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

Abstract

Multi-view hierarchical clustering (MCHC) plays a pivotal role in comprehending the structures within multi-view data, which hinges on the skillful interaction between hierarchical feature learning and comprehensive representation learning across multiple views. However, existing methods often overlook this interplay due to the simple heuristic agglomerative strategies or the decoupling of multi-view representation learning and hierarchical modeling, thus leading to insufficient representation learning. To address these issues, this paper proposes a novel Multi-view Hierarchical Clustering Network (MHCN) model by performing simultaneous multi-view learning and hierarchy modeling. Specifically, to uncover efficient tree-like structures among all views, we derive multiple hyperbolic autoencoders with latent space mapped onto the Poincaré ball. Then, the corresponding hyperbolic embeddings are further regularized to achieve the multi-view representation learning principles for both view-common and view-private information, and to ensure hyperbolic uniformity with a well-balanced hierarchy for better interpretability. Extensive experiments on real-world and synthetic multi-view datasets have demonstrated that our method can achieve state-of-the-art hierarchical clustering performance, and empower the clustering results with good interpretability.

1. Introduction

Clustering is one of the fundamental research topics in data analysis [28, 44, 51]. With the advances in data acquisition, lots of real-world data could be presented by multiple views, e.g., different visual descriptors like GIST [48] + Histogram of Oriented Gradients (HOG) [13] + other deep features [53] for an image. Multiple views contain both congruent and incongruent information, which can be leveraged to explore potential consistency and complementarity across views and provide more comprehensive latent representations to facilitate downstream tasks, especially for unsupervised learning. This leads to the prevalence of multi-view clustering including multi-view partitional clustering [29, 35, 39, 49, 59] and multi-view hierarchical clustering [37, 68]. Aiming for high interpretability, multi-view hierarchical clustering is particularly appealing in the analysis of multi-view data at various levels of gran-

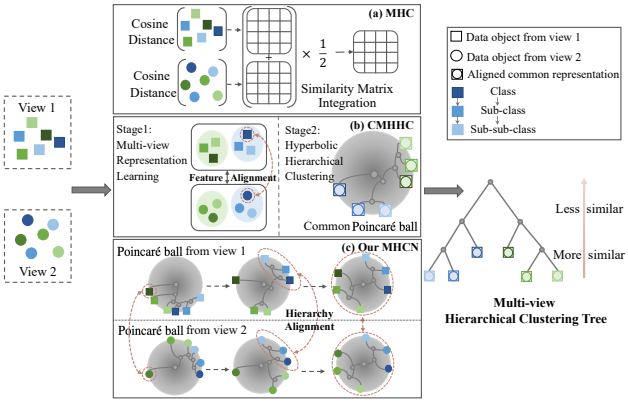


Figure 1: Comparison between prior MVHC methods, i.e., MHC and CMHHC, versus our proposed MHCN. **(a) MHC** calculates the average cosine distance matrix of all views and agglomerates the nearest neighbors according to the simply-aligned similarity matrix. **(b) CMHHC** performs multi-view alignment and similarity learning in the Euclidean space, and hierarchical clustering in hyperbolic space, separately. **(c) Our MHCN**, a one-stage pipeline, directly aligns the latent hierarchies on different Poincaré balls, leading to more effective MVHC trees.

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

ularity. For instance, gene expression analysis [36] integrates multi-omics data and clusters cells into an organization/organ/system hierarchy. Besides, species trees [17], gathering animals by their physical appearances, sounds, or other features, reflect the hierarchical organization in phylogenetic inference.

Despite these merits, there are still significant limitations in existing multi-view hierarchical clustering methods. As shown in Figure 1, existing models for solving the MVHC problem include the discrete linkage-based method Multi-view Hierarchical Clustering (MHC) [68], and the continuous deep neural network-based Contrastive Multi-view Hyperbolic Hierarchical Clustering (CMHHC) [37]. As a seminal work for MVHC, shallow MHC partitions multi-view data at multiple levels through the cosine distance integration and the nearest neighbor agglomeration components. However, without applying deep representation learning, this method simply considers the mean latent feature vectors as the common representations of all views. Thus, it cannot fully capture the consistency and complementarity among different views, leading to degenerated HC performance, especially on complicated real-world datasets. The two-stage deep model CMHHC integrates a multi-view alignment learning module, an aligned feature similarity learning module, and a continuous hyperbolic hierarchical clustering module. However, CMHHC takes into account two independent geometry settings, treating multi-view learning in the Euclidean space and HC in hyperbolic space, enlarging the gap between the two processes. Thus, the performance may still be suboptimal.

To address these issues, this paper proposes the Multi-view Hierarchical Clustering Network (MHCN), as sketched in Figure 2. Taking the desired MVHC tree structures into consideration, we make corresponding designs for MHCN as follows. Firstly, drawing inspiration from the recent success of latent hyperbolic anatomy [42, 52], which excels in modeling hierarchies with minimal distortion, we incorporate multiple hyperbolic autoencoders to capture the inherent hierarchies present in different views. Secondly, considering multi-view consistent and complementary principles [34, 66], we design the multi-view alignment loss between two arbitrary views to discover the view-common information, and use the reconstruction loss to retain the view-private information, respectively. Thirdly, uniformly-distributed hyperbolic embeddings on the Poincaré ball create well-balanced trees, leading to more interpretable semantics compared with skewed ones. In order to improve interpretability in clustering trees, we encourage the balanced hierarchy construction via a hyperbolic uniformity loss, which facilitates learning such hyperbolic embeddings. In this way, MHCN provides explicit guidance for ensuring the quality of the tree structure and integrates multi-view representation learning and hierarchy discovery processes on hyperbolic manifolds with hierarchy-friendly hyperbolic autoencoders. As the pioneering one-stage solution for the MVHC problem, the MHCN model collectively optimizes the three objectives with mini-batch training. Thus, large datasets can be efficiently processed. After obtaining the optimal hyperbolic hierarchical representations for unseen data with trained neural networks, MHCN can decode back to the underlying tree structures, making inductive hierarchical clustering possible.

Different from existing MVHC models, our contributions are summarized as follows.

- We analyze the multi-view hierarchical clustering problem and propose a novel hyperbolic neural network-based model to tackle the problem, where the latent space is grafted on the Poincaré ball of hyperbolic space for explicit hierarchical structures.
- We introduce three objectives to facilitate the final multi-view hierarchical clustering trees with desired properties, i.e., a multi-view alignment loss, a reconstruction loss, and a hyperbolic uniformity loss.
- We conduct extensive experiments to show the superiority of MHCN compared with other multi-view HC methods in terms of both effectiveness and scalability.

2. Related Work

2.1. Hierarchical Clustering

Bottom-up linkage algorithms for solving HC are simple and easy to implement, which recursively merge similar data points to build larger cluster sub-trees until a complete dendrogram appears [10, 31]. Typical agglomerative heuristics include Single Linkage, Complete Linkage, Average Linkage, and Ward Linkage [10]. While these traditional algorithms are widely used in practice, there is a limit to fit continuous optimization due to their inherent discreteness [14, 43, 61]. Hence, there have been more attempts at gradient-based HC through embedding methods recently [5, 9, 24, 46, 56]. For example, UFit [9] proposed a gradient-based fitting framework over ultrametric. Besides, similarity-based HypHC [5] directly relaxed Dasgupta's cost with the aid of the continuous form of hyperbolic leaf nodes and lowest common ancestors (LCAs).

In contrast to the above HC methods, our unified framework is multi-view data-oriented, which can be optimized continuously via specifically-designed hierarchical clustering objectives. Also, little previous research pays close attention to the balancing property of clustering trees, which is one of the mainly-addressed spots in our method.

2.2. Multi-view Clustering

Existing multi-view clustering (MVC) methods can be roughly classified into four categories, namely, canon-

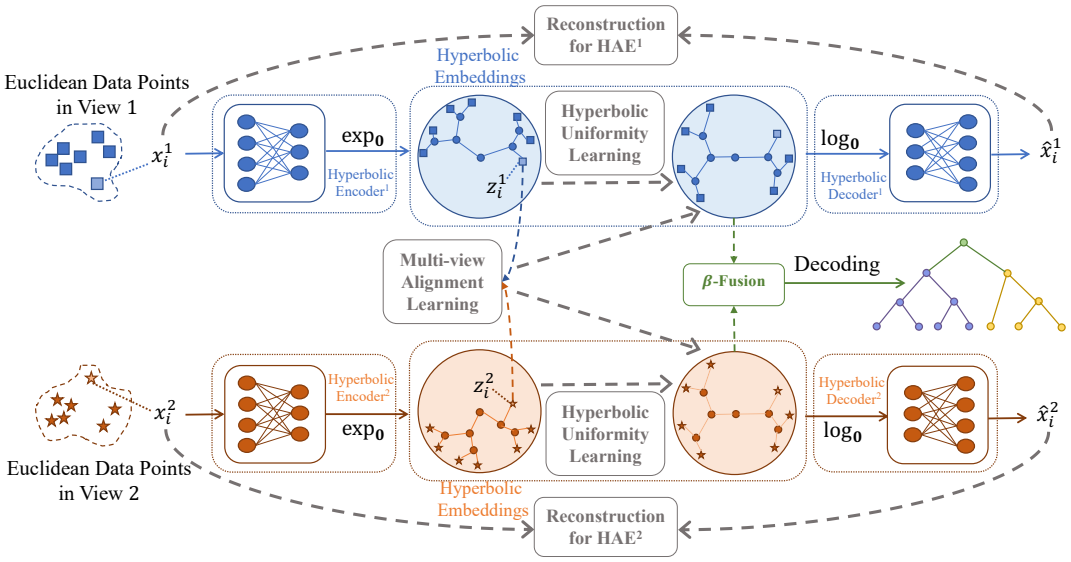


Figure 2: Overview of the proposed MHCN (taking two views as an example). MHCN assigns a hyperbolic autoencoder to each view, with the latent space grafted on the Poincaré ball for explicit hierarchies, where the exponential mapping function $\exp_{\mathbf{0}}$ follows the hyperbolic encoder, and the hyperbolic decoder connects behind the logarithm mapping function $\log_{\mathbf{0}}$. The whole model is trained with the three joint learning objectives: (1) the multi-view alignment learning aims to align the hyperbolic embeddings z_i^1 and z_i^2, (2) the reconstruction for HAE is used to project the complementary information in each view into the latent space, and (3) the hyperbolic uniformity learning guarantees z_i^1 and z_i^2 to distribute uniformly on the corresponding Poincaré balls, respectively. After β-fusion, the final MVHC tree can be decoded from the common hyperbolic representations.

ical correlation analysis-based [8], matrix factorization-based [3], subspace-based [27, 33] and graph-based [30] methods. Recently, deep MVC methods have been developed increasingly [35, 38, 49, 53, 59, 67]. However, most methods are limited to partitional clustering. The high computational complexities of these methods hinder the practicability and development of multi-view hierarchical clustering. To this end, research about MVHC has gradually emerged. Shallow MHC [68] roughly averages the cosine distance graphs for common latent features, which cannot achieve consistency of all views. A recent two-stage deep method CMHHC [37] presents an MVHC tree over aligned common representations through three modules, i.e., multi-view alignment learning, unsupervised metric learning, and hyperbolic hierarchical clustering.

Unlike CMHHC, which regards multi-view representation learning and hierarchical clustering as two separate stages, our method unifies them into a one-stage training pipeline, where they can reinforce each other for naturally contributing to the final HC trees.

2.3. Hyperbolic Models

There is extensive literature on hierarchical models conducted in the Euclidean space and its applications [22, 25, 58]. However, suffering from the inherent discrete nature of trees, the Euclidean hierarchical models have difficulty in constructing the hierarchical structure of data explicitly, while hyperbolic models can. The underlying reason is that the exponential growth of the volume in hyperbolic space with its radius is analogous to the exponential growth of the number of leaves in a tree with its altitude, which is infeasible in the Euclidean space [43]. Thus, there has been an increasing interest in the generalization of hyperbolic neural networks to learn hyperbolic representations [6, 20, 26, 42, 50, 55].

3. Method

In this part, we propose a hyperbolic neural network model for MVHC, termed as Multi-view Hierarchical Clustering Network (MHCN), as shown in Figure 2. For clarity, we first briefly review the background of hyperbolic geometry and a specific hyperbolic model, i.e., the Poincaré ball. Then, we describe the architecture of HAE, define what desired trees look like through specific loss functions, and introduce the decoding strategy. Lastly, we summarize the whole optimization process.

Let $\{\boldsymbol{X}^m \in \mathbb{R}^{N\times D_m}\}_{m=1}^M$ be a multi-view dataset including N samples across M views, where each $\boldsymbol{x}_i^m \in \mathbb{R}^{D_m}$ denotes the i-th instance of D_m dimension from the m-th view. Our target is to present a unified method for

clustering tree learning by searching over multi-view continuous hierarchical representations in hyperbolic space.

3.1. Preliminaries

We first introduce the Riemannian geometry and the Poincaré ball model of hyperbolic geometry, with which we build the hyperbolic autoencoders for our model. A more detailed review can be found in the appendix.

3.1.1 Riemannian Geometry

An n-dimensional *manifold* $\mathcal{M}$ [19, 54] is a real and smooth space, which can be locally approximated to a linear n-dimensional space $\mathbb{R}^n$ at each point $\boldsymbol{x} \in \mathcal{M}$. For any point $\boldsymbol{x}$ of the manifold $\mathcal{M}$, the corresponding local *tangent space* $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$, is the first order linear approximation of $\mathcal{M}$ around the point $\boldsymbol{x}$. The related *metric tensor* $g_x : \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \times \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \to \mathbb{R}$ defines an inner product on $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$, and a *Riemannian metric* $g = (g_x)_{x\in\mathcal{M}}$ is a set of such inner products on $\mathcal{M}$. In this way, a *Riemannian manifold* can be presented as a matching tuple $(\mathcal{M}, g)$. With g_x, the local geometric attributions of $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ are accessible, and then the global distances can be achieved by integrating the local properties together. To project a tangent vector $\boldsymbol{z}$ 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, and the inverse form is *logarithmic map* $\log_{\boldsymbol{x}} : \mathcal{M} \to \mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ [20, 52, 55]. More detailed review is in the appendix.

3.1.2 Poincaré Ball Model of Hyperbolic Geometry

Many studies [12, 21, 47] have demonstrated that data representations in most machine learning applications lie on a Riemannian manifold [32]. Later on, how to generalize neural networks to a Riemannian space has made remarkable progress [15, 42, 45]. An n-dimensional hyperbolic space $\mathbb{H}^n$ is an n-dimensinal Riemannian manifold with constant negative curvature [2, 4]. We choose to perform our model on the n-dimensional Poincaré ball $(\mathbb{B}^n, g^{\mathbb{B}})$ with a constant negative curvature -1, 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.

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 *Möbius Addition* $\oplus$ [60], the closed-form expressions of $\exp_{\boldsymbol{z}}$ and $\log_{\boldsymbol{z}}$ are respectively given by

$$\begin{aligned} \exp_{\boldsymbol{z}}(\boldsymbol{t}) &= \boldsymbol{z} \oplus (\tanh(\frac{\lambda_{\boldsymbol{z}}||\boldsymbol{t}||}{2})\frac{\boldsymbol{t}}{||\boldsymbol{t}||}), \\ \log_{\boldsymbol{z}}(\boldsymbol{z'}) &= \frac{2}{\lambda_{\boldsymbol{z}}}\text{arctanh}(||-\boldsymbol{z} \oplus \boldsymbol{z'}||)\frac{-\boldsymbol{z} \oplus \boldsymbol{z'}}{||-\boldsymbol{z} \oplus \boldsymbol{z'}||}. \end{aligned} \tag{1}$$

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

$$\begin{aligned} \exp_{\boldsymbol{0}}(\boldsymbol{t}) &= \tanh(||\boldsymbol{t}||)\frac{\boldsymbol{t}}{||\boldsymbol{t}||}), \\ \log_{\boldsymbol{0}}(\boldsymbol{z'}) &= \text{arctanh}(||\boldsymbol{z'}||)\frac{\boldsymbol{z'}}{||\boldsymbol{z'}||}. \end{aligned} \tag{2}$$

By means of the above main operations $\exp_{\boldsymbol{0}}(\boldsymbol{t})$ and $\log_{\boldsymbol{0}}(\boldsymbol{z'})$, MHCN is able to perform the basic transformations of the latent representations between the Euclidean space and the hyperbolic space.

3.2. Hyperbolic Autoencoders

The conventional approach to learning useful representations is through a regular Euclidean autoencoder [18, 23, 64]. However, there has been significant interest in hierarchical representation learning using non-Euclidean geometry, specifically the hyperbolic space. This attention arises from the fact that the number of leaf nodes in tree structures increases exponentially with depth, mirroring the exponential growth of the hyperbolic surface area with its radius, while the growth in the Euclidean space is polynomial [6, 20, 52, 55]. As a result, we augment the standard autoencoder with hyperbolic geometry in the embedding space, and propose a novel multi-view hyperbolic model consisting of multiple hyperbolic autoencoders assigned for different views, shown in Figure 2. Each hyperbolic autoencoder's geometry-aware encoder and decoder are constructed via the introduced exponential and logarithmic map functions in Eq.(2). We show the detailed structures of the encoders and the decoders in the following.

3.2.1 Encoder with the Exponential Map

With regard to the m-th view $\boldsymbol{X}^m$, the encoder of m-th HAE, i.e., $f_{\text{enc}}(\boldsymbol{X}^m; \theta^m_{\text{enc}}) : \boldsymbol{X}^m \in \mathbb{R}^{D_m} \to \boldsymbol{Z}^m_{\text{hyp}} \in \mathbb{B}^d$, is designed as a regular Euclidean encoder followed by the exponential map $\exp_{\boldsymbol{0}}(\cdot)$, along with which the output of the hyperbolic encoder is projected onto the Poincare ball. $\boldsymbol{Z}^m_{\text{hyp}}$ represents the d-dimensional hyperbolic latent code with explicit hierarchical patterns and θ^m_{enc} denotes the encoder parameters.

3.2.2 Decoder with the Logarithm Map

Similar to the hyperbolic encoder, the decoder of the m-th HAE, $f_{\text{dec}}(\boldsymbol{X}^m; \theta^m_{\text{dec}}) : \boldsymbol{Z}^m_{\text{hyp}} \in \mathbb{B}^d \to \boldsymbol{X}^m \in \mathbb{R}^{D_m}$

for the m-th view $\boldsymbol{X}^m$ which is parameterized by θ^m_{dec}, is constituted of a regular Euclidean decoder preprocessed by the logarithm map $\log_{\mathbf{0}}(\cdot)$, projecting the latent hyperbolic features back to the Euclidean space.

3.3. Loss Functions for Desired HC Trees

High-quality MVHC trees should: 1) preserve the intrinsic hierarchy of each view via learned hyperbolic embeddings, 2) make full use of multi-view complementary and consistent information to acquire common hierarchical representations, and 3) not increase hierarchies without strong necessity [1, 7, 14, 62]. In this section, based on the structure of multi-view HAEs, we design the total loss functions for our model to realize the above characteristics:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{mvl}} + \mathcal{L}_{\text{uni}}\,. \tag{3}$$

In our model, we jointly achieve the multi-view consistency and complementarity in the hyperbolic common-view space by optimizing the combined loss $\mathcal{L}_{\text{mvl}}$ of the multi-view alignment loss $\mathcal{L}_{\text{align}}$ and the reconstruction loss $\mathcal{L}_{\text{rc}}$:

$$\mathcal{L}_{\text{mvl}} = \alpha\mathcal{L}_{\text{align}} + (1-\alpha)\mathcal{L}_{\text{rc}}\,, \tag{4}$$

where α is a trade-off coefficient to balance the effects of consistency and complementarity for multi-view learning.

Learning view-common consistency with the multi-view alignment loss. The extracted common representations $\{\boldsymbol{Z}^m_{\text{hyp}} = f_{\text{enc}}(\boldsymbol{X}^m)\}_{m=1}^M$ are expected not only to uncover the complementary relationships among different views but also to contain sufficient consensus information of all views. To achieve consistency across multiple views, we apply distance and similarity-based alignment by conducting a multi-view alignment loss endowed on the hyperbolic embeddings. Here, we consider the cosine distance between two latent features $\boldsymbol{z}_i^{m_1}$ and $\boldsymbol{z}_j^{m_2}$ as the similarity measurement for multi-view alignment:

$$\text{sim}(\boldsymbol{z}_i^{m_1}, \boldsymbol{z}_i^{m_2}) = \frac{\langle \boldsymbol{z}_i^{m_1}, \boldsymbol{z}_j^{m_2}\rangle}{||\boldsymbol{z}_i^{m_1}||||\boldsymbol{z}_j^{m_2}||}\,. \tag{5}$$

Supported by the conformality property of the Riemannian geometry, i.e., the Poincaré ball preserves the same angels as the Euclidean space, the computation of the cosine similarity in the Euclidean space is equal to that on the Poincaré ball [20, 55]. Besides, unlike the geodesic distances, values of the cosine distances are kept in a certain range, which is friendly to optimization processes.

Then, the multi-view alignment loss between the hyperbolic embeddings of the same instance from any two views $\boldsymbol{Z}^{m_1}_{\text{hyp}}, \boldsymbol{Z}^{m_2}_{\text{hyp}}$ is given by:

$$\mathcal{L}^{(m_1,m_2)}_{\text{align}} = -\frac{1}{N}\sum_{i=1}^{N}\text{sim}(\boldsymbol{z}_i^{m_1}, \boldsymbol{z}_i^{m_2})/\tau_{\text{align}}\,, \tag{6}$$

where τ_{align} is used as a temperature parameter for the scaled distance function. Here, we notice that CMHHC [37] also serves to learn common representations. However, CMHHC aligns Euclidean features, while MHCN directly makes the corresponding hyperbolic embeddings closer.

Therefore, the complete multi-view alignment loss across all view pairs is formulated as:

$$\mathcal{L}_{\text{align}} = \sum_{m_1=1}^{M}\sum_{m_2=1, m_1\neq m_2}^{M}\mathcal{L}^{(m_1,m_2)}_{\text{align}}\,. \tag{7}$$

Learning view-private complementarity with the reconstruction loss of HAEs. In multi-view representation learning models, autoencoders are widely used for their impressive strengths, such as preserving the useful view-private information and preventing data points from collapsing to a small number of clusters [29, 67]. In our multi-view hierarchical clustering model, the learned representations by multiple HAEs consider the preservation of the local hierarchical structure within each view. This implies that similar instances from each view remain close to each other in the hyperbolic space, while the latent codes obtained from dissimilar instances remain distant from one another. To this end, the total reconstruction loss function of HAEs for multiple views is formulated as

$$\mathcal{L}_{\text{rc}} = \sum_{m=1}^{M}\mathcal{L}^m_{\text{rc}} = \sum_{m=1}^{M}\frac{1}{N}\sum_{i=1}^{N}||\boldsymbol{x}_i^m - f_{\text{dec}}(f_{\text{enc}}(\boldsymbol{x}_i^m))||_2^2\,, \tag{8}$$

where $\mathcal{L}^m_{\text{rc}}$ is the corresponding m-th view objective to reconstruct $\boldsymbol{X}^m$.

Learning tree balancing property with the hyperbolic uniformity loss. Through the multi-view alignment and reconstruction related to the Poincaré ball, our model obtains reliable multi-view representations and discovers the hierarchical structure of each view simultaneously. Last but not least, a hierarchical clustering tree with explicit balancing property provides fewer hierarchies and higher interpretability than skewed ones.

Consequently, we leverage the uniformity of the hyperbolic embedding distribution located on the Poincaré ball to learn separable features and enhance the contribution of the balanced affinity. Concretely, considering the form of Gaussian potential kernel [11, 40], we define a hyperbolic uniformity objective based on the cosine distance as:

$$\mathcal{L}_{\text{uni}} = \sum_{m=1}^{M}\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\log(e^{-t(-\text{sim}(\boldsymbol{z}_i^m, \boldsymbol{z}_j^m)/\tau_{\text{uni}})})\,. \tag{9}$$

Still, the temperature parameter τ_{uni} is a scaling factor to the cosine distance. Intuitively, the hyperbolic uniformity loss encourages the latent hyperbolic embeddings to push farther away from each other and to benefit the uniform global distribution of embeddings on the Poincaré ball.

3.4. From Hyperbolic Embeddings to Binary Trees

We first adopt β-fusion [55] to concatenate the optimal hyperbolic embeddings from multiple views into the common-view space, to guarantee the concatenated norms maintain upper-bounded by the radius of the Poincaré ball. To output an intuitive HC tree from the common continuous hyperbolic representations on the Poincaré ball back to discrete HC trees with low error, we apply the decoding strategy proposed by [5], which realizes the underlying correspondence between the hyperbolic embeddings and the HC trees with low-error. This bottom-up decoding algorithm decodes the common hyperbolic embeddings by measuring the similarity of any two embeddings with the distance from their lowest common ancient (LCA) to the origin of the ball and merging more similar ones iteratively.

3.5. Optimization Process

The whole optimization process of MHCN is summarized in the appendix. We train this model by adopting a one-stage pipeline of mini-batch gradient descent, making our method feasible to scale up to large-scale scenarios. Concretely, the integrated framework of multiple HAEs is optimized collectively with the $\mathcal{L}_{\text{total}}$ in Eq.(3).

4. Experiments

In the following, we conduct experiments to verify the effectiveness and efficiency of the proposed method and analyze the results.

4.1. Experimental Settings

Datasets. Following literature [37, 53, 57, 59, 63, 67], we adopt seven common multi-view datasets in our experiments, including five regular-scale datasets (MNIST-USPS, BDGP, Caltech, COIL-20 and BBCSport) and two large-scale datasets (Multi-Fashion and NR-MNIST). MNIST-USPS [53] is a handwritten digital dataset with 5,000 images from 10 categories, where the digits from MNIST and USPS are two views. BDGP [35] includes 2,500 images of Drosophila embryos divided into 5 classes, characterized by visual and textual features. Caltech [16] containing 1,400 RGB images of 7 classes is constructed with 5 different visual descriptors. COIL-20 [59] contains 480 grayscale images of 20 classes, described by 3 different angles. BBCSport [41] 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. Multi-Fashion [66] is also a grayscale image dataset on 10 kinds of 10,000 fashionable products. Three different views are represented by three products from the same category. In terms of NR-MNIST [65], we regard the noise-processed MNIST and the rotated-processed MNIST as two 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.

Dendrogram Purity measurement for HC. The tree structures with various fine-grained partitional clusters can be truncated at any level. Therefore, instead of using partitional clustering metrics, e.g., ACC and NMI, evaluating the performance of a clustering tree needs a much more comprehensive measurement. To this end, we adopt the Dendrogram Purity (DP) measurement to evaluate the quality of the clustering tree [31, 37, 43], which computes the average purity over the descendant leaves of LCA of all data point pairs belonging to the same ground-truth clusters. Trees with higher DP results tend to be more similar to the ground-truth clusters. Note that the Dasgupta's Cost, another HC metric applied in [5], requires an uncontroversial predefined similarity measurement as input, which is not available in our settings. The detailed reasons for using DP and not using ACC or NMI are given in the appendix.

Baseline methods. The baselines include three kinds of methods as follows: (1) shallow discrete single-view hierarchical agglomerative clustering methods (HACs), e.g., Single-linkage, Complete-linkage, Average-linkage, and Ward-linkage algorithms; (2) deep continuous single-view HC approaches, e.g., UFit and HypHC, where we concatenate the raw features of all views into a single-view pattern [53]; and (3) State-of-the-art MVHC methods, including MHC and CMHHC.

Implementation. The proposed network architecture is trained with the PyTorch platform. The fully connected networks with the same architecture are adopted to implement the HAEs for all views in our MHCN. We use the minimal dataset-dependent hyperparameter set for tuning. Since the parameters in hyperbolic space can be considered as Euclidean parameters computed through $\boldsymbol{Z}^m_{\text{hyp}} = \exp_{\mathbf{0}}(\boldsymbol{Z}^m_{\text{tan}})$, where $\boldsymbol{Z}^m_{\text{tan}} \in \mathbb{R}^d$, we directly train our model by using the common optimizer Adam. All experiments are conducted on a Linux Server with an Intel Xeon E5-2630 v4 CPU, an NVIDIA TITAN Xp GPU, and 128GB RAM. More implementation details are provided in the supplementary materials.

4.2. Experimental Results and Analysis

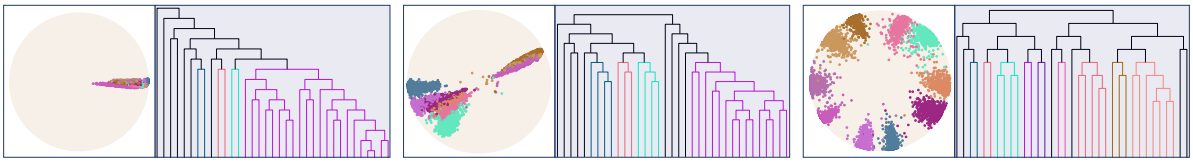
The comparison results are shown in Table 1. DP results of the baseline methods on all datasets except NR-MNSIT are directly taken from [37]. We can observe that: (1) MHCN outperforms the second-best baseline, i.e., CMHHC, on all datasets. Unlike two-stage CMHHC, our one-stage framework relieves the separation of multi-view representation learning and hierarchical clustering and naturally makes two associated processes boost each other. (2) Compared with the representative shallow MHC, our deep model also obtains considerable improvements on all datasets. MHC is not capable of learning rich multi-view

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%	31.21%	22.46%	28.50%	29.08%	25.65%	*OOM*
MHC	78.27%	89.14%	45.22%	66.50%	42.43%	54.81%	40.87%
CMHHC	94.49%	91.53%	66.52%	84.89%	53.50%	96.25%	*OOM*
MHCN	**99.22%**	**96.22%**	**77.14%**	**94.70%**	**78.93%**	**97.67%**	**98.71%**

Table 1: Comparison results. Higher DP values indicate better clustering performance. "*OOM*" is out-of-memory on our server.

Variants	MNIST-USPS	BDGP	Caltech	COIL-20	BBCSport	Multi-Fashion	NR-MNIST
MHCN (complete)	**99.22%**	**96.22%**	**77.14%**	**94.70%**	**78.93%**	**97.67%**	**98.71%**
w/o $\mathcal{L}_{\text{rc}}$	99.07%	95.46%*	75.17%†	93.85%	78.38%	96.78%*	98.20%
w/o $\mathcal{L}_{\text{align}}$	41.29%†	56.83%†	57.56%†	78.68%†	40.58%†	53.16%†	28.36%†
w/o $\mathcal{L}_{\text{uni}}$	19.42%†	44.31%†	36.69%†	51.57%†	28.52%†	40.37%†	32.64%†
Euclidean AEs	92.72%†	52.47%†	32.50%†	70.69%†	56.58%†	60.55%†	53.30%†

Table 2: Ablation study of our method. "*" and "†" indicate that the difference is significant at 0.05 and 0.01, respectively.



(a) Without $\mathcal{L}_{\text{uni}}$ (b) With 0.1 * $\mathcal{L}_{\text{uni}}$ (c) With $\mathcal{L}_{\text{uni}}$

Figure 3: Illustrations of the effectiveness of hyperbolic uniformity learning. We visualize the hyperbolic embeddings and the top-35 non-leaf internal nodes of corresponding decoding dendrograms. Left: Distributions of the learned hyperbolic embeddings optimized by different weights of $\mathcal{L}_{\text{uni}}$, i.e., 0.0, 0.1, and 1.0. Right: Top-35 non-leaf internal nodes of corresponding decoding dendrograms. With the weight of $\mathcal{L}_{\text{uni}}$ increasing, the distribution becomes uniform and the dendrograms are gradually balanced.

hierarchical representations as a result of poor hierarchical representation ability and ignorance of the balancing property of the tree structure. (3) Our method performs significantly better than all multi-view concatenation followed by single-view discrete or continuous HC methods. In contrast to concatenating features of all views simply, through learning the multi-view alignment and preserving each view's local relationships, MHCN encodes the meaningful common hierarchical representations across multiple views into the latent hyperbolic space. (4) The performance of MHCN on large-scale NR-MNIST is much better than other comparison methods. Meanwhile, conducting deep methods, like UFit, HyperHC, and CMHHC, on oversized NR-MNSIT results in high time complexity and unbearable memory cost. The scalability of MHCN can be attributed to the one-stage pipeline to optimize the total loss by batch-based training.

4.3. Ablation Studies

To further understand the effectiveness of all the factors of MHCN, we conduct ablation studies and report the results in Table 2. We mainly consider the following ablations, i.e., (a) MHCN without $\mathcal{L}_{\text{rc}}$, (b) MHCN without $\mathcal{L}_{\text{align}}$, (c) MHCN without $\mathcal{L}_{\text{uni}}$, and (d) MHCN with Euclidean autoencoders. We also report the significance of test results for the differences compared with MHCN.

As shown in Table 2, $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uni}}$ both contribute

Running Time	MHCN	HAC-Single	HAC-Complete	HAC-Average	HAC-Ward
NR-MNIST	900.92s	1823.43s	1903.12s	2055.16s	2204.11s

Table 3: Total time spent of MHCN and HACs on NR-MNIST dataset.

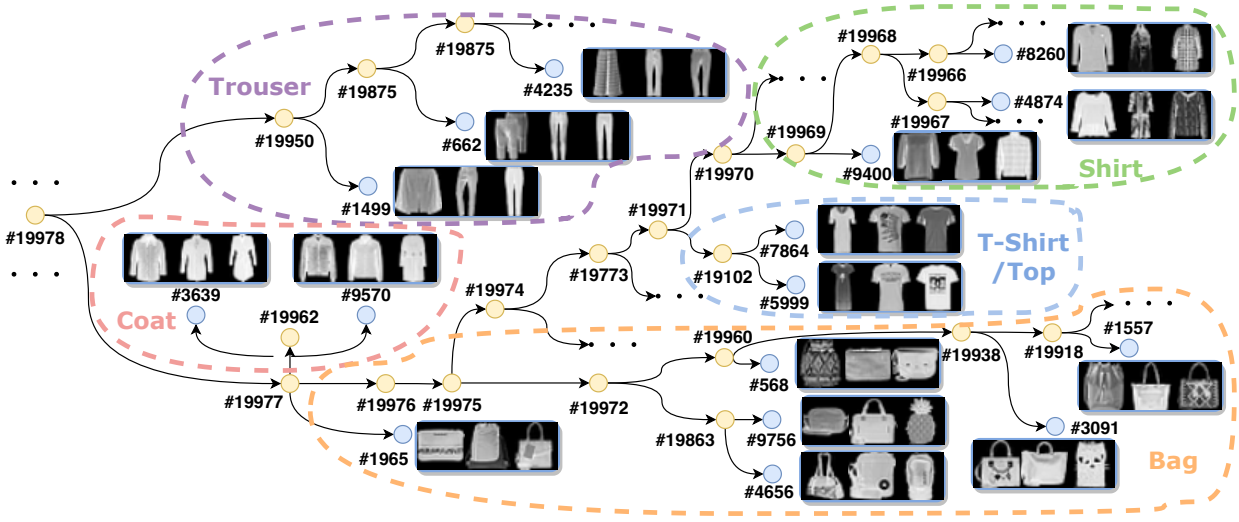


Figure 4: The sampled sub-tree of LCA #19978 from the Multi-Fashion HC tree. We can observe that the sub-trees containing more similar kinds of products are closer, e.g., categories Shirt in the green circle and T-Shirt/Top in the blue circle, while less similar products are merged together at higher LCA, e.g., categories Shirt in the green circle and Trouser in the purple circle, or categories Bag in the orange circle and Trouser in the purple circle.

a lot to the final performance. The effectiveness of $\mathcal{L}_{\text{rc}}$ is relatively less significant. This may be due to the property of experimental datasets. Especially, such datasets e.g., MNIST-USPS, Multi-Fashion, and NR-MNSIT are constructed by picking pairs of individual objects from the corresponding classes of multiple datasets [53, 65, 67]. So consistency may play a much more important role compared with complementarity. As for MHCN with Euclidean AEs, we can find that the performance also degenerates significantly, indicating the effectiveness of hyperbolic embeddings for modeling hierarchies.

In addition, to intuitively show the effectiveness of $\mathcal{L}_{\text{uni}}$, taking MNIST-USPS as an example, we train an MHCN model mapping the data to a two-dimensional Poincaré ball with different weights for $\mathcal{L}_{\text{uni}}$. We visualize the averaged embeddings and the decoded clustering trees. For clarity, we present the top-35 non-leaf nodes in Figure 3. As shown, if $\mathcal{L}_{\text{uni}}$ is absent, only $\mathcal{L}_{\text{align}}$ for multi-view consistency and $\mathcal{L}_{\text{rc}}$ for multi-view complementarity will force the embeddings to gather in a small corner of the Poincaré ball, resulting in a skewed tree that lacks interpretability. With the weight of $\mathcal{L}_{\text{uni}}$ growing, the embeddings become more uniformly distributed, and the corresponding clustering trees become more balanced.

4.4. Scalability Analysis

As shown in Table 1, the DP results of our method exceed those of other baselines on the large-scale NR-MNIST dataset. Also, Table 3 shows that the running time of our method achieves a significant decline when compared with the classic baselines, i.e., HACs. The theoretical complexity analysis can be found in the appendix.

4.5. Inductive Hierarchical Clustering Experiments

It is difficult for traditional HAC-based methods to train on one dataset and then predict on another. However, this inductive HC setting becomes relatively easy with MHCN. We train the model with the training set of NR-MNIST, then infer the results with the test set, and finally, the DP result is 96.54%. This result is numerically comparable with the result in Table 1 (98.71%). The ability of inductive HC may help improve the performance, e.g., transfer learning.

4.6. Parameter Sensitivity Analysis

Trade-off efficient α. To start with, we focus on the choice of the trade-off efficient α. By fixing the other hyperparameters, we investigate how the α parameter balances the effects of reconstruction and alignment loss components for multi-view learning. We show the DP result of different

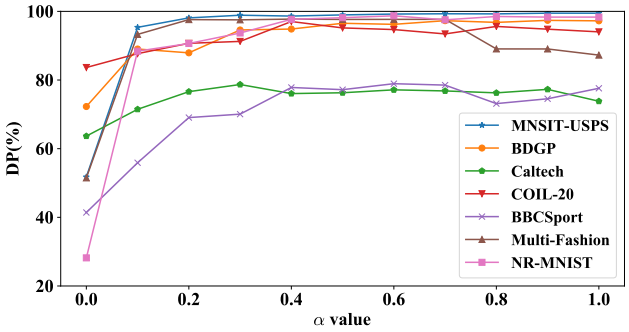


Figure 5: α sensitivity analysis

α values in a range between 0.0 and 1.0. As shown in Figure 5, with the increase of α values, the quality of the final tree structures consistently stabilizes at an optimal level on all datasets. When $\alpha = 0$, the performance of HC gets near the lowest point, especially on MNIST-USPS, BBCSport, Multi-Fashion, and NR-MNSIT datasets. The principal reason is that in MHCN, the consistency across all views is achieved via the multi-view alignment loss, which is much essential to capture view-common meaningful semantics and further boost the final clustering trees.

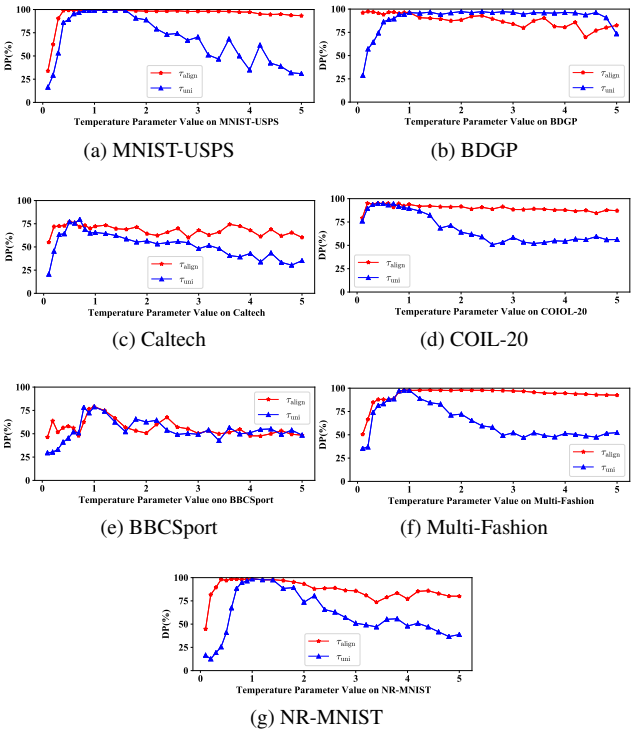
Temperature parameters τ_{align} and τ_{uni}. Similarly, with the other hyperparameters fixed, we present the DP results on all datasets by varying τ_{align} and τ_{uni} from 0.0 to 5.0. As shown in Figure 6, both τ_{align} and τ_{uni} set to around 1.0 achieve optimal performance on MNIST-USPS, BDGP, BBCSport, Multi-Fashion, and NR-MNIST datasets, while the HC performance on Caltech and COIL-20 datasets are more promising with $\tau_{\text{align}} = \tau_{\text{uni}} = 0.5$. Besides, the model is more robust to changes of τ_{align}.

4.7. Case Study

In addition to quantitative analysis, we qualitatively examine the quality of the final MVHC tree structures learned by our method. We take a truncated sub-tree of the complete Multi-Fashion tree as an example in Figure 4, where more similar leaf nodes are merged together earlier in the tree structure, guaranteeing explicit hierarchical partitioning. We can observe that more similar leaf nodes are merged together earlier in the tree structure, e.g., categories Shirt and T-Shirt/Top, while less similar products are merged together at higher LCA, e.g., categories Shirt and Trouser, or categories Bag and Trouser. Therefore, MHCN is capable of guaranteeing meaningful fine-grained hierarchical partitioning with high interpretability for the multi-view data.

5. Conclusion & Future Work

In this work, we present a novel one-stage framework MHCN to solve the multi-view hierarchical clustering problem. With the help of the hyperbolic model of the Poincaré ball, we conduct our multi-view representation learning via multiple hyperbolic autoencoders, where the exponential map of each encoder and its inverse logarithm map of each decoder generalize the extracted latent hierarchies to hyperbolic space efficiently. Additionally, we introduce how to promote the final hierarchical clustering tree from the learned hyperbolic representations by jointly optimizing our designed loss functions, including the multi-view alignment loss for view-common information, the reconstruction loss for view-private information, and the hyperbolic uniformity loss for more balanced embedding distribution on the Poincaré ball. Extensive experiments on seven widespread multi-view datasets demonstrate our method achieves state-of-the-art hierarchical clustering performance. In the future, we expect to improve our method with a more general scheme to better learn the different effects of consistency and complementarity of different multi-view datasets.



(a) MNIST-USPS (b) BDGP (c) Caltech (d) COIL-20 (e) BBCSport (f) Multi-Fashion (g) NR-MNIST

Figure 6: τ_{align} and τ_{uni} sensitivity analysis

Acknowledgments

This work was partially supported by the National Key Research and Development Program of China (No. 2018AAA0100204), a key program of fundamental research from Shenzhen Science and Technology Innovation Commission (No. JCYJ20200109113403826), the Major Key Project of PCL (No. 2022ZD0115301), and an Open Research Project of Zhejiang Lab (NO.2022RC0AB04).

References

[1] MohammadHossein Bateni, Soheil Behnezhad, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, Raimondas Kiveris, Silvio Lattanzi, and Vahab Mirrokni. Affinity clustering: Hierarchical clustering at scale. In *NeurIPS*, pages 6864–6874, 2017.

[2] Eugenio Beltrami. Teoria fondamentale degli spazii di curvatura costante. *Annali di Matematica Pura ed Applicata (1867-1897)*, 2(1):232–255, 1868.

[3] Xiao Cai, Feiping Nie, and Heng Huang. Multi-view k-means clustering on big data. In *IJCAI*, pages 2598–2604, 2013.

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

[5] 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.

[6] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *NeurIPS*, pages 4869–4880, 2019.

[7] Vaggos Chatziafratis, Rad Niazadeh, and Moses Charikar. Hierarchical clustering with structural constraints. In *ICML*, pages 774–783. PMLR, 2018.

[8] Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *ICML*, pages 129–136, 2009.

[9] Giovanni Chierchia and Benjamin Perret. Ultrametric fitting by gradient descent. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124004, 2020.

[10] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM (JACM)*, 66(4):1–42, 2019.

[11] Henry Cohn and Abhinav Kumar. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*, 20(1):99–148, 2007.

[12] Shuyang Dai, Zhe Gan, Yu Cheng, Chenyang Tao, Lawrence Carin, and Jingjing Liu. Apo-vae: Text generation in hyperbolic space. *arXiv preprint arXiv:2005.00054*, 2020.

[13] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. Ieee, 2005.

[14] Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In *STOC*, pages 118–127, 2016.

[15] Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

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

[17] Joseph Felsenstein and Joseph Felenstein. *Inferring phylogenies*, volume 2. Sinauer associates Sunderland, MA, 2004.

[18] Fangxiang Feng, Xiaojie Wang, and Ruifan Li. Cross-modal retrieval with correspondence autoencoder. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 7–16, 2014.

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

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

[21] Mina GhadimiAtigh, Julian Schoep, Erman Acar, Nanne van Noord, and Pascal Mettes. Hyperbolic image segmentation. *arXiv preprint arXiv:2203.05898*, 2022.

[22] Zoubin Ghahramani, Michael Jordan, and Ryan P Adams. Tree-structured stick breaking for hierarchical data. In *NeurIPS*, volume 23, 2010.

[23] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *ICCV*, pages 5736–5745, 2017.

[24] Prasoon Goyal, Zhiting Hu, Xiaodan Liang, Chenyu Wang, and Eric P Xing. Nonparametric variational auto-encoders for hierarchical representation learning. In *ICCV*, pages 5094–5102, 2017.

[25] Katherine A Heller and Zoubin Ghahramani. Bayesian hierarchical clustering. In *ICML*, pages 297–304, 2005.

[26] Joy Hsu, Jeffrey Gu, Gong Wu, Wah Chiu, and Serena Yeung. Capturing implicit hierarchical structure in 3d biomedical images with self-supervised hyperbolic representations. In *NeurIPS*, pages 5112–5123, 2021.

[27] Shudong Huang, Yixi Liu, Yazhou Ren, Ivor W Tsang, Zenglin Xu, and Jiancheng Lv. Learning smooth representation for multi-view subspace clustering. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3421–3429, 2022.

[28] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[29] Yangbangyan Jiang, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Dm2c: Deep mixed-modal clustering. In *NeurIPS*, pages 5880–5890, 2019.

[30] Zhao Kang, Guoxin Shi, Shudong Huang, Wenyu Chen, Xiaorong Pu, Joey Tianyi Zhou, and Zenglin Xu. Multi-graph fusion for multi-view spectral clustering. *Knowledge-Based Systems*, 189:105102, 2020.

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

[32] John M Lee. Smooth manifolds. In *Introduction to Smooth Manifolds*, pages 1–31. Springer, 2013.

[33] Ruihuang Li, Changqing Zhang, Huazhu Fu, Xi Peng, Tianyi Zhou, and Qinghua Hu. Reciprocal multi-layer subspace learning for multi-view clustering. In *ICCV*, pages 8172–8180, 2019.

[34] Yingming Li, Ming Yang, and Zhongfei Zhang. A survey of multi-view representation learning. *TKDE*, 31(10):1863–1883, 2018.

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

[36] Dongdong Lin, Jigang Zhang, Jingyao Li, Chao Xu, Hong-Wen Deng, and Yu-Ping Wang. An integrative imputation method based on multi-omics datasets. *BMC bioinformatics*, 17(1):1–12, 2016.

[37] 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.

[38] Bulou Liu, Bing Bai, Weibang Xie, Yiwen Guo, and Hao Chen. Task-optimized user clustering based on mobile app usage for cold-start recommendations. In *KDD*, pages 3347–3356, 2022.

[39] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Li Liu, Siqi Wang, Weixuan Liang, and Jiangyong Shi. One-pass multi-view clustering for large-scale data. In *ICCV*, pages 12344–12353, 2021.

[40] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. In *NeurIPS*, volume 31, 2018.

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

[42] Emile Mathieu, Charline Le Lan, Chris J Maddison, Ryota Tomioka, and Yee Whye Teh. Continuous hierarchical representations with poincaré variational auto-encoders. In *NeurIPS*, volume 32, 2019.

[43] 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.

[44] Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[45] Yoshihiro Nagano, Shoichiro Yamaguchi, Yasuhiro Fujita, and Masanori Koyama. A differentiable gaussian-like distribution on hyperbolic space for gradient-based learning. *CoRR*, abs/1902.02992, 2019.

[46] Stanislav Naumov, Grigory Yaroslavtsev, and Dmitrii Avdiukhin. Objective-based hierarchical clustering of deep embedding vectors. In *AAAI*, pages 9055–9063, 2021.

[47] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NeurIPS*, pages 6338–6347, 2017.

[48] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[49] Erlin Pan and Zhao Kang. Multi-view contrastive graph clustering. In *NeurIPS*, volume 34, 2021.

[50] Jiwoong Park, Junho Cho, Hyung Jin Chang, and Jin Young Choi. Unsupervised hyperbolic representation learning via message passing auto-encoders. In *CVPR*, pages 5516–5526, 2021.

[51] Sakshi Patel, Shivani Sihmar, and Aman Jatain. A study of hierarchical clustering algorithms. In *2015 2nd international conference on computing for sustainable global development (INDIACom)*, pages 537–541. IEEE, 2015.

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

[53] 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.

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

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

[56] Su-Jin Shin, Kyungwoo Song, and Il-Chul Moon. Hierarchically clustered representation learning. In *AAAI*, pages 5776–5783, 2020.

[57] Huayi Tang and Yong Liu. Deep safe incomplete multi-view clustering: Theorem and algorithm. In *International Conference on Machine Learning*, pages 21090–21110. PMLR, 2022.

[58] Yee Teh, Hal Daume III, and Daniel M Roy. Bayesian agglomerative clustering with coalescents. In *NeurIPS*, volume 20, 2007.

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

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

[61] Dingkang Wang and Yusu Wang. An improved cost function for hierarchical cluster trees. *Journal of Computational Geometry*, 11(1):283–331, 2020.

[62] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, pages 9929–9939. PMLR, 2020.

[63] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *ICML*, pages 1083–1092, 2015.

[64] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487. PMLR, 2016.

[65] 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.

[66] Jie Xu, Yazhou Ren, Huayi Tang, Xiaorong Pu, Xiaofeng Zhu, Ming Zeng, and Lifang He. Multi-vae: Learning disentangled view-common and view-peculiar visual representations for multi-view clustering. In *ICCV*, pages 9234–9243, 2021.

[67] 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.

[68] Qinghai Zheng, Jihua Zhu, and Shuangxun Ma. Multi-view hierarchical clustering. *arXiv preprint arXiv:2010.07573*, 2020.