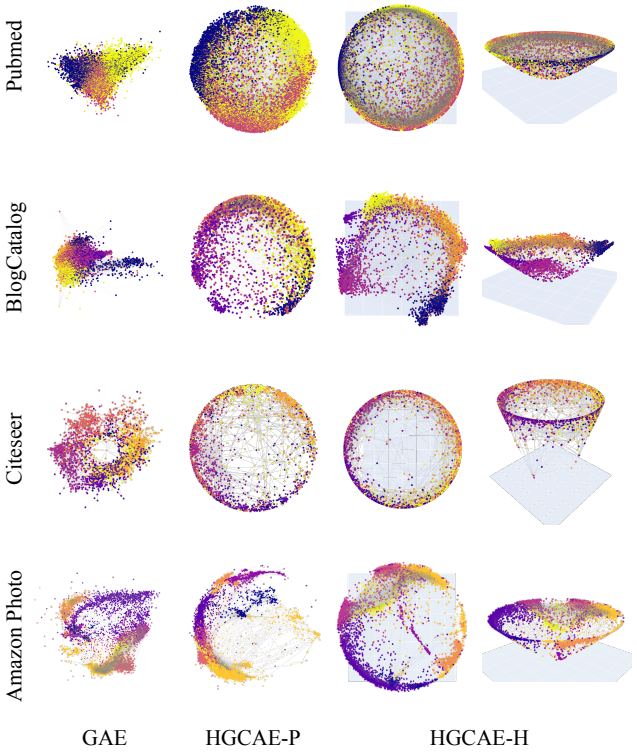

Figure 2: 2-dimensional embeddings in Euclidean, Poincaré ball, and hyperboloid latent spaces on Pubmed, BlogCatalog, Citeseer, and Amazon Photo datasets.

state-of-the-art unsupervised message passing methods operating in Euclidean space. Notably, on BlogCatalog and Amazon Photo datasets, HGCAE achieves more than 30% higher performances compared to Euclidean counterparts. These results support that hyperbolic space is effective than Euclidean space even in the very low-dimensional latent space.

5.3. Visualization of The Network Datasets

We explored the latent representations of GAE [18] and our models on Pubmed [33], BlogCatalog [35], Citesser [33], and Amazon Photo [23] datasets 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 space and well-clustered with the nodes in the same class.

5.4. Sensitivity of Hyperparameter Setting

One of the important hyperparameters of HGCAE is λ in Eq. (12) in the manuscript. If λ is required large (small) value, this means that the node attributes (subgraph structures) are the more important factor of latent representation. Since node attributes and the graph structure are different for each dataset, the optimal λ has different values for each dataset. In cases of BlogCatalog and Citeseer (Cora), we empirically found that small (large) λ value is optimal for both link prediction and node clustering tasks.

6. Further Discussions

6.1. Connection to Contrastive Learning

The hyperbolic geometry can be extended to contrastive learning [8]. A recent study [36] has uncovered the link between contrastive learning and deep metric learning. In this respect, it is becoming more significant to find the informative (hard) negative samples, embeddings that are difficult to distinguish from anchors, beyond uniform sampling [29]. Our work empirically showed that *H*yperbolic *D*istance from the *O*rigin (HDO) is an effective criterion for selecting samples without supervision for better generalization. The concept of HDO could be extended to informative negative sampling. Since the embeddings hard to discriminate is equal to those that are hard to classify by the model, the samples near the origin of hyperbolic space can be the impactful negative samples to increase the ability of the unsupervised contrastive learning.

6.2. Failure Cases of Hyperbolic Embedding Spaces

The inductive bias of hyperbolic representation learning is assuming that there exist hierarchical relationships in the dataset. Thus if the structure of the graph modeling the relation between data points is close to a tree, the hyperbolic space, a continuous version of a tree, is a suitable latent space. However, not all datasets' latent structures have the topological properties of the tree. For instance, datasets obtained from omnidirectional sensors of drones and autonomous cars are indeed more suitable to latent hyperspherical manifold rather than the hyperbolic manifold [9].

References

[1] Gary Becigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *International Conference on Learning Representations*, 2019. 3

[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. 3

[3] Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998. 3

[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. 3

[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. 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. 3

[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. 2, 3

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020. 5

[9] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *International Conference on Learning Representations*, 2018. 5

[10] 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. 1

[11] Christopher De Sa, Albert Gu, Christopher Ré, and Frederic Sala. Representation tradeoffs for hyperbolic embeddings. *Proceedings of machine learning research*, 80:4460, 2018. 4

[12] 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. 1

[13] K Chidananda Gowda and G Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2):105–112, 1978. 3

[14] 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. 1

[15] 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. 3

[16] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 3

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[18] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning*, 2016. 2, 4, 5

[19] 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. 2

[20] 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. 3

[21] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 3

[22] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011. 3

[23] 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. 2, 4, 5

[24] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011. 3

[25] 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. 2, 4

[26] 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. 2, 4

[27] Peter Petersen, S Axler, and KA Ribet. *Riemannian geometry*, volume 171. Springer, 2006. 1

[28] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 3

[29] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021. 5

[30] 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. 1

[31] Serim Ryou, Seong-Gyun Jeong, and Pietro Perona. Anchor loss: Modulating loss scale based on prediction difficulty. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5992–6001, 2019. 3

[32] 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. 1

[33] 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

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[35] 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. 2, 4, 5

[36] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations*, 2019. 5

[37] 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. 3

[38] 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. 2, 4

[39] 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. 3

[40] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487, 2016. 3

[41] 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. 1

[42] 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. 3

[43] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 3

[44] 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. 3

[45] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2005. 3

[46] Junbo Zhao, Michael Mathieu, Ross Goroshin, and Yann Lecun. Stacked what-where auto-encoders. *arXiv preprint arXiv:1506.02351*, 2015. 3

[47] 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. 2, 3