MHCN: A Hyperbolic Neural Network Model for Multi-view Hierarchical Clustering

Fangfei Lin[1,3]*, Bing Bai[3]†, Yiwen Guo[6]†, Hao Chen[4], Yazhou Ren[1], Zenglin Xu[2,5]†

[1]University of Electronic Science and Technology of China, China [2]Peng Cheng Lab, China

[3]Tencent Security Big Data Lab, China [4]University of California, Davis, USA

[5]Harbin Institute of Technology Shenzhen, China [6]Independent Researcher

{phoebe.lin1108,guoyiwen89,zenglin}@gmail.com,icebai@tencent.com

chen@ucdavis.edu,yazhou.ren@uestc.edu.cn

A. Appendix

A.1. Optimization Process of MHCN

Algorithm 1 presents the detailed optimization steps of the proposed MHCN.

Algorithm 1 Pseudocode to optimize our MHCN

Input: Multi-view dataset $\{\boldsymbol{X}^m \in \mathbb{R}^{N\times D_m}\}_{m=1}^M$;
Temperature parameters τ_{align} and τ_{uni};
Gaussian potential kernel parameter t;
Trade-off coeffi申ent α.

Output: Multi-view hierarchical clustering tree $\mathbf{T}$.

1: **Initialization**: Initialize the parameters of the hyperbolic autoencoders $\{\theta_{\text{enc}}, \theta_{\text{dec}}\} = \{\theta_{\text{enc}}^m, \theta_{\text{dec}}^m\}_{m=1}^M$.
2: **While** not reaching the maximal epochs **do**:
3: Update $\{\theta_{\text{enc}}, \theta_{\text{dec}}\}$ by $\mathcal{L}_{\text{total}}$ to learn $\{\boldsymbol{Z}_{\text{hyp}}^m \in \mathbb{B}^d\}$.
4: **End While**
5: Generate the common hyperbolic representations $\boldsymbol{Z}_{\text{hyp}}^* = \beta\text{-fusion}(\boldsymbol{Z}_{\text{hyp}}^1, \boldsymbol{Z}_{\text{hyp}}^2, \ldots, \boldsymbol{Z}_{\text{hyp}}^M)$.
6: Decoding $\mathbf{T}$ from $\boldsymbol{Z}_{\text{hyp}}^*$ by the bottom-up decoding strategy.

A.2. Description of Notations

To be clear, The notations and corresponding definitions used in this paper are summarized in Table 1.

A.3. Riemannian Geometry

In this section, we give a more detailed review of Riemannian geometry and Riemannian manifolds to keep this paper self-contained. Note that we denote the Euclidean inner product and norm for any real vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ as $\langle \boldsymbol{x}, \boldsymbol{y}\rangle$ and $||\boldsymbol{x}||$. An n-dimensional *manifold* $\mathcal{M}$ [5, 15] is a real and smooth space, which can be locally approximated to a linear n-dimensional Euclidean space $\mathbb{R}^n$ at each point $\boldsymbol{x} \in \mathcal{M}$. Giving an example in the real world, the earth can be modeled as a hypersphere from the global perspective which is a smooth manifold, while its local area can be regarded as a flat plane which can be approximated by 2-dimensional Euclidean space $\mathbb{R}^2$. Associated with any point $\boldsymbol{x}$ of the manifold $\mathcal{M}$, the corresponding local space is called the *tangent space* at the point $\boldsymbol{x}$, denoted as $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$. It represents an n-th dimensional space of all directions in which a smooth path on the manifold $\mathcal{M}$ can tangentially pass through $\boldsymbol{x}$, which is isomorphic to $\mathbb{R}^n$. On each tangent space $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$, the corresponding *metric tensor* $g_{\boldsymbol{x}} : \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \times \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \to \mathbb{R}$ defines an inner product on $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$, and the matrix form $G(\boldsymbol{x})$ of the metric tensor is represented as $g_{\boldsymbol{x}}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T G(\boldsymbol{x})\boldsymbol{v}$, where $\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \times \mathcal{T}_{\boldsymbol{x}}\mathcal{M}$. The norm of a vector $\boldsymbol{z}$ in tangent space can be also derived from the inner product, denoted as $||\boldsymbol{z}||_{\boldsymbol{x}} = \sqrt{|\langle \boldsymbol{z}, \boldsymbol{z}\rangle_{\boldsymbol{x}}|}$. Then, a *Riemannian metric* $g = (g_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathcal{M}}$ can be defined as a collection of inner products on the associated tangent space. By means of the metric tensor, the local geometric attributions of angle, length of curves, surface area and volume, can be integrated to derive global quantities the manifold. In this way, a *Riemannian manifold* can be defined as a matching tuple $(\mathcal{M}, g)$, where a manifold $\mathcal{M}$ is prepared with a Riemannian metric g.

The concept of a geodesic is generalized to the shortest path passing through two data points $\boldsymbol{x}, \boldsymbol{y}$ on manifold $\mathcal{M}$ via a constant speed vector, in analogy with the concept of a straight line in Euclidean space. Let $\gamma : a \to b \in \mathcal{M}$ denotes a curve on the manifold $\mathcal{M}$, which is defined by the *length* of γ, $L(\gamma) = \int_a^b |\gamma'(t)|_{\gamma(t)}^{\frac{1}{2}} dt$ [6]. Therefore, the geodesic distance is a smooth path γ of minimal length between two points $\boldsymbol{x}$ and $\boldsymbol{y}$ on the manifold $\mathcal{M}$, defined as $d_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{y}) = \inf L(\gamma)$, where inf represents all possible

*This work was done when Fangfei Lin was an intern at Tencent.
†Corresponding authors.

Notation	Definition
$\boldsymbol{X}$	The set of data samples.
$\boldsymbol{X}^m$	The m-th view of multi-view data.
$\boldsymbol{Z}^m_{\text{tan}}$	The Euclidean features in tangent space of the m-th view.
$\boldsymbol{Z}^m_{\text{hyp}}$	The hyperbolic latent codes of the m-th view.
$\boldsymbol{Z}^*_{\text{hyp}}$	The concatenated common hyperbolic representations.
$\hat{\boldsymbol{X}}^m$	The reconstructed data samples of the m-th view.
$\boldsymbol{x}^m_i$	The i-th input data sample of m-th view.
$\boldsymbol{z}^m_i$	The i-th latent feature of m-th view.
$\mathbf{T}$	The decoding multi-view hierarchical clustering tree.
N	The number of data samples.
M	The number of views.
D_m	The dimensionality of the m-th view.
d	The dimensionality of latent hyperbolic space.

Table 1: Notations used in MHCN

curves γ from the point $\boldsymbol{x}$ to the point $\boldsymbol{y}$.

The parallel transport $P_{\boldsymbol{x}\to\boldsymbol{y}} : \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \to \mathcal{T}_{\boldsymbol{y}}\mathcal{M}$ from $\boldsymbol{x}$ to $\boldsymbol{y}$ is defined as a linear isometry from $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ to $\mathcal{T}_{\boldsymbol{y}}\mathcal{M}$, moving a tangent vector in $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ along the geodesic from $\boldsymbol{x}$ to $\boldsymbol{y}$ in a parallel way. In order to project a tangent vector in $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ onto $\mathcal{M}$ along a geodesic with constant velocity, the *exponential map* $\exp_{\boldsymbol{x}} : \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \to \mathcal{M}$ is given. The *logarithmic map* $\log_{\boldsymbol{x}} : \mathcal{M} \to \mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ is the inverse form of the exponential map, projecting a vector from $\mathcal{M}$ back to $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ [17, 13].

A.4. Poincaré Ball Model of Hyperbolic Geometry

Hyperbolic Geometry is a non-Euclidean geometry with constant negative curvature, which satisfies all five basic rules in Euclidean Geometry only except the fifth parallel postulate [2]. Hence, the volume of the hyperbolic space grows exponentially with its radius in finite dimensions, which allows a meaningful compacted hierarchical structure naturally.

To describe this mathematically, an n-dimensional hyperbolic space $\mathbb{H}^n$ can be established through several isometric models, e.g., the basic Lorentz (hyperboloid) model, the Poincaré ball model and the Poincaré half space model [2]. We choose to perform our model on the n-dimensional Poincaré ball model with a constant negative curvature -1, denoted as $(\mathbb{B}^n, g^{\mathbb{B}})$, where $\mathbb{B}^n = \{\boldsymbol{x} \in \mathbb{R}^n : ||\boldsymbol{x}||^2 < 1\}$ is an open ball of curvature -1, and its hyperbolic metric tensor $g^{\mathbb{B}}_{\boldsymbol{x}} = \lambda^2_{\boldsymbol{x}} g^E$ is conformal to the Euclidean one. $\lambda^2_{\boldsymbol{x}} = \frac{2}{1-||x||^2}$ is a conformal factor, and $g^E = \boldsymbol{I}_n$ denotes the dot product in the Euclidean space. The distance $d_{\mathbb{B}}(\boldsymbol{x}, \boldsymbol{y})$ between two points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{B}^n$ is given by

$$d_{\mathbb{B}}(\boldsymbol{x}, \boldsymbol{y}) = \cosh^{-1}(1 + 2\frac{||\mathrm{x}-\mathrm{y}||^2}{(1-||\mathrm{x}||^2)(1-||\mathrm{y}||^2)}). \quad (1)$$

Given $\boldsymbol{z}, \boldsymbol{z'} \in \mathbb{B}^n$ and $\boldsymbol{t} \in \mathcal{T}_{\boldsymbol{z}}\mathbb{B}^n$, the exponential map $\exp_{\boldsymbol{z}} : \mathcal{T}_{\boldsymbol{z}}\mathbb{B}^n \to \mathbb{B}^n$ and the logarithm map $\log_{\boldsymbol{z}} : \mathbb{B}^n \to \mathcal{T}_{\boldsymbol{z}}\mathbb{B}^n$ realize the projection from the Euclidean space onto the Poincaré ball and vice versa, respectively. To enable the mathematical operations for hyperbolic space models, the framework of gyrovector spaces provides the algebraic setting for the hyperbolic geometry, with the $M\ddot{o}bius\ Addition$ $\oplus$ for any $\boldsymbol{z}, \boldsymbol{z'} \in \mathbb{B}^n$ as

$$\boldsymbol{z} \oplus \boldsymbol{z'} = \frac{(1 + 2\langle \boldsymbol{z}, \boldsymbol{z'}\rangle + ||\boldsymbol{z'}||^2)\boldsymbol{z} + (1 - ||\boldsymbol{z}||^2)\boldsymbol{z'}}{1 + 2\langle \boldsymbol{z}, \boldsymbol{z'}\rangle + ||\boldsymbol{z}||^2||\boldsymbol{z'}||^2}. \quad (2)$$

With the $M\ddot{o}bius\ Addition$ $\oplus$ [19], the closed-form expressions of $\exp_{\boldsymbol{z}}$ and $\log_{\boldsymbol{z}}$ on the Poincaré ball are respectively given by

$$\begin{aligned} \exp_{\boldsymbol{z}}(\boldsymbol{t}) &= \boldsymbol{z} \oplus (\tanh(\frac{\lambda_{\boldsymbol{z}}||\boldsymbol{t}||}{2})\frac{\boldsymbol{t}}{||\boldsymbol{t}||}), \\ \log_{\boldsymbol{z}}(\boldsymbol{z'}) &= \frac{2}{\lambda_{\boldsymbol{z}}}\operatorname{arctanh}(||-\boldsymbol{z} \oplus \boldsymbol{z'}||)\frac{-\boldsymbol{z} \oplus \boldsymbol{z'}}{||-\boldsymbol{z} \oplus \boldsymbol{z'}||}. \end{aligned} \quad (3)$$

For convenience in practice, $\boldsymbol{z}$ is usually set to the origin $\mathbf{0}$, so the exponential and the logarithm maps can be simplified as

$$\begin{aligned} \exp_{\mathbf{0}}(\boldsymbol{t}) &= \tanh(||\boldsymbol{t}||)\frac{\boldsymbol{t}}{||\boldsymbol{t}||}), \\ \log_{\mathbf{0}}(\boldsymbol{z'}) &= \operatorname{arctanh}(||\boldsymbol{z'}||)\frac{\boldsymbol{z'}}{||\boldsymbol{z'}||}. \end{aligned} \quad (4)$$

With the help of the above mapping operations $\exp_{\mathbf{0}}(\boldsymbol{t})$ and $\log_{\mathbf{0}}(\boldsymbol{z'})$, our model is able to perform the basic transformations of the latent representations between the Euclidean space and the hyperbolic space.

A.5. The Details of Datasets

The detailed information about the datasets is shown in Table 2. We conduct our experiments on six widespread multi-view datasets [10, 21, 18, 14], including four regular-scale datasets (i.e., MNIST-USPS, BDGP, Caltech, COIL-20, and BBCSport) and two large-scale datasets (i.e., Multi-Fashion and NR-MNIST).

- MNIST-USPS [14] is a common handwritten digital dataset with 5,000 images size from 10 categories(0-9), where the digits with 28×28 dimensions from MNIST and those with 16×16 dimensions from USPS are treated as two views. The MNIST and USPS views are both randomly sampled from the MNIST and USPS datasets respectively. The way of constructing the MNIST-USPS dataset is to pick pairs of individual objects from the corresponding classes of multiple different datasets, i.e., MNIST and USPS datasets. The above construction strategy is also applied for Multi-Fashion and NR-MNIST datasets.

- BDGP [9] is also a popular multi-view dataset characterized by a visual-feature view and a textual-visual view. The visual view is with 1750 dimensions, and the textual view is with 79 dimensions. BDGP includes 2,500 images of Drosophila embryos divided into 5 classes. Different from the construction strategy for MNIST-USPS, multi-view datasets, like BDGP, Caltech, and COIL-20, are built by concatenating multiple feature extractors or multi-modal measurements.

- Caltech [4] is an RBG image dataset constructed with 5 different visual descriptors, i.e., 40-dim wavelet moments (WM) feature, 254-dim CENTRIST feature, 1,984-dim HOG feature, 512-dim GIST feature, and 928-dim LBP feature. Each view of Caltech contains 1400 images and 7 classes.

- COIL-20 [18], consisting of 480 grayscale images in 128×128 pixel size of 20 categories, is described by 3 views. Different views represent different poses of the same object.

- BBCSport [11] is a text dataset in 5 topic areas. It consists of 544 documents collected from the BBC Sport website of sports news articles, related to 2 different viewpoints. The first view is with 3183 dimensions, and the second view is with 3203 dimensions.

- Multi-Fashion [20] is also a 28×28-dimensional grayscale image dataset on 10 different kinds of 10,000 fashionable products. Different views of the same item are represented by the different products from the same categories.

- In terms of NR-MNIST, which is also a variant of the handwritten digital image dataset MNIST, we also follow [20] to regard the noisy-processed MNIST and the rotated-processed MNIST as two different views. We use 60,000 image pairs for the general MVHC experiments in Section 4.2, and use the rest 10,000 image pairs for the inductive HC experiments in Section 4.5.

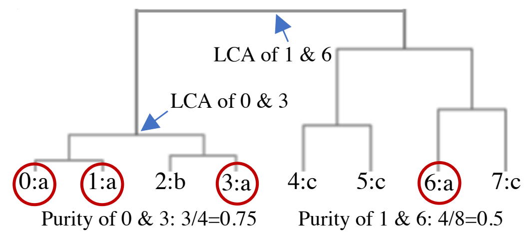

Figure 1: Illustration of DP

A.6. Dendrogram Purity Measurement

To evaluate the quality of the final hierarchical clustering tree, we follow [12, 8, 10] to adopt Dendrogram Purity (DP) as the metric for more complex hierarchical clustering.

Assume there is a ground truth flat clustering $\mathcal{C}^\star = \{\mathcal{C}_k^\star\}_{k=1}^K$ containing K clusters and denote any data point pairing (x_i, x_j) that is grouped into the same ground truth cluster as $\text{Pairs}^\star = \{(x_i, x_j) | \mathcal{C}^\star(x_i) = \mathcal{C}^\star(x_j)\}$. It is intuitive that with regard to the ground truth clustering, the comprehensive DP measurement for the HC tree T can be computed through the following steps, i.e., (1) traveling through all arbitrary data point pairs x_i, x_j belonging to the same ground truth cluster, i.e., $\mathcal{C}^\star(x_i) = \mathcal{C}^\star(x_j)$, (2) finding the descendant leaves of the LCA of the two nodes x_i, x_j in the tree, represented as $\text{subtree}(T[x_i \vee x_j])$, and (3) averaging the purity that any two leaves from the subtree also belongs to the same cluster $\text{pur}(\text{subtree}(T[x_i \vee x_j]), \mathcal{C}_k^\star)$. Thus, as shown in Figure 1 the DP measurement is formulated as

$$\text{DP}(T) = \frac{1}{|\text{Pairs}|} \sum_{k=1}^{K} \sum_{x_i, x_j \in \mathcal{C}_k^\star} \text{pur}(\text{subtree}(T[x_i \vee x_j]), \mathcal{C}_k^\star). \tag{5}$$

More intuitively, the tree structures with higher DP values are purer, which means the decoding dendrograms extract more similar hierarchies to the clusters of the ground truth flat partition.

A.7. Implementation Details

MHCN. The proposed MHCN model is implemented in the PyTorch platform. For efficient tree exploration and representation, the corresponding SciPy, networkx, and ETE (Environment for Tree Exploration) Python toolkits are adopted. All experiments are conducted on a Linux Server with an Intel Xeon E5-2630 v4 CPU, an NVIDIA TITAN Xp GPU, and 128GB RAM.

In MHCN, multiple common fully connected networks attached to the latent hyperbolic space with the

Dataset	Type	# Sample	# View	# Class
MNIST-USPS	Digits of Different Styles	5,000	2	10
BDGP	Image+Text	2,500	2	5
Caltech	WM+CENTRIST+LBP+GIST+HOG	1,400	5	7
COIL-20	Objects from Different Angles	480	3	20
BBCSport	Different Segments of the Same Document	544	2	5
Multi-Fashion	Clothes of Different Styles	10,000	3	10
NR-MNIST	Noisy MNIST+Rotated MNIST	70,000	2	10

Table 2: The description of multi-view datasets.

same architectures are adopted as the hyperbolic autoencoders (HAEs). To learn the latent hierarchical hyperbolic embeddings, the encoder of the HAE for each view is followed by the $\exp_{\mathbf{0}}$ mapping function, and the decoder is composed of an MLP pre-mapped by the $\log_{\mathbf{0}}$ mapping function. To be specific, the structure of every HAE can be represented as $\boldsymbol{X}^m - \mathrm{FC}_{500} - \mathrm{FC}_{500} - \exp_{\mathbf{0}}(\boldsymbol{Z}^m_{\tan}) - \boldsymbol{Z}^m_{\mathrm{hyp}} - \log_{\mathbf{0}}(\boldsymbol{Z}^m_{\tan}) - \mathrm{FC}_{500} - \mathrm{FC}_{500} - \boldsymbol{X}^m$, where FC_l is the fully connected layer including l neurons.

We use the minimal dataset-dependent hyper-parameter set for tuning. We set the latent dimension d, i.e., the dimensionality of the Poincaré ball, to 20 for all datasets. Since the parameters in hyperbolic space for our model can be considered as Euclidean parameters computed through $\boldsymbol{Z}^m_{\mathrm{hyp}} = \exp_{\mathbf{0}}(\boldsymbol{Z}^m_{\tan})$, where $\boldsymbol{Z}^m_{\tan} \in \mathbb{R}^d$, we directly train our model by using the common optimizer Adam with the learning rate set to $1e-3$ on MNIST-USPS, BDGP and Noisy-Rotated-MNIST, $5e-4$ on Caltech, COIL-20, and BBCSport, and $5e-3$ on Multi-Fashion. The trade-off coefficient α is set to 0.6. We empirically set $t = 1.0$ on all datasets, and set $\tau_{\mathrm{align}} = \tau_{\mathrm{uni}}$ to 1.0 on MNIST-USPS, BDGP, BBCSport, Multi-Fashion and NR-MNIST datasets, and 0.5 on Caltech and COIL-20 datasets. Moreover, we train the whole model for 20, 20, 200, 200, 150, 200, 50 epochs on MNIST-USPS, BDGP, Caltech, COIL-20, BBCSport, Multi-Fashion and NR-MNIST datasets, respectively. The batch size is set to 128 for all datasets. We run our model for 5 times and report the average performance in Section 4.2.

Baseline methods. In terms of baseline methods, the DP results of the baselines reported in Section 4.2 on all datasets except NR-MNSIT are excerpted from CMHHC [10]. More specifically, firstly, for the shallow discrete single-view hierarchical agglomerative clustering methods (HACs), like Single-linkage, Complete-linkage, Average-linkage, and Ward-linkage algorithms, we regard the concatenation of multiple views as a single-view pattern and directly apply the above single-view HACs by SciPy Python library, where we use the default distance metric by SciPy for HACs, i.e., the Euclidean distance. In addition, the Ward-linkage method tends to perform the best among the HACs, and Ward-linkage is correctly defined only if the Euclidean metric is adopted. Secondly, for the deep continuous single-view hierarchical clustering approaches (UFit and HypHC), and the existing multi-view hierarchical clustering methods (MHC and CMHHC), we follow the settings of CMHHC [10] for a fair comparison.

A.8. Experimental Results and Analysis

The DP comparison including the average values and the standard deviations (std) of 5 runs is presented in Table 4. As observed in Section 4.2, MHCN outperforms all baseline methods, especially the second-best CMHHC, on all datasets. In addition, the std results reported in Table 4 indicate the stable clustering performance given the mini-batch variance. These observations demonstrate the superiority of our MHCN against other methods, which is due to the one-stage pipeline to optimize the total objective designed to realize the characteristics of the high-quality multi-view hierarchical clustering trees.

A.9. Complexity Analysis

Let n represent the batch size. Generally, $N \gg M, n$. In the mini-batch optimization process, the complexities of computing the multi-view alignment loss, the reconstruction loss, and the hyperbolic uniformity loss are $O(M^2 n)$, $O(Mn)$, and $O(Mn^2)$, respectively. Additionally, the complexity of the bottom-up decoding strategy is $O(N^2)$ [1, 7, 3]. The whole complexity of MHCN can be calculated as $O(N/n(Mn + M^2 n + Mn^2) + N^2)$. Furthermore, the lowest complexity of baseline HACs is $O(N^2)$ of single linkage heuristic [16], which is equal to that of MHCN.

Therefore, combined with the DP results on all datasets in Section 4.2, and the total time spent on NR-MNIST dataset in Section 4.4, both the theoretical value and the experimental results demonstrate the scalability of our method for large-scale scenarios.

A.10. Training Time

OOM (out-of-memory) is encountered with NR-MNIST on our server, so we provide the runtime on other datasets as a reference. Table 3 shows the average runtime of 5 runs for

Method	MNIST-USPS	BDGP	Caltech	COIL-20	BBCSport	Multi-Fashion
UFit	23.03s	17.16s	12.69s	9.13s	10.04s	143.87s
HyperHC	5796.12s	1586.22s	191.09s	199.24s	250.32s	15969.39s
MHC	51.46s	38.42s	43.13s	234.42s	301.03s	101.55s
CMHHC	6153.49s	1964.75s	716.78s	376.45s	685.43s	20932.18s
MHCN	38.36s	22.79s	146.75s	50.90s	142.72s	499.98s

Table 3: The average time spent for "*OOM*" methods on datasets except NR-MNIST.

Method	MNIST-USPS	BDGP	Caltech	COIL-20	BBCSport	Multi-Fashion	NR-MNIST
HAC-Single	29.81%	61.88%	23.67%	72.56%	27.66%	27.89%	25.77%
HAC-Complete	54.36%	56.57%	30.19%	69.95%	34.78%	48.72%	27.71%
HAC-Average	69.67%	45.91%	30.90%	73.14%	29.05%	65.70%	59.74%
HAC-Ward	80.38%	58.61%	35.69%	80.81%	62.65%	72.33%	76.91%
UFit	21.67%	69.20%	19.00%	55.41%	30.33%	25.94%	*OOM*
HyperHC	32.99%±1.69%	31.21%±5.33%	22.46%±0.46%	28.50%±1.69%	29.08%±1.75%	25.65%±1.69%	*OOM*
MHC	78.27%±0.01%	89.14%±0.01%	45.22%±0.03%	66.50%±0.30%	42.43%±0.02%	54.81%±0.01%	40.87%±0.80%
CMHHC	94.49%±0.26%	91.53%±2.52%	66.52%±4.12%	84.89%±2.97%	53.50%±2.79%	96.25%±2.15%	*OOM*
MHCN	**99.22%±0.11%**	**96.22%±0.49%**	**77.14%±1.94%**	**94.70%±0.63%**	**78.93%±2.08%**	**97.67%±0.34%**	**98.71%±0.27%**

Table 4: The DP comparison results (%). Since there are no mini-batch training procedures, the std values of HAC-Single, HAC-Complete, HAC-Average, HAC-Ward, and UFit are 0.00%.

UFit, HyperHC, MHC, CMHHC, and our method MHCN on other datasets. Our method is significantly faster than HypHC and CMHHC, and comparable with MHC in terms of time cost. Since MHCN involves the procedure of representation learning, the time cost may be influenced by the speed of model convergence, which is dataset-dependent.

References

[1] Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In *FOCS*, pages 136–150. IEEE, 2015.

[2] James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31(59-115):2, 1997.

[3] Ines Chami, Albert Gu, Vaggos Chatziafratis, and Christopher Ré. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. In *NeurIPS*, pages 15065–15076, 2020.

[4] Delbert Dueck and Brendan J Frey. Non-metric affinity propagation for unsupervised image categorization. In *ICCV*, pages 1–8, 2007.

[5] Sylvestre Gallot, Dominique Hulin, and Jacques Lafontaine. *Riemannian geometry*, volume 2. Springer, 1990.

[6] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *NeurIPS*, volume 31, 2018.

[7] CS Karthik and Pasin Manurangsi. On closest pair in euclidean metric: Monochromatic is as hard as bichromatic. *Combinatorica*, 40(4):539–573, 2020.

[8] Ari Kobren, Nicholas Monath, Akshay Krishnamurthy, and Andrew McCallum. A hierarchical algorithm for extreme clustering. In *KDD*, pages 255–264, 2017.

[9] Zhaoyang Li, Qianqian Wang, Zhiqiang Tao, Quanxue Gao, and Zhaohua Yang. Deep adversarial multi-view clustering network. In *IJCAI*, pages 2952–2958, 2019.

[10] Fangfei Lin, Bing Bai, Kun Bai, Yazhou Ren, Peng Zhao, and Zenglin Xu. Contrastive multi-view hyperbolic hierarchical clustering. In *IJCAI*, pages 3250–3256, 2022.

[11] Shirui Luo, Changqing Zhang, Wei Zhang, and Xiaochun Cao. Consistent and specific multi-view subspace clustering. In *AAAI*, pages 3730–3737, 2018.

[12] Nicholas Monath, Manzil Zaheer, Daniel Silva, Andrew McCallum, and Amr Ahmed. Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space. In *KDD*, pages 714–722, 2019.

[13] Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *arXiv preprint arXiv:2101.04562*, 2021.

[14] Xi Peng, Zhenyu Huang, Jiancheng Lv, Hongyuan Zhu, and Joey Tianyi Zhou. Comic: Multi-view clustering without parameter selection. In *ICML*, pages 5092–5101, 2019.

[15] Peter Petersen. *Riemannian geometry*, volume 171. Springer, 2006.

[16] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

[17] Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. *arXiv preprint arXiv:2006.08210*, 2020.

[18] Daniel J Trosten, Sigurd Lokse, Robert Jenssen, and Michael Kampffmeyer. Reconsidering representation alignment for multi-view clustering. In *CVPR*, pages 1255–1265, 2021.

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

[20] Jie Xu, Yazhou Ren, Guofeng Li, Lili Pan, Ce Zhu, and Zenglin Xu. Deep embedded multi-view clustering with collaborative training. *Information Sciences*, pages 279–290, 2021.

[21] Jie Xu, Huayi Tang, Yazhou Ren, Liang Peng, Xiaofeng Zhu, and Lifang He. Multi-level feature learning for contrastive multi-view clustering. In *CVPR*, pages 16051–16060, 2022.