Learning Attentive and Hierarchical Representations for 3D Shape Recognition

Jiaxin Chen[1], Jie Qin[1⋆], Yuming Shen[3], Li Liu[1], Fan Zhu[1], and Ling Shao[1,2]

[1] Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
[2] Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
[3] eBay

Abstract. This paper proposes a novel method for 3D shape representation learning, namely Hyperbolic Embedded Attentive Representation (HEAR). Different from existing multi-view based methods, HEAR develops a unified framework to address both multi-view redundancy and single-view incompleteness. Specifically, HEAR firstly employs a hybrid attention (HA) module, which consists of a view-agnostic attention (VAA) block and a view-specific attention (VSA) block. These two blocks jointly explore distinct but complementary spatial saliency of local features for each single-view image. Subsequently, a multi-granular view pooling (MVP) module is introduced to aggregate the multi-view features with different granularities in a coarse-to-fine manner. The resulting feature set implicitly has hierarchical relations, which are therefore projected into a Hyperbolic space by adopting the Hyperbolic embedding. A hierarchical representation is learned by Hyperbolic multi-class logistic regression based on the Hyperbolic geometry. Experimental results clearly show that HEAR outperforms the state-of-the-art approaches on three 3D shape recognition tasks including generic 3D shape retrieval, 3D shape classification and sketch-based 3D shape retrieval.

Keywords: 3D shape recognition; View-agnostic/specific attentions; Multi-granularity view aggregation; Hyperbolic neural networks

1 Introduction

Recently, 3D shape analysis [46, 45, 47, 70, 72, 8, 71, 24, 18, 33] has emerged as a hot research topic in computer vision, due to the increasing demand from real applications in virtual reality, autonomous driving, 3D printing and gaming. Learning 3D shape representations for downstream tasks, *e.g.*, 3D shape classification/retrieval, is a fundamental problem for 3D shape analysis. However, this problem is very challenging, considering the varying modalities, complicated geometries and variability of 3D shapes.

A variety of methods have been proposed to learn 3D shape representations, which can generally be divided into the following two categories: 1) 3D model-based methods, learning representations directly from the raw format of 3D

⋆ indicates the corresponding author.

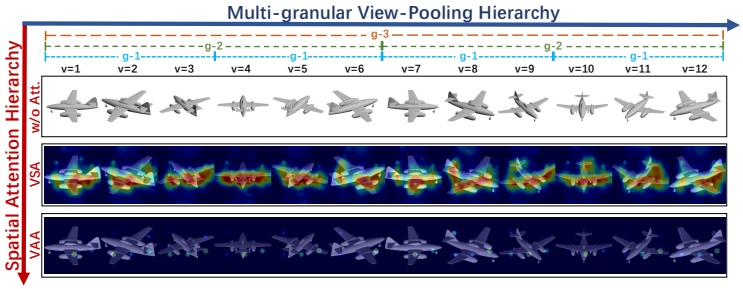

Fig. 1. Illustration of the spatial attention hierarchy as well as the multi-granular view-pooling hierarchy.

shapes, such as point cloud [45, 47, 32], voxel [42, 46, 3] and mesh [14]; 2) multi-view based approaches [6, 51, 1, 56, 2, 13, 25, 21, 72, 24, 71], which first represent a 3D object by a set of rendered 2D images to extract individual features, and then aggregate the features to a global descriptor. Benefiting from the success of CNN in 2D image representation learning, the multi-view based approaches have surpassed their model-based counterparts in most cases. However, it remains difficult to effectively aggregate multi-view data because of their following characteristics: **(a)** *Single-view incompleteness.* As shown in Fig. 1, a 2D image rendered from a single view only captures partial appearance and geometry structures of a 3D object, due to the self-occlusion and information loss by 2D projection during the rendering procedure; **(b)** *Multi-view redundancy.* Multiple images rendered from a sequence of over-completely sampled views contain a large amount of redundant information, since images from neighboring views often capture similar geometric structures of the 3D object and many 3D shapes are geometrically symmetric. This kind of redundancy suppresses the effects of discriminative local regions, which will deteriorate the final performance.

Most of the existing works focus on addressing problem (a) by developing various view-wise pooling strategies [55, 65], exploring view importance [73] or modeling multi-view data by sequence [10, 22, 24], while they improperly neglect problem (b). In our work, we take into account both problems (a) and (b) and propose a unified framework, namely **Hyperbolic Embedded Attentive Representation** (**HEAR**), as illustrated in Fig. 2.

On the one hand, HEAR develops a hybrid attention (HA) module to extensively explore the spatial attentions of local features for each single-view image. Specifically, HA consists of two blocks, *i.e.*, the View Agnostic Attention (VAA) block and the View Specific Attention (VSA) block. Basically, VAA attempts to learn high-level spatial attentions by adopting a trainable spatial attention network shared across different views. In contrast, the parameter-free VSA aims to explore low-level view-specific spatial attentions by calculating the maximal accumulated top-M correlations with local features from other views. As shown in Fig. 1, VAA and VSA capture complementary spatial attentions that corre-

spond to discriminative local parts of a 3D shape. Accordingly, HEAR imposes large weights on salient local features, whilst suppressing less salient ones. In this way, HEAR alleviates the negative effect caused by the multi-view redundancy.

On the other hand, HEAR employs a multi-granular view-pooling (MVP) module to aggregate multi-view features. Concretely, as shown in Fig. 1, MVP evenly partitions the 12 views into 1,2,4 non-overlapped segments, in each of which the views are ensembled by average/max pooling. In this manner, MVP can preserve more visual details by using this coarse-to-fine view aggregation strategy, and thus can mitigate the single-view incompleteness, as mentioned in problem (a). Based on HA and MVP, a 3D shape can be represented by a set of features, which encode distinct spatial attentions and view-pooling granularities. As observed in Fig. 1, these features implicitly have hierarchical relations w.r.t. the spatial attention and multi-granular view-pooling. We therefore employ a Hyperbolic embedding, to endow the feature space with a Hyperbolic geometry, which has recently been successfully applied to represent hierarchical structured data [20, 49, 19, 4, 30]. Accordingly, the Hyperbolic multi-class logistic regression (MLR) is applied to accomplish classification/retrieval in the Hyperbolic space.

Our main contributions are summarized as follows:

- We simultaneously address the problems of single-view incompleteness and multi-view redundancy for 3D shape representation learning by a unified framework, namely Hyperbolic Embedded Attentive Representation (HEAR).
- We propose a hybrid attention module to explore view-agnostic and view-specific attentions, which capture distinct but complementary spatial saliency.
- We present a multi-granular view-pooling mechanism to aggregate multi-view features in a hierarchical manner, which are subsequently encoded into a hierarchical representation space by employing the Hyperbolic embedding.

2 Related Work

Model-based methods Several recent works learn representations from raw 3D shape data, which can be divided into the following categories. 1) Voxel-based models such as 3DShapeNet [67], VoxelNet [42], Subvolume Net[46] and VRN [3]. They directly apply the 3D convolution neural networks to learn the representation based on voxelized shapes. However, these approaches are usually computationally costly, and severely affected by the low resolution caused by the data sparsity. 2) Point cloud-based methods. Point cloud is a set of unordered 3D points, which has attracted increasing interests due to its wide applications. Qi *et al.* propose the seminal work, *i.e.*, PointNet [45] by building deep neural networks on the point sets. Afterwards, a large amount of approaches, such as PointNet++ [47], Kd-Networks[32], SO-Net[39], KPConv[62], IntepCNN [41], DPAM [40]), have been proposed to improve PointNet [45] by modeling fine-grained local patterns. 3) Mesh-based methods. A majority of CAD models are stored as meshes, which consist of 3D vertices, edges and faces. [14] presents the MeshNet to solve the complexity and irregularity problems of meshes, and achieves comparable performance with methods using other types of data.

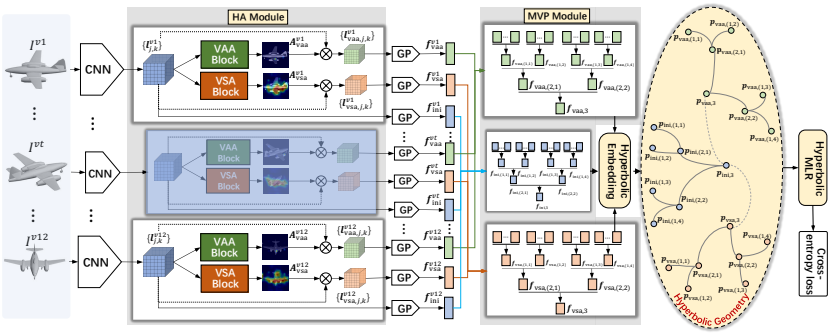

Fig. 2. Framework of Hyperbolic Embedded Attentive Representation (HEAR). HEAR firstly adopts base CNN networks to extract local feature maps, based on which a hybrid attention (HA) module is employed to explore distinct attentions, *i.e.*, the View-Agnostic Attention (VAA) and View-Specific Attention (VSA). The local features re-weighted by each attention are transformed to global features through global pooling (GP). A Multi-granular View Pooling (MVP) is subsequently adopted to aggregate the multi-view global features in a hierarchical manner. The resulting feature set is further endowed with a Hyperbolic geometry through Hyperbolic embedding, and used for classification or retrieval by the Hyperbolic multi-class logistic regression (MLR).

Multi-view based methods The multi-view based method represents a 3D object by a set of 2D images, rendered from a sequence of views. This kind of approaches leverage the well-studied 2D convolutional neural networks, and thus performs better than the model-based ones. In [55], Su *et al.* develops a multi-view convolutional neural network (MVCNN) which extracts features from each single-view image followed by a view-pooling strategy to aggregate the multi-view data into the 3D descriptor. Based on MVCNN, various methods have been proposed by developing different view-wise pooling strategies [65], exploring the view importance [73] and modeling the multi-view data by sequence [10, 22, 24] or graphs [12]. Among these methods, VDN [35] is closely related to our work, which also explores locality attentions. However, VDN mainly focuses on single-view images, and fails to capture cross-view attention patterns. In contrast, our method explores the inter-view correlations by the VSA block. RelationNet [73] also learns the cross-view relations of local features. Nevertheless, our work employs hybrid attentions, as well as considering the multi-granular view-pooling.

3 Proposed Method

3.1 Framework

As shown in Fig. 2, the proposed method mainly consists of three modules: the hybrid attention (HA) module, the multi-granular view pooling (MVP) module and the Hyperbolic neural networks with Hyperbolic embedding (HE).

Specifically, suppose that $\boldsymbol{\mathcal{T}} = \left\{\mathcal{O}^i; y^i\right\}_{i=1}^N$ is a training set of 3D shapes, where $\mathcal{O}^i$ is the i-th 3D shape and $y^i \in \{1, \cdots, C\}$ refers to the class label. We firstly represent the 3D shape $\mathcal{O}^i$ as a group of gray-scale images by rendering with the Phong reflection model [43] from N_v views, which are evenly placed in a plane around the 3D shape. The resulting multi-view representation is denoted by $\boldsymbol{I}^i = \{I^{v,i}\}_{v=1}^{N_v}$, where $I^{v,i}$ is a 2D rendered image of $\mathcal{O}^i$ from the v-th view. In this paper, we use $N_v = 12$ views.

Subsequently, we adopt a base convolutional network (*e.g.*, VGG-A [53], VGG-19 [53] and ResNet-50 [23]) $\mathcal{F}_{\boldsymbol{\theta}}(\cdot)$, parameterized by $\boldsymbol{\theta}$, to extract an initial feature map $\boldsymbol{L}_{\text{ini}}^{v,i} = [\boldsymbol{l}_{j,k}^{v,i}]_{1\leq j\leq H, 1\leq k\leq W} \in \mathbb{R}^{H\times W\times d}$ for each image $I^{v,i}$, where H, W and d denote the height, width and number of channels of the feature map, respectively. $\boldsymbol{l}_{j,k}^{v,i}$ refers to the d-dimensional local feature at the (j,k)-th location. Thereafter, a *view-agnostic attention* (VAA) block $\mathcal{VAA}_{\boldsymbol{\phi}}(\cdot)$ together with a *view-specific attention* (VSA) block $\mathcal{VSA}(\cdot)$ are proposed to learn two different kinds of attention weights for each local feature, which we denote by $\boldsymbol{A}_{\text{vaa}}^{v,i} = [\alpha_{\text{vaa},j,k}^{v,i}] \in \mathbb{R}^{H\times W}$ and $\boldsymbol{A}_{\text{vsa}}^{v,i} = [\alpha_{\text{vsa},j,k}^{v,i}] \in \mathbb{R}^{H\times W}$, respectively. Here, $\boldsymbol{\phi}$ refers to learnable parameters of $\mathcal{VAA}_{\boldsymbol{\phi}}(\cdot)$. Accordingly, we can obtain three local feature maps for a single image $I^{v,i}$: the initial feature map $\boldsymbol{L}_{\text{ini}}^{v,i}$ without attentions, the VAA induced feature map $\boldsymbol{L}_{\text{vaa}}^{v,i} = \left[\alpha_{\text{vaa},j,k}^{v,i} \cdot \boldsymbol{l}_{j,k}^{v,i}\right]_{j,k} \in \mathbb{R}^{H\times W\times d}$, as well as the VSA induced feature map $\boldsymbol{L}_{\text{vsa}}^{v,i} = \left[\alpha_{\text{vsa},j,k}^{v,i} \cdot \boldsymbol{l}_{j,k}^{v,i}\right]_{j,k} \in \mathbb{R}^{H\times W\times d}$. By passing through a global pooling module $GP(\cdot)$, the local feature maps are successively aggregated into three global features $\boldsymbol{f}_{\text{ini}}^{v,i} \in \mathbb{R}^D$, $\boldsymbol{f}_{\text{vaa}}^{v,i} \in \mathbb{R}^D$ and $\boldsymbol{f}_{\text{vsa}}^{v,i} \in \mathbb{R}^D$. For N_v rendering views, we therefore obtain three sets of global features $\boldsymbol{F}_{\text{ini}}^i = [\boldsymbol{f}_{\text{ini}}^{1,i}, \cdots, \boldsymbol{f}_{\text{ini}}^{v,i}, \cdots, \boldsymbol{f}_{\text{ini}}^{N_v,i}]$, $\boldsymbol{F}_{\text{vaa}}^i = [\boldsymbol{f}_{\text{vaa}}^{1,i}, \cdots \boldsymbol{f}_{\text{vaa}}^{v,i}, \cdots, \boldsymbol{f}_{\text{vaa}}^{N_v,i}]$, and $\boldsymbol{F}_{\text{vsa}}^i = [\boldsymbol{f}_{\text{vsa}}^{1,i}, \cdots, \boldsymbol{f}_{\text{vsa}}^{v,i}, \cdots, \boldsymbol{f}_{\text{vsa}}^{N_v,i}] \in \mathbb{R}^{D\times N_v}$. Subsequently, the *multi-granular view pooling* (MVP) $\mathcal{MVP}(\cdot)$ is proposed to aggregate multi-view features $\boldsymbol{F}_{\text{ini}}^i$, $\boldsymbol{F}_{\text{vaa}}^i$ and $\boldsymbol{F}_{\text{vsa}}^i$ in a multi-granular manner. The aggregated features implicitly have hierarchical relations. We therefore employ a Hyperbolic embedding $\mathcal{HE}(\cdot)$ to project them into a Hyperbolic space, and learn hierarchical representations by the Hyperbolic multi-class logistic regression (MLR) with parameters $\boldsymbol{\psi}$. The cross-entropy loss is adopted to train the overall network.

3.2 Hybrid Attentions

In this section, we will elaborate the view-agnostic and view-specific attentions. Without loss of generality, we omit the index i for a more neat description.

View-Agnostic Attention. Basically, the VAA block is a variant of the Squeeze-and-Excitation network [26]. It firstly applies a 1×1 convolutional layer $conv_{1\times 1}(\cdot)$ to squeeze the local feature map $\boldsymbol{L}_{\text{ini}}^v$ to an $H\times W$ matrix $\boldsymbol{E}_{\text{ini}}^v$, which is subsequently flattened into an $HW\times 1$ vector $\overrightarrow{\boldsymbol{E}_{\text{ini}}^v}$. Specifically, the spatial attention map $\boldsymbol{A}_{\text{vaa}}^v$ is computed by:

$$\boldsymbol{A}_{\text{vaa}}^v = Reshape\left(\sigma\left(W_2 \cdot ReLU\left(W_1 \cdot \overrightarrow{\boldsymbol{E}_{\text{ini}}^v}\right)\right)\right), \tag{1}$$

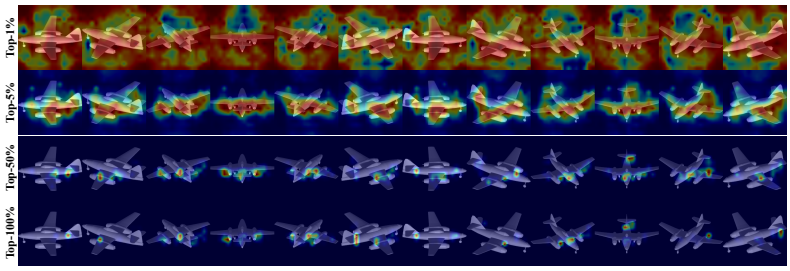

Fig. 3. Visualization of the view-specific attentions by selecting top-M cross-view responses with $M = 1\%/5\%/50\%/100\% \times H \times W$, respectively.

where $W_1 \in \mathbb{R}^{\frac{HW}{r} \times HW}$ and $W_2 \in \mathbb{R}^{HW \times \frac{HW}{r}}$ are learnable parameter matrices, $ReLU(\cdot)$ denotes the ReLU activation function, $\sigma(\cdot)$ is the Softmax activation function, $Reshape(\cdot)$ indicates the operation of reshaping an $HW \times 1$ vector to an $H \times W$ matrix, and r refers to the ratio of dimension reduction.

Note that the parameter matrices W_1 and W_2 of $\mathcal{VAA}_{\boldsymbol{\psi}}(\cdot)$ are shared across different rendering views, *i.e.*, for all $v \in \{1, \cdots, N_v\}$. As a consequence, $\mathcal{VAA}_{\boldsymbol{\psi}}(\cdot)$ are encouraged to pay more attention to view-independent salient local regions. In this regard, we call $\mathcal{VAA}_{\boldsymbol{\psi}}(\cdot)$ the view-agnostic attention block.

By encoding the view-agnostic spatial attention $\boldsymbol{A}^v_{\text{vaa}}$ to the initial feature map, we can obtain the VAA induced feature map $\boldsymbol{L}^v_{\text{vaa}} = \boldsymbol{L}^v_{\text{ini}} \odot \boldsymbol{A}^v_{\text{vaa}}$, where $\odot$ indicates the element-wise production along the channel.

View-Specific Attention. Despite that $\mathcal{VAA}(\cdot)$ is view-agnostic, it tends to neglect some view-dependent local details (as shown in Fig. 1), which are discriminative for distinguishing 3D shapes. Therefore, it is reasonable to explore the view-specific attention as a complement to VAA. To this end, we propose a parameter-free view-specific attention block $\mathcal{VSA}(\cdot)$.

Given a feature map $\boldsymbol{L}^v_{\text{ini}}$ from the v-th view, $\mathcal{VSA}(\cdot)$ aims to compute the spatial attentions by exploring its saliency in feature maps from the rest $N_v - 1$ views, *i.e.*, $\{\boldsymbol{L}^w_{\text{ini}} : w \in \{1, \cdots, N_v\}; w \neq v\}$. Specifically, $\mathcal{VSA}(\cdot)$ first densely computes the response $\gamma^v_{j,k}(p, q, w)$ at location (p, q) in $\boldsymbol{L}^w_{\text{ini}}$ w.r.t. $\boldsymbol{l}^v_{j,k}$:

$$\gamma^v_{j,k}(p, q, w) = \frac{\left(\boldsymbol{l}^v_{j,k}\right)^T \cdot \boldsymbol{l}^w_{p,q}}{\|\boldsymbol{l}^v_{j,k}\|_2 \cdot \|\boldsymbol{l}^w_{p,q}\|_2}, \tag{2}$$

where $\boldsymbol{l}^w_{p,q}$ is the local feature at location (p, q) of $\boldsymbol{L}^w_{\text{ini}}$.

As shown in Eq. (2), the response $\gamma^v_{j,k}(p, q, w)$ is actually the Cosine distance between local features, implying that a large $\gamma^v_{j,k}(p, q, w)$ corresponds to a high visual similarity, and vice versa. Subsequently, we select the subset $\tilde{R}^v_{j,k}(w; M)$ with the top-M largest responses from $R^v_{j,k}(w) = \{\gamma^v_{j,k}(p, q, w)\}_{1 \leq p \leq H, 1 \leq q \leq W}$:

$$\tilde{R}^v_{j,k}(w; M) = \text{argmax}_{R' \subset R, |R'| = M} \sum\nolimits_{\gamma \in R'} \gamma, \tag{3}$$

where $|R^{'}|$ indicates the number of elements in $R^{'}$.

The unnormalized view-specific spatial attention at location (j,k) for the v-th view is then formulated as:

$$\tilde{\alpha}^{v}_{\text{vsa},j,k} = \max_{w\in\{1,\cdots,N_v;w\neq v\}} \sum_{\gamma\in\tilde{R}^{v}_{j,k}(w;M)} \gamma. \tag{4}$$

The normalized view-specific attention $\boldsymbol{A}^{v}_{\text{vsa}}$ can be obtained by applying the Softmax function as follows: $\boldsymbol{A}^{v}_{\text{vsa}} = \text{Softmax}\left([\tilde{\alpha}^{v}_{\text{vsa},j,k}]\right)$.

From Eqs. (3) and (4), we can observe that $\boldsymbol{A}^{v}_{\text{vsa}}$ firstly explores the non-local saliency of $\boldsymbol{l}^{v}_{j,k}$ in each cross-view feature map $\boldsymbol{L}^{w}_{\text{ini}}$. Different from the standard non-local mean operation [69] used in non-local attentions, we adopt the top-M largest responses. As illustrated in Fig. 3, for a large M, the local patch $\boldsymbol{l}^{v}_{j,k}$ with high non-local co-occurrence appearance will have a large value. And the attentions are concentrated on an extremely small number of local parts, which may lose some discriminative local details. For a small M, $\boldsymbol{l}^{v}_{j,k}$ with high local co-occurrence will have a large value. In this case, the attentions become more diverse, but will be more sensitive to noise (*e.g.*, backgrounds). In order to simultaneously maintain more local details and remove outliers, we set M to a mediate value, which is fixed to $5\% \times H \times W$ in our work.

Similar to VAA, we encode the view-specific attention $\boldsymbol{A}^{v}_{\text{vsa}}$ to the initial feature map, and attain the VSA induced feature map as $\boldsymbol{L}^{v}_{\text{vsa}} = \boldsymbol{L}^{v}_{\text{ini}} \odot \boldsymbol{A}^{v}_{\text{vsa}}$.

By using the global pooling in base networks (*e.g.*, the global average pooling in ResNet-50), the original local feature map $\boldsymbol{L}^{v,i}_{\text{ini}}$, the VAA induced local feature map $\boldsymbol{L}^{v,i}_{\text{vaa}}$ and the VSA induced local feature map $\boldsymbol{L}^{v,i}_{\text{vsa}}$ of $\mathcal{O}^i$ are aggregated into three sets of global features $\{\boldsymbol{f}^{v,i}_{\text{ini}}\}^{N_v}_{v=1}$, $\{\boldsymbol{f}^{v,i}_{\text{vaa}}\}^{N_v}_{v=1}$ and $\{\boldsymbol{f}^{v,i}_{\text{vsa}}\}^{N_v}_{v=1}$, respectively.

3.3 Hierarchical Representation Learning

Multi-granular View Pooling. As shown in Fig. 1, the rendered 2D images $\{I^{v,i}\}$ of a 3D shape $\mathcal{O}^i$ from different views capture *distinct* but *incomplete* spatial and visual structures of $\mathcal{O}^i$. Conventional methods aggregate the multi-view features from N_v views by using view-level average/max pooling [55, 70, 25, 7, 72], exploring view attentions [24], adopting the sequence modeling model such as the recurrent neural networks [22], or using 3D convolutions [34] as well as graph neural networks [12].

In our work, we develop a multi-granular view pooling (MVP) module to aggregate the multi-view features $\{\boldsymbol{f}^{v,i}_{t}\}^{N_v}_{v=1}$ ($t \in \{\text{ini}, \text{vaa}, \text{vsa}\}$) based on the following three levels of granularity. 1) **Granularity-1 (g-1).** The N_v rendering views are sequentially divided into four groups, each of which consists of $\frac{N_v}{4}$ views. The features in each group are aggregated into one single vector by average pooling, and finally resulting in four vectors $\{\boldsymbol{f}^{i}_{t,(1,g_1)}\}_{g_1=1,\cdots,4}$. 2) **Granularity-2 (g-2).** Similar to Granularity-1, the N_v rendering views are divided into two groups, each of them having $\frac{N_v}{2}$ views and therefore outputting two feature vectors $\{\boldsymbol{f}^{i}_{t,(2,g_2)}\}_{g_2=1,2}$. 3) **Granularity-3 (g-3).** All N_v features are aggregated into the averaged vector $\boldsymbol{f}^{i}_{t,3}$.

The above three sets of aggregated features, *i.e.*, $\boldsymbol{f}^i_{t,3}$, $\{\boldsymbol{f}^i_{t,(2,g_2)}\}_{g_2=1,2}$ and $\{\boldsymbol{f}^i_{t,(1,g_1)}\}_{g_1=1,\cdots,4}$, capture different view-dependent visual details of $\mathcal{O}^i$ in a coarse-to-fine granularity. In this way, we desire to mitigate the single-view incompleteness problem as aforementioned.

Based on the hybrid attentions and multi-granular view pooling, a 3D object $\mathcal{O}^i$ can be represented by a feature set $F^i = \{\{\boldsymbol{f}^i_{t,(1,g_1)}\}, \{\boldsymbol{f}^i_{t,(2,g_2)}\}, \boldsymbol{f}^i_{t,3} : t \in \{\text{ini}, \text{vaa}, \text{vsa}\}\}$. We note that F^i has the following kinds of hierarchical relations:

1) *Spatial Attention Hierarchy.* As shown in Fig. 1, the global feature $\boldsymbol{f}^{v,i}_{\text{ini}}$ is pooled from the original feature map $\boldsymbol{L}^{v,i}_{\text{ini}}$, which equally treats each local feature. Therefore, $\boldsymbol{f}^{v,i}_{\text{ini}}$ represents the most diversified but less salient visual information. In contrast, $\boldsymbol{f}^{v}_{\text{vaa}}$, which is pooled from the VAA induced feature map $\boldsymbol{L}^{v}_{\text{vaa}}$, encodes extremely concentrated but salient local details. $\boldsymbol{f}^{v}_{\text{vsa}}$, derived from $\boldsymbol{L}^{v}_{\text{vaa}}$, alternatively makes a trade-off, and intermediately keep the diversity and saliency controlled by K. In this manner, $\boldsymbol{f}^{v,i}_{\text{ini}}$, $\boldsymbol{f}^{v,i}_{\text{ini}}$ and $\boldsymbol{f}^{v,i}_{\text{ini}}$ have hierarchical relations in regard to the diversity and saliency of spatial attentions.

2) *Pooling-view Hierarchy.* As described above, $\boldsymbol{f}^i_{t,3}$, $\{\boldsymbol{f}^i_{t,(2,g_2)}\}_{g_2=1,2}$ and $\{\boldsymbol{f}^i_{t,(1,g_1)}\}_{g_1=1,\cdots,4}$ aggregate multi-view features using the full Nv, partially $\frac{Nv}{2}$ and $\frac{Nv}{4}$ views, respectively. As a consequence, they naturally have hierarchical relations in terms of the aggregation granularity.

Based on the above two observations, we therefore leverage the Hyperbolic geometry to learn the embedding of F^i, due to their intrinsic capability of representing hierarchies, such as the tree graphs, taxonomies and linguistic ontology in natural language processing (NLP) [49, 50].

Hyperbolic Space. Formally, we denote a D-dimensional Hyperbolic space by $\mathbb{H}^D$, which is defined as a simply connected n-dimensional Riemannian manifold of constant negative sectional curvature. Basically, there exist many distinct but isomorphic models of the Hyperbolic geometry. In our work, we adopt the Poincaré ball model, considering its preveiling applications in NLP and its numerical stability as well.

Specifically, a Poincaré ball is a manifold $\mathbb{P}^D_c = \{\boldsymbol{x} \in \mathbb{R}^D : c\|\boldsymbol{x}\|^2 < 1, c \geq 0\}$ endowed with the Riemannian metric $g^{\mathbb{P}}(\boldsymbol{x}) = (\lambda^c_{\boldsymbol{x}})^2 g^{\mathbb{E}}$, where $\lambda^c_{\boldsymbol{x}} = \frac{2}{1-c\|\boldsymbol{x}\|^2}$ is the conformal factor and $g^{\mathbb{E}}$ is the Euclidean metric tensor, *i.e.*, $g^{\mathbb{E}} = \mathbf{I}_D$. Note that in a standard definition of the Poincaré ball, c equals to 1. We follow [19] and introduce the hyperparameter c to represent the radius of the Poincaré ball. Actually, c allows one to make a balance between the Hyperbolic and Euclidean geometry, considering that $\mathbb{P}^D_c$ converges to the Euclidean space $\mathbb{R}^D$ as $c \to 0$.

Basic Operations in $\mathbb{P}^D_c$. To formulate our method, we introduce some basic arithmetic operations in the Hyperbolic space.

Möbius addition. Given two points $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{P}^D_c$, $\boldsymbol{p} \oplus_c \boldsymbol{q}$ refers to the Möbius addition, which is defined as the following:

$$\frac{(1 + 2c < \boldsymbol{p}, \boldsymbol{q} > +c\|\boldsymbol{q}\|^2) \cdot \boldsymbol{p} + (1 - c\|\boldsymbol{q}\|^2) \cdot \boldsymbol{q}}{1 + 2c < \boldsymbol{p}, \boldsymbol{q} > +c^2\|\boldsymbol{p}\|^2\|\boldsymbol{q}\|^2}, \tag{5}$$

where $< \cdot >$ refers the Euclidean inner product, and $\|\|$ is the l-2 vector norm.

Geodesic distance. Based on $\oplus_c$, the geodesic distance is formuated as:

$$d_c(\boldsymbol{p}, \boldsymbol{q}) = \frac{2}{\sqrt{c}} \text{arctanh}(\sqrt{c} \cdot \| - \boldsymbol{p} \oplus_c \boldsymbol{q} \|). \tag{6}$$

Möbius matrix-vector product. Suppose we have a standard Euclidean linear matrix $\boldsymbol{M} \in \mathbb{R}^{d \times D}$, the Möbius matrix-vector product $\boldsymbol{M}^{\otimes_c}(\boldsymbol{p})$ between $\boldsymbol{M}$ and $\boldsymbol{p}$ is defined as follows:

$$\boldsymbol{M}^{\otimes_c}(\boldsymbol{p}) = \frac{1}{\sqrt{c}} \tanh\left(\frac{\|\boldsymbol{M}\boldsymbol{p}\|}{\|\boldsymbol{p}\|} \text{arctanh}(\sqrt{c}\|\boldsymbol{p}\|) \right) \frac{\boldsymbol{M}\boldsymbol{p}}{\|\boldsymbol{M}\boldsymbol{p}\|}, \tag{7}$$

if $\boldsymbol{M}\boldsymbol{p} \neq \boldsymbol{0}$, and otherwise $\boldsymbol{M}^{\otimes_c}(\boldsymbol{p}) = \boldsymbol{0}$.

Hyperbolic Embedding. Usually, the feature set F^i is not located in $\mathbb{P}_c^D$. We utilize the *exponential map* $\exp_{\boldsymbol{0}}^c$ at $\boldsymbol{0}$ as the projection from $\mathbb{R}^D$ to $\mathbb{P}_c^D$:

$$\boldsymbol{0} \oplus_c \left(\tanh\left(\sqrt{c} \cdot \frac{\lambda_{\boldsymbol{0}}^c \cdot \|\boldsymbol{f}\|}{2} \right) \frac{\boldsymbol{f}}{\sqrt{c} \cdot \|\boldsymbol{f}\|} \right). \tag{8}$$

The inverse projection *logarithmic map* $\log_{\boldsymbol{0}}^c(\boldsymbol{y})$ from $\mathbb{P}_c^D$ to $\mathbb{R}^D$ is defined by:

$$\frac{2}{\sqrt{c} \cdot \lambda_{\boldsymbol{0}}^c} \text{arctanh}(\sqrt{c} \cdot \| - \boldsymbol{0} \oplus_c \boldsymbol{y}\|) \frac{-\boldsymbol{0} \oplus_c \boldsymbol{y}}{\| - \boldsymbol{0} \oplus_c \boldsymbol{y}\|}. \tag{9}$$

As a result, F^i is projected to the Hyperbolic space and turned to

$$P^i = \left\{ \{\boldsymbol{p}_{t,(1,g_1)}^i\}, \{\boldsymbol{p}_{t,(1,g_2)}^i\}, \boldsymbol{p}_{t,g_3}^i \right\},$$

where $\boldsymbol{p}_{t,(1,g_1)}^i = \exp_{\boldsymbol{0}}^c(\boldsymbol{f}_{t,(1,g_1)}^i)$, $\boldsymbol{p}_{t,(1,g_2)}^i = \exp_{\boldsymbol{0}}^c(\boldsymbol{f}_{t,(1,g_2)}^i)$ and $\boldsymbol{p}_{t,g_3}^i = \exp_{\boldsymbol{0}}^c(\boldsymbol{f}_{t,g_3}^i)$. We can finally obtain the 3D representation $\boldsymbol{p}^i \in \mathbb{P}_c^{d'}$ by using the following vector concatenation in P^i:

$$\mathcal{HE}(F^i) = \boldsymbol{M}^{\oplus_c}(\boldsymbol{p}_{\text{ini},(1,\text{g}_1)}^i) \oplus_c \cdots \oplus_c \boldsymbol{M}^{\oplus_c}(\boldsymbol{p}_{\text{vsa},\text{g}_3}^i),$$

where $\boldsymbol{M} \in \mathbb{R}^{d' \times D}$ is the parameter matrix of $\mathcal{HE}(\cdot)$, and d' indicates the dimension of the concatenated feature vector.

Hyperbolic Neural Networks. In order to perform 3D shape classification and retrieval, we leverage the generalized multi-class logistic regression (MLR) to Hyperbolic spaces [19]. The basic idea lies in the following observation: the logits of MLR in the Euclidean space can be represented as the distances to certain hyperplanes, where each hyperplane can be specified with a point of origin and a normal vector. This observation can be extended to the Poincaré ball $\mathbb{P}_c^n$. Specifically, suppose C points $\{\boldsymbol{h}_k \in \mathbb{P}_c^n\}_{k=1}^C$ and normal vectors $\{\boldsymbol{a}_k \in T_{\boldsymbol{h}_k}\mathcal{P}_c^n \backslash \{\boldsymbol{0}\}\}_{k=1}^C$ are learnable parameters, where $T_{\boldsymbol{h}_k}$ stands for the tangent space at $\boldsymbol{h}_k$. Given a feature $\boldsymbol{p} \in \mathbb{P}_c^n$, the Hyperbolic MLR $\mathcal{H}_{\boldsymbol{\psi}}(\cdot)$ for C classes is thereafter formulated as follows:

$$p_k(\boldsymbol{p}) = p(y = k|\boldsymbol{p}) \propto \exp\left(\frac{\lambda_{\boldsymbol{h}_k}^c \|\boldsymbol{a}_k\|}{\sqrt{c}} \text{arcsinh}\left(\frac{2\sqrt{c}\langle -\boldsymbol{h}_k \oplus_c \boldsymbol{p}, \boldsymbol{a}_k \rangle}{(1 - c\|-\boldsymbol{h}_k \oplus_c \boldsymbol{p}\|^2)\|\cdot \boldsymbol{a}_k\|} \right) \right). \tag{10}$$

Based on Eq. (10), we then apply the cross-entropy loss for the concatenated feature $\boldsymbol{p}^i$ as well as for all the individual features $\{\boldsymbol{p}^i_{\text{ini},(1,\text{g}_1)}, \cdots, \boldsymbol{p}^i_{\text{vsa},\text{g}_3}\}$:

$$\mathcal{L}_{xent} = -\frac{1}{N}\sum_{i}^{N}\sum_{k=1}^{C}\sum_{\boldsymbol{p}\in P^i\cup\{\boldsymbol{p}^i\}} y^i_k \cdot \log(p_k(\boldsymbol{p})). \tag{11}$$

Optimization. As shown in Eq. (11), the parameters $\{\boldsymbol{h}_k\}_{k=1}^C$ of the Hyperbolic MLR $\mathcal{H}_{\boldsymbol{\psi}}(\cdot)$ are located inside the Poincaré ball. One way to optimize $\mathcal{H}_{\boldsymbol{\psi}}(\cdot)$ is using the Riemannian Adam optimizer [17] with pre-conditioners [63, 64]. However, as suggested in [30], we utilize a more efficient yet effective solution, *i.e.*, first optimizing $\{\boldsymbol{h}_k\}_{k=1}^C$ via the standard Adam optimizer, and then mapping them to their Hyperbolic counterparts with the exponential map $\exp^c_{\mathbf{0}}(\cdot)$.

4 Experimental Results and Analysis

4.1 3D Shape Classification and Retrieval

Datasets. For 3D shape classification and retrieval, we conduct experiments on two widely used datasets: **ModelNet10** and **ModelNet40**, both of which are subsets of ModelNet [67] with 151,128 3D CAD models from 660 categories. ModelNet10 includes 4,899 3D shapes belonging to 10 classes. We follow the 3,991/908 training/test split as commonly used in literature [73]. ModelNet40 contains 12,311 3D shapes from 40 categories. For 3D shape retrieval, most existing works select 80/20 objects per class for training/testing [67] and [55]. In regard to 3D shape classification, more recent works use the full split [46, 45, 47, 12, 73], which has 9,843/2,468 training/test 3D models. Therefore, we adopt the 80/20 split for 3D shape retrieval, and the full split for 3D shape classification.
Evaluation Metrics. As for classification, we follow previous works and report both *per instance accuracy* and *per class accuracy*, regarding the class-imbalance problem in the ModelNet40 dataset. Concretely, the per instance accuracy is the percentage of correctly classified 3D models among all the whole test set, and the per class accuracy refers to the averaged accuracy per class. To evaluate the retrieval performance, we report the widely used *mean Average Precision (mAP)* and *Area Under Curve (AUC)* of the precision-recall curve.
Implementation Details. Following the identical rendering protocol as MVCNN [55], we render a 3D object to a set of 2D 224×224 greyscale images by placing virtual cameras around the 3D model every 30 degrees. Each 3D shape is then represented by 12 view images.

As suggested in MVCNN [56], we train our model by two stages. In the first stage, we adopt the the CNN backbone network pre-trained on ImageNet [48], and fine-tune it on the training set by training as a single-view image classification task. In the second stage, we initialize the convolutional layers with the model fine-tuned in stage 1, and train the full model in Fig. 2, by removing the fully-connected classifier. We adopt the Adam optimizer [31], and set the learning rate to 5×10^{-5} and 1×10^{-5} for the first and second stages, respectively. For both stages, the model is trained within 30 epochs with weight decay

Table 1. Comparison results on 3D shape classification. (Best results in **bold**.)

Method	Reference	Input Modality	ModelNet40		ModelNet10	
			Per instance (%)	Per class (%)	Per instance (%)	Per class (%)
SPH [29]	SPG2003	Hand-crafted	-	68.2	-	-
LFD [6]	CGF2003	Hand-crafted	-	75.5	-	40.9
Subvolume Net [46]	CVPR2016	Volume	89.2	86.0	-	-
Voxception-ResNet [3]	NIPS2016	Volume	91.3	-	93.6	-
PointNet++ [47]	NIPS2017	Points	91.9	-	-	-
SO-Net [39]	CVPR 2018	Points	93.4	90.8	95.7	95.5
DensePoint [40]	ICCV 2019	Points	93.2	-	96.6	-
MeshNet [14]	AAAI 2019	Mesh	-	-	93.1	-
MVCNN^{V-M} [55]	CVPR2015	Multi-view (#Views=12)	92.1	89.9	-	-
MVCNN-MultiRes^{V-M} [46]	CVPR2016	Multi-resolution Views	93.8	91.4	-	-
MVCN-New^{R-50} [56]	ECCVW2018	Multi View (#Views=12)	95.5	94.0	-	-
Pairwise Network^{V-M} [27]	CVPR2016	Multi-view (#Views=12)	-	91.1	-	93.2
GVCNN^{G} [13]	CVPR2018	Multi-view	93.1	-	-	-
RotationNet^{R-50} [28]	ICCV2017	Multi-view (#Views=12)	-	-	94.8	-
MHBN^{V-M} [75]	CVPR2018	Multi-view	94.1	92.2	94.9	94.9
HGNN [73]	AAAI2019	Multi-view (#Views=12)	**96.7**	-	-	-
RelationNet^{V-M} [73]	ICCV2019	Multi-view (#Views=12)	94.3	92.3	95.3	95.1
HEAR^{V-M}	Ours	Multi-view (#Views=12)	95.5	94.2	98.2	98.1
HEAR^{R-50}	Ours	Multi-view (#Views=12)	**96.7**	**95.2**	**98.6**	**98.5**

($V-M/G/R-50$ **indicate VGG-M [5]/GoogLeNet [58]/ResNet-50 [23].**)

0.001. As to the Hyperbolic embedding, we set the hyper-parameter c and the dimension d' of concatenated features as 5×10^{-5} and 1,024, respectively. All the experiments are conducted on a Telsa V100 GPU.

As summarized in Table 1, HEAR achieves the best performance on ModelNet40 and ModelNet10, when using the same base networks. For instance, the per instance/class accuracy of HEAR is 1.6%/2.6% higher than the second best one on ModelNet10, when using VGG-M. With the ResNet-50 backbone, the performance of HEAR can be further improved. Note that HGNN achieve the same per instance accuracy, *i.e.*, 96.7%, as ours on ModelNet40. However, it combines multiple types of deep features including GVCNN and MVCNN, while HEAR only requires one backbone. In [28], a higher result is reported for RotationNet, by extensively exploring the rendering view coordinates. In contrast, our method uses the standard 12 fixed views. For a fair comparison, we only report the averaged accuracy of RotationNet.

Results on 3D Shape Classification. We compare our method with the hand-crafted [29, 6], voxel-based [46, 3], points-based [47, 39, 40], mesh-based [14], and multi-view based approaches [55, 46, 56, 27, 13, 28, 75, 12, 73]. As for the multi-view based methods, different base networks are utilized, such as VGG-M [5], GoogLeNet [58] and ResNet-50 [23]. We adopt the VGG-M and ResNet-50 to make a fair comparison, which are used by most existing works.

Results on 3D Shape Retrieval. We compare HEAR with the state-of-the-art approaches including model-based [29, 67, 16] and multi-view based ones [1, 56, 2, 13, 25, 21, 35, 72, 24, 71]. We report the results of our method based on three commonly-used backbones (*i.e.*, VGG-A, VGG-19 and ResNet-50). In addition, we use the 1,024-dimensional concatenated vector after Hyperbolic embedding as

Table 2. Comparison results on 3D shape retrieval. (Best results in **bold**.)

Method	Reference	ModelNet40		ModelNet10	
		AUC	MAP	AUC	MAP
SPH [29]	SPG2003	34.5	33.3	46.0	44.1
3DShapeNet [68]	CVPR2015	49.9	49.2	69.3	68.3
DLAN[16]	BMVC2016	-	85.0	-	90.6
MVCNN^{V-M} [55]	CVPR2015	-	80.2	-	-
MVCNN^{V-A} [55]	CVPR2015	73.7	72.9	80.8	80.1
GIFT^{V-S} [1]	CVPR2016	83.1	81.9	92.4	91.1
RED $^{R-50}$ [2]	ICCV2017	87.0	86.3	93.2	92.2
GVCNN ^{G} [13]	CVPR2018	-	85.7	-	-
TCL^{V-A} [25]	CVPR2018	89.0	88.0	-	-
SeqViews^{V-19} [22]	TIP2018	-	89.1	-	91.4
VDN^{G} [35]	TVCG2018	87.6	86.6	93.6	93.2
Batch-wise [72]	CVPR2019	-	83.8	-	87.5
VNN $^{V-A}$ [24]	ICCV2019	89.6	88.9	93.5	92.8
VNN $^{V-19}$ [24]	ICCV2019	90.2	89.3	-	-
NCENet ^{G} [71]	ICCV2019	88.0	87.1	-	-
HEAR^{V-A}	Ours	91.8	91.1	95.0	94.2
HEAR^{V-19}	Ours	92.5	91.6	95.3	94.4
HEAR^{R-50}	Ours	**92.8**	**92.0**	**95.5**	**94.7**

($V-S/V-A/V-19$ **indicate VGG-S [53]/VGG-A [53]/VGG-19 [53].**)

the representation. Note that we **do not employ the triplet loss** as commonly used in 3D shape retrieval, and only use the cross-entropy loss for training.

The comparison results are summarized in Table 2. As shown, our mehod remarkably outperforms the state-of-the-art methods, and achieves 2.7% and 1.5% improvement w.r.t. mAP on ModelNet40 and ModelNet10, respectively. The improvement is consistent, regardless of the choice of backbone networks.

4.2 Sketch-based 3D Shape Retrieval

Datasets. SHREC'13 [36] contains 7,200 human-drawn sketches, and 1,258 shapes from 90 classes, which are collected from the Princeton Shape Benchmark (PSB) [52]. There are a total of 80 sketches per class, 50 of which are selected for training and the rest for test. **SHREC'14** [38] consists of 13,680 sketches and 8,987 3D shapes belonging to 171 classes. There are 80 sketches, and around 53 3D shapes on average per class. The sketches are split into 8,550 image for training and 5,130 for testing.

Evaluation Metrics. We utilize the following widely-adopted metrics [37, 11, 70] for sketch-based 3D shape retrieval: *nearest neighbor (NN)*, *first tier (FT)*, *second tier (ST)*, *E-measure (E)*, *discounted cumulated gain (DCG)* as well as the *mean average precision (mAP)*.

Implementation Details. We employ the ResNet-50 and Inception-ResNet-v2 as the base network, similar to the state-of-the-art methods [44, 7, 8]. We follow the same 'two branch' architecture as depicted in [44], *i.e.*, one branch for sketches and the other one for 3D shapes. The same batch-hard triplet loss and cross-entropy loss in [44] are utilized for training. The only difference between our method and [44] lies in the designed 3D shape branch. The learning rate is set to 3×10^{-5} with decay rate 0.9 for every 20,000 steps.

Table 3. Comparison results on sketch-based 3D shape retrieval. (Best results in **bold**.)

Method	Reference	SHREC'13						SHREC'14					
		NN	FT	ST	E	DCG	mAP	NN	FT	ST	E	DCG	mAP
CDMR [15]	ICW2013	27.9	20.3	29.6	16.6	45.8	25.0	10.9	5.7	8.9	4.1	32.8	5.4
SBR-VC [36]	SHREC13'track	16.4	9.7	14.9	8.5	34.8	11.4	9.5	5.0	8.1	3.7	31.9	5.0
SP [54]	JVLC2010	1.7	1.6	3.1	1.8	24.0	2.6	-	-	-	-	-	-
FDC [36]	SHREC13'track	11.0	6.9	10.7	6.1	30.7	8.6	-	-	-	-	-	-
DB-VLAT [61]	SIPAASC2012	-	-	-	-	-	-	16.0	11.5	17.0	7.9	37.6	13.1
CAT-DTW [74]	VC2017	23.5	13.5	19.8	10.9	39.2	14.1	13.7	6.8	10.2	5.0	33.8	6.0
Siamese [66]	CVPR2015	40.5	40.3	54.8	28.7	60.7	46.9	23.9	21.2	31.6	14.0	49.6	22.8
KECNN [59]	NC2017	32.0	31.9	39.7	23.6	48.9		-	-	-	-	-	-
DCML [11]	AAAI2017	65.0	63.4	71.9	34.8	76.6	67.4	27.2	27.5	34.5	17.1	49.8	28.6
DCHML [9]	TIP2018	73.0	71.5	77.3	36.8	81.6	74.4	40.3	32.9	39.4	20.1	54.4	33.6
LWBR [70]	CVPR2017	71.2	72.5	78.5	36.9	81.4	75.2	40.3	37.8	45.5	23.6	58.1	40.1
DCML^{R-50} [11]	TIP2017	74.0	75.2	79.7	36.5	82.9	77.4	57.8	59.1	64.7	72.3	35.1	61.5
LWBR^{R-50} [70]	CVPR2017	73.5	74.5	78.4	35.9	82.5	76.7	62.1	64.1	69.1	76.0	36.1	66.5
Shape2Vec [60]	TOG2016	-	-	-	-	-	-	71.4	69.7	74.8	36.0	81.1	72.0
DCA^{R-50} [7]	ECCV2018	78.3	79.6	82.9	37.6	85.6	81.3	77.0	78.9	82.3	39.8	85.9	80.3
Semantic^{IR} [44]	BMCV2018	82.3	82.8	86.0	40.3	88.4	84.3	80.4	74.9	81.3	39.5	87.0	78.0
DSSH^{R-50} [8]	CVPR2019	79.9	81.4	86.0	40.4	87.3	83.1	77.5	78.8	83.1	40.4	87.0	80.6
DSSH^{IR} [8]	CVPR2019	83.1	84.4	88.6	41.1	89.3	85.8	79.6	81.3	85.1	41.2	88.1	82.6
HEAR^{R-50}	Ours	82.1	83.7	87.8	40.9	88.8	85.4	79.2	80.7	84.6	40.9	87.8	82.2
HEAR^{IR}	Ours	**84.2**	**85.6**	**88.8**	**41.3**	**90.0**	**86.9**	**80.9**	**82.6**	**86.3**	**41.4**	**89.0**	**83.6**

(*IR* **represents using Inception-ResNet-v2 [57] as the base network.)**

Experimental Results. We compare HEAR with the state-of-the-art methods for sketch-based 3D shape retrieval, including hand-crafted [15, 36, 54, 36, 61, 74] and deep learning based ones [66, 59, 11, 9, 70, 11, 70, 60, 7, 44, 8].

As summarized in Table 3, our method achieves the best performance on both SHREC'13 and SHREC'14. For instance, by using the same ResNet-50 base model, HEAR improves the mAP of DCML, LWBR, DCA and DSSH by 8.0%, 10.2%, 4.1% and 2.3% on SHREC'13, respectively. Similar improvements can be seen on SHREC'14 and by using the Incpetion-ResNet-v2 backbone network. It also can be seen that the performance margin between HEAR and Semantic^{IR} [44] is significant, though the identical learning objective is applied. This suggests that the proposed network learns more descriptive 3D shape patterns for the respective task. In addition, HEAR works well with different learning objectives, further endorsing its ability to learn compact 3D representations.

4.3 Ablation Study

To evaluate each component of our method, *i.e.*, the Hybrid Attention (HA) module consisting of VAA and VSA, the Multi-granular View Pooling (MVP) module as well as the Hyperbolic Neural Networks with Hyperbolic Embedding (HNet), we conduct ablation studies on ModelNet10 and ModelNet40 for the 3D retrieval task. Specifically, we choose MVCNN with VGG-A network structure as the baseline, denoted by *Baseline_MVCNN*. We then successively add VAA/VSA, HA, MVP and HNet to validate their influences on the performance of HEAR. Note that *Baseline_MVCNN+HA+MVP* uses the concatenation and

Table 4. Ablation study of HEAR w.r.t. mAP by using the VGG-A backbone.

Method	**ModelNet40**	**ModelNet10**
Baseline_MVCNN	72.9	80.1
Baseline_MVCNN+VAA	86.3	88.7
Baseline_MVCNN+VSA	87.5	90.2
Baseline_MVCNN+HA	89.7	92.8
Baseline_MVCNN+HA+MVP	90.0	93.0
Baseline_MVCNN+HA+MVP+HNet (**HEAR**)	91.1	94.2

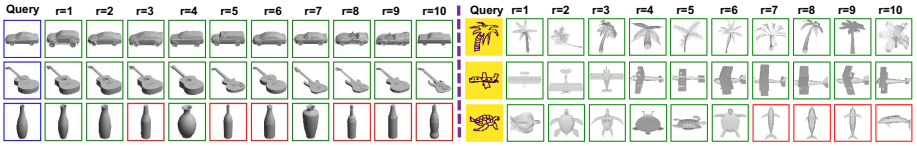

Fig. 4. Retrieval results by using HEAR on ModelNet40 (Left) and SHREC'13 (Right). Images with yellow backgrounds and blue/green/red bounding boxes indicate query 2D sketches, query 3D shapes/correct matches/false matches, respectively.

the standard linear classifier in the Euclidean space, instead of adopting the concatenation with the Hyperbolic embedding and the Hyperbolic MLR.

Table 4 summarizes the mAP of the baselines with different combinations of the components involved. We can observe that both VAA and VSA significantly improve the baseline by exploring the spatial saliency. After combining VAA and VSA, the hybrid attention (HA) can further boost the performance. By employing MVP, HEAR can be slightly improved. The view shift of a 3D object is literally continuous throughout different view points. MVP provides a non-parametric way to perceive this via fusing multi-view data with minimal information wastage. The Hyperbolic embedding and the Hyperbolic MLR can further promote the mAP of HEAR, by endowing and modeling the hierarchical structures in the Hyperbolic space. Without the hyperbolic projection, the proposed model reduces to a conventional representation learning scheme, which is not able to fully acknowledge the structured conceptual similarities.

In addition, we qualitatively show some retrieval results by HEAR in Fig. 4.

5 Conclusion

This paper proposed a novel 3D shape representation method, namely Hyperbolic Embedded Attentive Representation (HEAR). HEAR developed a hybrid attention to explore distinct yet complementary spatial attentions. A multi-granular view-pooling module was subsequently employed to aggregate features from multi-views in a coarse-to-fine hierarchy. The resulting feature set was finally encoded into a hierarchical representation by the Hyperbolic geometry. Experiments on various tasks revealed the superiority of the proposed method.

References

1. Bai, S., Bai, X., Zhou, Z., Zhang, Z., Jan Latecki, L.: Gift: A real-time and scalable 3d shape search engine. In: CVPR (2016)
2. Bai, S., Zhou, Z., Wang, J., Bai, X., Jan Latecki, L., Tian, Q.: Ensemble diffusion for retrieval. In: ICCV (2017)
3. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Generative and discriminative voxel modeling with convolutional neural networks. In: NeurIPS (2016)
4. Chami I, Ying R, R.C.L.J.: Hyperbolic graph convolutional neural networks. In: NeurIPS (2019)
5. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531 (2014)
6. Chen, D.Y., Tian, X.P., Shen, Y.T., Ouhyoung, M.: On visual similarity based 3d model retrieval. In: Computer Graphics Forum. vol. 22, pp. 223–232. Wiley Online Library (2003)
7. Chen, J., Fang, Y.: Deep cross-modality adaptation via semantics preserving adversarial learning for sketch-based 3d shape retrieval. In: ECCV (2018)
8. Chen, J., Qin, J., Liu, L., Zhu, F., Shen, F., Xie, J., Shao, L.: Deep sketch-shape hashing with segmented 3d stochastic viewing. In: CVPR (2019)
9. Dai, G., Xie, J., Fang, Y.: Deep correlated holistic metric learning for sketch-based 3d shape retrieval. IEEE Transactions on Image Processing (2018)
10. Dai, G., Xie, J., Fang, Y.: Siamese cnn-bilstm architecture for 3d shape representation learning. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. pp. 670–676. IJCAI'18 (2018)
11. Dai, G., Xie, J., Zhu, F., Fang, Y.: Deep correlated metric learning for sketch-based 3d shape retrieval. In: AAAI (2017)
12. Feng, Y., You, H., Zhang, Z., Ji, R., Gao, Y.: Hypergraph neural networks. In: AAAI (2019)
13. Feng, Y., Zhang, Z., Zhao, X., Ji, R., Gao, Y.: Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In: CVPR (2018)
14. Feng, Y., Feng, Y., You, H., Zhao, X., Gao, Y.: Meshnet: Mesh neural network for 3d shape representation. AAAI 2019 (2018)
15. Furuya, T., Ohbuchi, R.: Ranking on cross-domain manifold for sketch-based 3d model retrieval. In: International Conference on Cyberworlds (2013)
16. Furuya, T., Ohbuchi, R.: Deep aggregation of local 3d geometric features for 3d model retrieval. In: BMVC (2016)
17. G, B., OE, G.: Riemannian adaptive optimization methods (2019)
18. Gabeur, V., Franco, J.S., Martin, X., Schmid, C., Rogez, G.: Moulding humans: Non-parametric 3d human shape estimation from single images. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
19. Gulcehre, C., Denil, M., Malinowski, N., Razavi, A., Pascanu, R., Hermann, K., Battaglia, P., Bapst, V., Raposo, D., Santoro, A., Freitas, N.d.: Hyperbolic neural networks. In: NeurIPS (2018)
20. Gulcehre, C., Denil, M., Malinowski, N., Razavi, A., Pascanu, R., Hermann, K., Battaglia, P., Bapst, V., Raposo, D., Santoro, A., Freitas, N.d.: Hyperbolic attention networks. In: ICLR (2019)
21. Han, Z., Lu, H., Liu, Z., Vong, C.M., Liua, Y.S., Zwicker, M., Han, J., Chen, C.P.: 3d2seqviews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation. IEEE Transactions on Image Processing **28**(8), 3986–3999 (2019)

22. Han, Z., Shang, M., Liu, Z., Vong, C.M., Liu, Y.S., Zwicker, M., Han, J., Chen, C.P.: Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention. IEEE Transactions on Image Processing **28**(2), 658–672 (2018)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
24. He, X., Huang, T., Bai, S., Bai, X.: View n-gram network for 3d object retrieval. In: ICCV (2019)
25. He, X., Zhou, Y., Zhou, Z., Bai, S., Bai, X.: Triplet-center loss for multi-view 3d object retrieval. In: CVPR (2018)
26. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR (2018)
27. Johns, E., Leutenegger, S., Davision, A.J.: Pairwise decomposition of image sequences for active multiview recognition. In: CVPR (2016)
28. Kanezaki, A., Matsushita, Y., Nishida, Y.: Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: CVPR (2018)
29. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3 d shape descriptors. In: Symposium on Geometry Processing. vol. 6, pp. 156–164 (2003)
30. Khrulkov, V., Mirvakhabova, L., Ustinova, E., Oseledets, I., Lempitsky, V.: Hyperbolic image embeddings. arXiv preprint arXiv:1904.02239 (2019)
31. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
32. Klokov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In: CVPR (2017)
33. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
34. Kumawat, S., Raman, S.: Lp-3dcnn: Unveiling local phase in 3d convolutional neural networks. In: CVPR (2019)
35. Leng, B., Zhang, C., Zhou, X., Xu, C., Xu, K.: Learning discriminative 3d shape representations by view discerning networks. IEEE transactions on visualization and computer graphics (2018)
36. Li, B., Lu, Y., Godil, A., Schreck, T., Aono, M., Johan, H., Saavedra, J.M., Tashiro, S.: SHREC13 track: large scale sketch-based 3D shape retrieval (2013)
37. Li, B., Lu, Y., Godil, A., Schreck, T., Bustos, B., Ferreira, A., Furuya, T., Fonseca, M.J., Johan, H., Matsuda, T., et al.: A comparison of methods for sketch-based 3d shape retrieval. CVIU **119**, 57–80 (2014)
38. Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., Burtscher, M., Fu, H., Furuya, T., Johan, H., et al.: Shrec14 track: Extended large scale sketch-based 3d shape retrieval. In: Eurographics Workshop on 3D Object Retrieval (2014)
39. Li, J., Chen, B., Hee, L.G.: So-net: Self-organizing network for point cloud analysis. In: CVPR (2018)
40. Liu, Y., Fan, B., Meng, G., Lu, J., Xiang, S., Pan, C.: Densepoint: Learning densely contextual representation for efficient point cloud processing. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
41. Mao, J., Wang, X., Li, H.: Interpolated convolutional networks for 3d point cloud understanding. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
42. Maturana, D., Scherer, S.: Multi-view harmonized bilinear network for 3d object recognition. In: IROS (2015)

43. Phong, B.T.: Illumination for computer generated pictures. Communications of the ACM **18**(6), 311–317 (1975)
44. Qi, A., Song, Y., Xiang, T.: Semantic embedding for sketch-based 3d shape retrieval. In: BMVC (2018)
45. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (2017)
46. Qi, C.R., Su, H., Niebner, M., Dai, A., Yan, M.: Volumetric and multi-view cnns for object classification on 3d data. In: CVPR (2016)
47. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017)
48. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
49. Sala, F., De Sa, C., Gu, A., R*é*, C.: Representation tradeoffs for hyperbolic embeddings. In: ICML (2019)
50. Sarkar, R.: Low distortion delaunay embedding of trees in hyperbolic plane. In: International Symposium on Graph Drawing (2011)
51. Shi, B., Bai, S., Zhou, Z., Bai, X.: Deeppano: Deep panoramic representation for 3-d shape recognition. IEEE Signal Processing Letters **22**(12), 2339–2343 (2015)
52. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The princeton shape benchmark. In: Shape Modeling Applications (2004)
53. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
54. Sousa, P., Fonseca, M.J.: Sketch-based retrieval of drawings using spatial proximity. Journal of Visual Languages & Computing **21**(2), 69–80 (2010)
55. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: ICCV (2015)
56. Su, J.C., Gadelha, M., Wang, R., Maji, S.: A deeper look at 3d shape classifiers. In: ECCV (2018)
57. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI (2017)
58. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
59. Tabia, H., Laga, H.: Learning shape retrieval from different modalities. Neurocomputing **253**, 24–33 (2017)
60. Tasse, F.P., Dodgson, N.: Shape2vec: semantic-based descriptors for 3d shapes, sketches and images. ACM Transactions on Graphics **35**(6), 208 (2016)
61. Tatsuma, A., Koyanagi, H., Aono, M.: A large-scale shape benchmark for 3d object retrieval: Toyohashi shape benchmark. In: Asia-Pacific Signal & Information Processing Association Annual Summit and Conference (2012)
62. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
63. Wang, C., Li, H., Zhao, D.: Preconditioning toeplitz-plus-diagonal linear systems using the sherman–morrison–woodbury formula. Journal of Computational and Applied Mathematics **309**, 312–319 (2017)
64. Wang, C., Li, H., Zhao, D.: Improved block preconditioners for linear systems arising from half-quadratic image restoration. Applied Mathematics and Computation **363**, 124614 (2019)

65. Wang, C., Pelillo, M., Siddiqi, K.: Dominant set clustering and pooling for multi-view 3d object recognition. In: BMVC (2017)
66. Wang, F., Kang, L., Li, Y.: Sketch-based 3d shape retrieval using convolutional neural networks. In: CVPR (2015)
67. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: CVPR (2015)
68. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: CVPR (2015)
69. X, W., R, G., A, G., K., H.: Non-local neural networks. In: CVPR (2018)
70. Xie, J., Dai, G., Zhu, F., Fang, Y.: Learning barycentric representations of 3d shapes for sketch-based 3d shape retrieval. In: CVPR (2017)
71. Xu, C., Li, Z., Qiu, Q., Leng, B., Jiang, J.: Enhancing 2d representation via adjacent views for 3d shape retrieval. In: ICCV (2019)
72. Xu, L., Sun, H., Liu, Y.: Learning with batch-wise optimal transport loss for 3d shape recognition. In: CVPR (2019)
73. Yang, Z., Wang, L.: Learning relationships for multi-view 3d object recognition. In: ICCV (2019)
74. Yasseen, Z., Verroust-Blondet, A., Nasri, A.: View selection for sketch-based 3d model retrieval using visual part shape description. The Visual Computer **33**(5), 565–583 (2017)
75. Yu, T., Meng, J., Yuan, J.: Multi-view harmonized bilinear network for 3d object recognition. In: CVPR (2018)