Cross-modal Scalable Hyperbolic Hierarchical Clustering

Teng Long
University of Amsterdam
uestc.longteng@gmail.com

Nanne van Noord
University of Amsterdam
n.j.e.vannoord@uva.nl

Abstract

Hierarchical clustering is a natural approach to discover ontologies from data. Yet, existing approaches are hampered by their inability to scale to large datasets and the discrete encoding of the hierarchy. We introduce scalable Hyperbolic Hierarchical Clustering (sHHC) which overcomes these limitations by learning continuous hierarchies in hyperbolic space. Our hierarchical clustering is of high quality and can be obtained in a fraction of the runtime.

Additionally, we demonstrate the strength of sHHC on a downstream cross-modal self-supervision task. By using the discovered hierarchies from sound and vision to construct continuous hierarchical pseudo-labels we can efficiently optimize a network for activity recognition and obtain competitive performance compared to recent self-supervised learning models. Our findings demonstrate the strength of Hyperbolic Hierarchical Clustering and its potential for Self-Supervised Learning.

1. Introduction

Concept hierarchies have been introduced for a variety of tasks including recognition [20, 7], retrieval [6], segmentation [30], fine-grained classification [15]. In many datasets, the hierarchy is manually defined [8] in terms of vision [11] or auditory [14] taxonomy. Those hierarchy definitions are subjective and may not suit domain-specific tasks. For example, vision hierarchy does not suit sound tasks and vice versa. In contrast to the abundance of applications of pre-defined hierarchies, models discovering hierarchy from data [39, 31] are scarce and outdated. In this paper, we strive for discovering the audio-visual hierarchy from video data in a self-supervised manner.

By learning the concept hierarchy we can, for example, use it as a pseudo-labeling scheme for training. Previous works on clustering-based self-supervised learning [9, 3, 42, 35, 27] have shown the benefits of assigning pseudo-labels to guide training. Continuous pseudo-labeling [25, 44, 10], in particular, draws attention as it can further improve self-supervision. Despite the appeal of continuous labels, hierarchical clustering models are naturally discrete [18], limiting their power to enhance self-supervision. Existing approaches for encoding discrete hierarchy into a continuous form have led to a noticeable information loss [38].

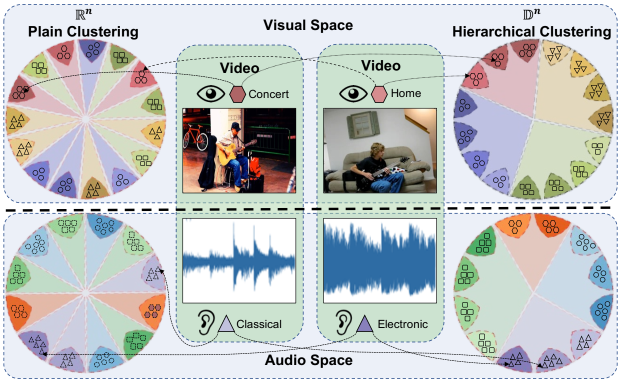


Figure 1. Example clustering result. Our model subdivides “playing guitar” for the visual modality into sub-categories *playing guitar in concert/at home* and for the audio modality into *playing classical/electronic guitar*. **(Left)** The plain clustering in Euclidean space $\mathbb{R}^n$ with randomly distributed clusters, no subordinate relationship between sub-categories, and no corresponding super-category. **(Right)** The proposed hierarchical clustering in hyperbolic space $\mathbb{D}^n$, similar clusters are grouped in the same fan-shaped zone, enabling us to subdivide sub-categories and aggregate super-categories.

However, hyperbolic geometry [37] allows lossless embedding of hierarchical trees [34, 21] into hyperbolic space $\mathbb{D}^n$. Previous work explored how hyperbolic geometry helps hierarchical clustering [33, 12]. Unfortunately, the application of these techniques to self-supervised learning has been limited by the computational complexity, they can hardly scale to datasets with $>\sim$20K data points. We solve this problem by breaking the hyperbolic clustering down into a two-stage process. The first stage, elaborated in Section 3.1, aims to generate K evenly distributed clusters. The second stage, described in Section 3.2 then aims to build a hierarchy from K clusters. This two-stage process enables us to efficiently train on large-scale video datasets, thereby enabling self-supervised learning with hierarchical pseudo-labels.

Our contributions are threefold: Firstly, we discover audio hierarchies and video hierarchies from data completely unsupervised. Secondly, we reduce the computational complexity of hyperbolic hierarchical clustering to scale, while maintaining the hierarchy quality. Thirdly, our model not only outperforms plain clustering-based competitors but also achieves faster convergence in a self-supervised setting.

2. Related Work

2.1. Cross-modal Self-supervision

Cross-modal learning aims to learn from the interaction between multiple modalities such as text, visual, audio, and more. In this paper, we focus on audio-visual cross-modal learning as they are naturally contained within videos.

There are two broad categories of cross-modal self-supervision methods for audio-visual data. 1) Constructing self-supervised tasks based on audio-video correspondence, and 2) Generating pseudo labels to (pre-)train the network. Our work falls in the second category, where we expand the notion of pseudo-labels to incorporate hierarchy.

Audio-video correspondence. Audio-visual synchronization is the most commonly seen audio-visual correspondence. For example, L^3 [4] constructs a binary classification task to learn whether the frame and audio come from the same clip, and the self-supervised task of [28] requires the correct classification of difficult samples and extremely difficult samples on the basis of consistent audio and video. Hu *et al.* [23] takes audio and video consistency as a self-supervised task, enabling localization based on audio and video activations on feature maps. Hassan *et al.* [2] proposed a Video-Audio-Text Transformer (VATT) that projects audio-video modalities into a common space to force their audio-visual correspondence. Although we are not training with audio-visual correspondence, we still use L^3 [4] pre-trained weights to initialize our model.

Pseudo-label Guided Training. Our research falls into the category of Pseudo-labeling methods [3, 5, 24, 1, 13, 41], which generate pseudo-labels in an unsupervised manner and subsequently guides training based on these labels. Pseudo-labeling is commonly used in self-supervised learning for audio-visual data. Alwassel *et al.* [3] uses both visual and audio clustering as pseudo-labels to supervise the other modality, thereby introducing cross-modal information for downstream tasks. Asano *et al.* [5] exploits the pseudo-labels from both sound and vision for self-supervision. Francisco *et al.* [41] proposed multi-modal vision teachers, each pre-trained modality-specific teacher predicts bounding boxes that are then used as pseudo-labels for training the audio-student network. Triantafyllos *et al.* [1] generate pseudo bounding boxes with pseudo labels to achieve object detection and sound-source localization. Chen *et al.* [13] fuses clustering results from each modality to get a multi-modal pseudo label for guiding training, achieving zero-shot video retrieval and localization.

This paper is related to SeLaVi [5] as they also perform clustering-based self-supervision, although based on an equal-sized cluster assumption. Similarly, we also perform equal-sized subdivisions as the pre-clustering stage of our model, but we differ as we allow a varying size and a flexible number of clusters K by combining clusters based on the learned hierarchy.

2.2. Hierarchical Clustering

Clustering is closely related to pseudo-label generation. In self-supervised learning, plain clustering is frequently used to generate the pseudo-labels while hierarchical clustering remains under-utilized.

Plain clustering. Many self-supervised learning methods [9, 25, 44, 10, 43] can be seen as a form of clustering, where the model learns to group similar examples together. DeepCluster [9] uses K-means clustering to define pseudo-labels, Self Labeling [44] uses, and Sinkhorn-Knoop clustering [17] is also widely used. SwAV [10] uses online clustering updates and soft clustering.

Discrete Hierarchical Clustering. Despite the different types of clustering explored for self-supervised learning, hierarchical clustering [22] remains unexplored for guiding self-supervision.

Hyperbolic Hierarchical Clustering. In many audio-visual datasets [8, 11, 4, 14], human experts define the hierarchy of categories, which benefits computer vision tasks [29, 15, 30]. Given that those pre-defined hierarchies are subjectively defined, they either lean to the sound side [14] or semantic side [11], it would be valuable to discover the natural hierarchy inside data, leading to hierarchical clustering (HC).

HC has been performed heuristically, without an objective evaluation and optimization metric, until a discrete objective function was proposed by Dasgupta [18]. This objective function enables gradient-based hierarchical clustering [33], by optimizing the Dasgupta Cost given to sample embeddings in a hyperbolic space. HypHC [12] further reduces the requirement for pre-defined embeddings by initializing all embeddings randomly.

Our method is closely related to HypHC [12], we follow their sampling strategy to generate triplets for adjusting embeddings. But we differ with HypHC in two ways; HypHC has a complexity of $\mathcal{O}(N^3)$ for N data points, which does not scale, and HypHC only runs one pass without integrating representation learning.

3. Scalable Hyperbolic Hierarchical Clustering

Given video clip v, passing it through the visual encoder $\phi_v(\cdot)$ gives us the visual feature $\mathbf{v} = \phi_v(v)$. Similarly, for audio encoder $\phi_a(\cdot)$ we obtain audio feature $\mathbf{a} = \phi_a(v)$. Performing scalable Hyperbolic Hierarchical Clustering (sHHC) on visual features $\{\mathbf{v}\}$ and audio features $\{\mathbf{a}\}$ yields visual hierarchy $\mathbb{T}_v$ and auditory hierarchy $\mathbb{T}_a$, each in a continuous form in hyperbolic space $\mathbb{D}^n$. To reduce the computational complexity of building hierarchical clusters on hyperbolic space, sHHC breaks the clustering process into two parts. 1) Pre-clustering, described in Section 3.1, generates evenly distributed pre-clusters. 2) Post-clustering, described in Section 3.2, uses the pre-clusters to construct the hierarchy. Then the nodes on the visual hierarchy $\hat{y}_v \in \mathbb{T}_v$ and audio hierarchy $\hat{y}_a \in \mathbb{T}_a$ act as pseudo-labels to supervise the network in a cross-modal manner. Our training target is to predict the continuous pseudo-labels in hyperbolic space $\mathbb{D}^n$, *i.e.*, we have four heads that maps each modality to audio and vision pseudo-labels.

$$\begin{array}{l} h_{v\to a}(\mathbf{v}) \approx \hat{y}_a \\ h_{a\to v}(\mathbf{a}) \approx \hat{y}_v \\ h_{v\to v}(\mathbf{v}) \approx \hat{y}_v \\ h_{a\to a}(\mathbf{a}) \approx \hat{y}_a \end{array}, \tag{1}$$

as elaborated in Section 3.3. The overall structure is shown in Figure 2.

3.1. Sinkorn Pre-clustering

For a given classification head, its output $\hat{y} = p(y|\mathbf{x})$ gives the probability that sample $\mathbf{x}$ belongs to class y, leading to the cross-entropy loss over dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1\cdots N}$:

$$L(\mathcal{D}) = -\sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i). \tag{2}$$

In a clustering setting, we only have the feature set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1\cdots N}$, while the label y_i is not available. Instead, we regard the cluster assignment $q(y_i|x_i) \in \{0, 1\}$ as a surrogate of label y_i (*i.e.*, the pseudo-label). Equation (2) is then modified to:

$$L(\mathcal{X}) = -\sum_{i=1}^{N}\sum_{j=1}^{K} q\left(y_i = j \mid \mathbf{x}_i\right) \log p\left(y_i = j \mid \mathbf{x}_i\right) \tag{3}$$

To avoid model collapse it is important for the cluster assignment $q(y|\mathbf{x}_i)$ to be evenly distributed [9] among all classes, leading to the constraint that the assignment of N data points must be partitioned into K equally-sized subsets.

$$\sum_{i=1}^{N} q\left(y = j \mid \mathbf{x}_i\right) = \frac{N}{K}, \quad j = 1, 2, \cdots, K. \tag{4}$$

As shown by Asano *et al.* [44], this problem can be solved efficiently in $\mathcal{O}(NK)$ using Sinkhorn fixed point iteration [17] to update $p(y_i = j \mid x_i)$:

$$\mathbf{u} = \frac{1}{K} \odot \frac{1}{\mathbf{P}^{\lambda}\mathbf{v}} \tag{5}$$

$$\mathbf{v} = \frac{1}{N} \odot \frac{1}{(\mathbf{u}^T\mathbf{P}^{\lambda})^T} \tag{6}$$

where $\odot$ is the element-wise product, $\mathbf{P} = [p_{ij}]$ is the matrix format predicted posterior probability where each element $p_{ij} = p(y_i = j \mid x_i)$.

3.2. Hyperbolic Post-clustering

Preliminary. Hyperbolic geometry is locally isomorphic to Euclidean space but has negative curvature in general, and thus is able to represent tree structures with arbitrary low distortion [38]. There are various models that satisfy the definition of hyperbolic space, in this paper, for the convenience of visualization, we use the Poincaré disk model, defined as:

$$\mathbb{D}_c^n = \{\mathbf{x} \in \mathbb{R}^d \mid c\,\|\mathbf{x}\| \le 1\}, \tag{7}$$

where n is the dimensionality of the space and c is the curvature of the hyperbolic space. Moderately set curvature enables the tree's low-distortion embedding. We use $c = 1$ in this paper, without causing ambiguity, we use $\mathbb{D}$ for $\mathbb{D}_c^n$ to simplify the notation. The distance between any two points $\mathbf{x}$ and $\mathbf{y}$ in Poincaré space is defined as:

$$d_c(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}} \operatorname{arctanh}\left(\sqrt{c}\,\|{-\mathbf{y}} \oplus_c \mathbf{x}\|\right), \tag{8}$$

where $\oplus_c$ is the Möbius addition [40] in $\mathbb{D}$, *i.e.*:

$$\mathbf{x} \oplus_c \mathbf{y} = \frac{\left(1 + 2c\langle \mathbf{x}, \mathbf{y}\rangle + c\|\mathbf{y}\|^2\right)\mathbf{x} + \left(1 - c\|\mathbf{x}\|^2\right)\mathbf{y}}{1 + 2c\langle \mathbf{x}, \mathbf{y}\rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}. \tag{9}$$

Hyperbolic Clustering. Constructs a continuous hierarchy $\mathbb{T}$ from data, we propose a continuous loss function and optimize it. Similar to Chami *et al.* [12], which positions embeddings into the hyperbolic space according to the similarity of data samples, we construct $\mathbb{T}$ using the pre-cluster centroids to calculate the pair-wise similarities among centroids, then the similarities are used to learn the continuous hierarchy. The learning objective is to minimize the clustering loss:

$$\min_{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n} \sum_{ijk} \left[\mathbf{1}^T \mathbf{w}_{ijk} - \sigma \begin{pmatrix} d_c(\mathbf{0}, \mathbf{z}_{i\vee j}) \\ d_c(\mathbf{0}, \mathbf{z}_{i\vee k}) \\ d_c(\mathbf{0}, \mathbf{z}_{j\vee k}) \end{pmatrix}^T \mathbf{w}_{ijk} \right], \tag{10}$$

where $\mathbf{w}_{ijk} = [w_{ij}, w_{ik}, w_{jk}]^T \in \mathbb{R}^3$ is the vector of similarities between sample (i, j), (i, k) and (j, k), $\mathbf{z}_i$ is the

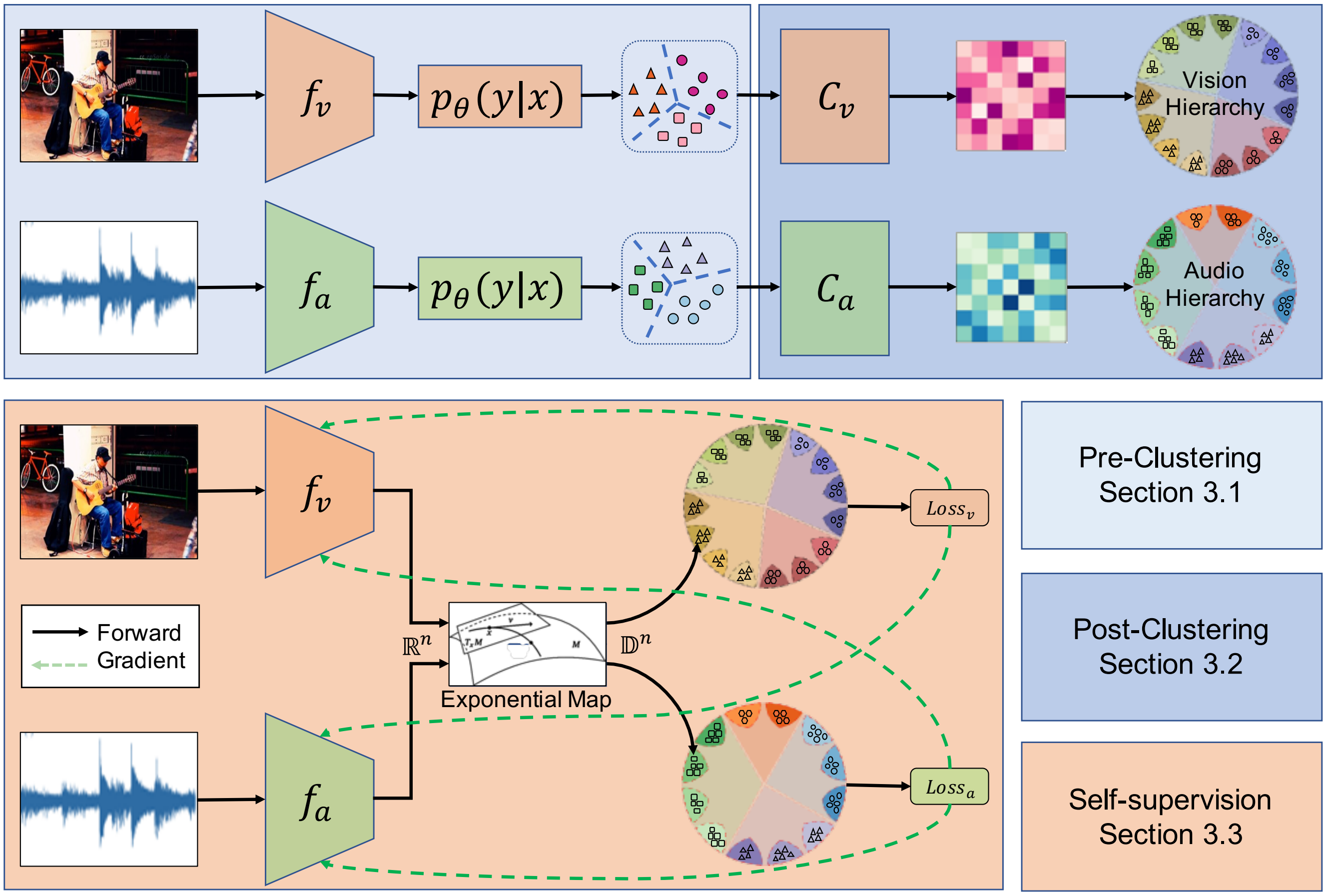


Figure 2. **Scalable Hyperbolic Hierarchical Clustering.** The dataset is first passed through the audio-visual encoders to get audio features and visual features. **(Step 1).** Features are divided into equal-sized subsets as optimal transport assignments. **(Step 2).** Subsets are then embedded hierarchically to hyperbolic space $\mathbb{D}^n$ based on similarities. **(Step 3).** The resultant hierarchy further guides the encoder training, the gradient passes through the network in a cross-modal manner. The Exponential Map translates data points from $\mathbb{R}^n$ to $\mathbb{D}^n$, solving the inconsistency between spaces.

hyperbolic embedding of $i-th$ sample, σ is the softmax function and $\mathbf{z}_{i\vee j}$ is the hyperbolic embedding of the LCA (lowest common ancestor) of $i-th$ sample and $j-th$ sample.

Minimizing Equation (10) forces similar pairs to have an LCA far from the origin and dissimilar pairs to have an LCA near the origin. For the triplet (i, j, k) the pair (i, j) is more similar than (i, k) and (j, k), as such the learning must force $d_c(\mathbf{0}, \mathbf{z}_{i\vee j}) > d_c(\mathbf{0}, \mathbf{z}_{i\vee k})$ and $d_c(\mathbf{0}, \mathbf{z}_{i\vee j}) > d_c(\mathbf{0}, \mathbf{z}_{j\vee k})$ to minimize Equation (10).

Exponential Initialization. Optimizing this objective requires sampling triplets from a space that is $\mathcal{O}(N^3)$ large, which can hardly be traversed for a large dataset like Kinetics400 [11]. Even when we reduce the sample space size to $\mathcal{O}(K^3)$ with pre-clustering, the sample space is still 10^9 large when we use $K = 10^3$ in our experiments. Hence, good initialization of the embeddings is vital to the model performance. Different to [12], which uses random initialization for all samples, we use the exponential map $\mathbb{E}(\cdot)$ to project the audio-visual features from $\mathbb{R}^n$ onto the Poincaré disk $\mathbb{D}$, with respect to tangent point $\mathbf{x}$:

$$\mathbb{E}_{\mathbf{x}}(\mathbf{v}) = \mathbf{x} \oplus_c \left(\tanh \left(\frac{\sqrt{c}\|\mathbf{v}\|}{1 - \|\mathbf{x}\|^2} \right) \frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|} \right), \tag{11}$$

so without loss of generality, we set $\mathbf{x} = 0$ as the tangent point for all the projections to $\mathbb{D}$, leading to a simplified notation:

$$\mathbb{E}(\mathbf{v}) = \tanh \left(\sqrt{c}\|\mathbf{v}\| \right) \frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}, \tag{12}$$

3.3. Cross-modal Hierarchical Supervision

After obtaining the visual hierarchy $\mathbb{T}_v$ and audio hierarchy $\mathbb{T}_a$ we use them for cross-modal pseudo-label supervision. Similar to SwAV [10], the pseudo-labels we use are continuous, but we shift to hyperbolic representation as labels in hyperbolic are naturally continuous [34], and our pseudo-labels are hierarchical.

We use the negative distance between sample embeddings and the cluster embeddings as the probability for logits, akin to [32]:

$$p(y=k|\mathbf{x}) = \frac{\exp\left(-d_c(\psi(y), \mathbb{E}(\mathbf{v})\right)}{\sum_{k'} \exp\left(-d_c(\psi(y), \mathbb{E}(\mathbf{v})\right)}, \tag{13}$$

where $\psi(y) \in \mathbb{D}^n$ is the hyperbolic embedding of category label y.

Cluster Merging Note that each leaf nodes act as one class template during hierarchical supervision, sometimes it can be the case that data within one cluster be forced to divide as we deploy overclustering [25]. One advantage of hierarchical clustering is that we are able to combine clusters easily by shifting to the parent node.

4. Experimental Setup

4.1. Dataset

For the experiments we use two datasets, VGGSound [14] and KineticsSound [4], which have been used in prior research on audio-visual learning [4, 5, 2]. We pretrain on the VGGSound dataset and fine-tune on the KineticsSound dataset. The statistics of both datasets are shown in Table 1.

Notably, both VGGSound and KineticsSound datasets provide hierarchy information. VGGSound's hierarchy divides 309 categories into 16 parent categories based on audio information, such as animal sounds, human sounds, natural sounds, musical instruments, and so on. KineticsSound divides 34 categories into 5 parent categories including Social Activities, Household Activities, Sports, and Entertainment based on visual and semantic information. We cannot use the pre-defined hierarchy for cross-modal learning because in both datasets the hierarchy definition only concerns one modality and the hierarchy regarding the other modality is missing.

Note that although VGGSound and KineticsSound datasets have similar categories, *i.e.* 13 of the 34 categories in Kinetics are also found in VGGSound (10 of these being musical instrument categories), we argue that this kind of overlap will not leak label information, because our pretraining is completely self-supervised.

4.2. Implementation Details

Audio-Visual Sample Alignment. For one video let n_v be the number of frames, f_v the frame sample rate in terms of *(fps) frame per second*, n_a the number of audio samples, and f_a the sample rate in terms of *Hz*. In such cases, we usually have $n_v = t \times f_v$. However, this is not the case for the audio signal, which means $n_a \neq t \times f_a$, because the audio signal and visual signal are encoded into different streams that require alignment. We recover the audio-visual alignment using the *PTS (Presentation Time Stamp)* encoded in the video file.

Table 1. Dataset statistics for VGGSound and KineticsSound.

Dataset	#samples	split	#cls	hierarchy
VGGSound	199276	pretrain	309	👂
KineticsSound	23845	train	34	👁
	1652	eval	34	👁

Audio Preprocessing. The audio signal comes with different sampling rates, mainly *48000hz* and *44100hz*, to make those sample rates consistent and generate fair comparison, we re-sample the audios to *24000hz* akin to the implementation of SeLaVi [5]. Similarly, we transform the raw audio waveform to its Logarithmic Mel-filterbank Energy [19] representation. i.e., from $a \in \mathbb{R}^{t\times 24000}$ to $\mathbf{a} \in \mathbb{R}^{t\times 257\times 99}$.

Multi-head clustering. Similar to [44] and [5], we use multiple clustering heads as it boosts performance. For fair comparison, we set $H = 10$ during self-supervision.

Problem Simplification. Limited by computing resources, we simplify the vision part to the same setting as [4], that is, we take one frame per clip to represent the visual information of the entire clip. In order to achieve a fair comparison, we re-run all the baseline algorithms under this simplified setting.

Fair comparison. To further ensure a fair comparison, all baseline methods were trained with the same backbone, which is the L^3 [4] network. All experiments were done with the same input and same data augmentation, and we used the same optimization tricks, thus resulting in fair comparison among the models.

Warm up. We pre-train the L^3 [4] network and then use these pre-trained weights to warm up the backbone. This prevents the clustering algorithm from generating meaningless clusters based on purely random initialization.

5. Results

5.1. Clustering Quality

Baseline models. To show the superiority of our model we compare it against representative clustering-based methods. In particular, **DeepCluster** [9] is the most representative clustering-based method, followed by **SeLa** [44] further extend the method with Sinkhorn clustering, and **SeLaVi** [5] which also considers cross-modal learning on audio-visual data.

The results of the clustering quality experiments are shown in Table 2. Among all the competitors, we can perform best across all clustering quality metrics, even at a lower number of clusters. Whereas more clusters may lead to better clustering performance, our method performs even better when having fewer #clusters. This is surprising, as over-clustering is generally easier, having fewer clusters

Table 2. The clustering quality on Kinetics-Sound dataset under different numbers of clusters. For NMI$\in [0,1]$ the higher the better, and ARI$\in [0,1]$ the higher the better. The numbers are in percentages. We also examine the setting when using fewer clusters.

Model	👁	👂	NMI↑	ARI↑	#cluster
Deep Cluster [9]	✓	✗	12.5	4.3	100
	✗	✓	11.3	4.0	100
XDC [3]	✓	✓	12.9	4.8	100
SeLa [44]	✓	✗	34.8	19.5	100
	✗	✓	33.4	18.7	100
SeLaVi [5]	✓	✓	36.4	20.2	100
Ours	✓	✗	35.2	20.1	100
	✗	✓	33.8	19.0	100
	✓	✓	36.9	20.4	100
Ours	✓	✓	37.2	**20.8**	60
	✓	✓	**37.4**	20.4	34

implies that the learned clusters are more informative than those of other methods. For instance, for the 34 annotated classes (semantic clusters) in KineticsSound, if we can learn proper clusters then the number of learned clusters should approach the number of annotated classes, resulting in a good performance.

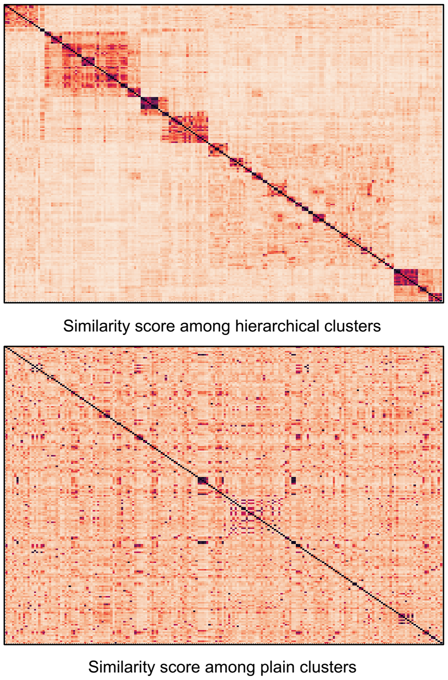
We also visualize the effect of hierarchical clusters in terms of cluster similarity in Figure 3. We achieve the visualization by re-ordering the clusters in a hierarchical-wise way, then we notice that, when considering the hierarchical structure among clusters, the clusters show the agglomeration effect, similar clusters tend to have a higher similarity, while in the non-hierarchical counterparts, we can hardly discover any agglomeration and all the clusters are just randomly distributed.

5.2. Hierarchy Discovery

One of the main contributions of our work is to establish the class hierarchy within the data. Based on our experiments we quantitatively show that our proposed method constructs the hierarchy well in terms of Dasgupta Cost [18] and running speed, as shown in Table 3.

Discrete Baseline models. For classical hierarchical clustering an important difference is the cluster merging technique, here we include two representative discrete alternatives of agglomerative clustering [22]: **Complete Link** that consider the nearest point pairs when combining clusters and **Single Link** that measure the furthest point pairs when combining clusters.

Continuous Baseline models. We also compared against continuous hierarchical clustering techniques that are recently developed. **UFit** [16] defines the dendrogram as an ultra-metric, then defines a relaxation of Dasgupta Cost for optimization to find the optimal ultra-metric. **HypHC** [12] also optimize a relaxed version of Dasgupta Cost, and it optimizes the embeddings of the hierarchy on $\mathbb{D}$, instead of the hierarchy $\mathbb{T}$ itself.

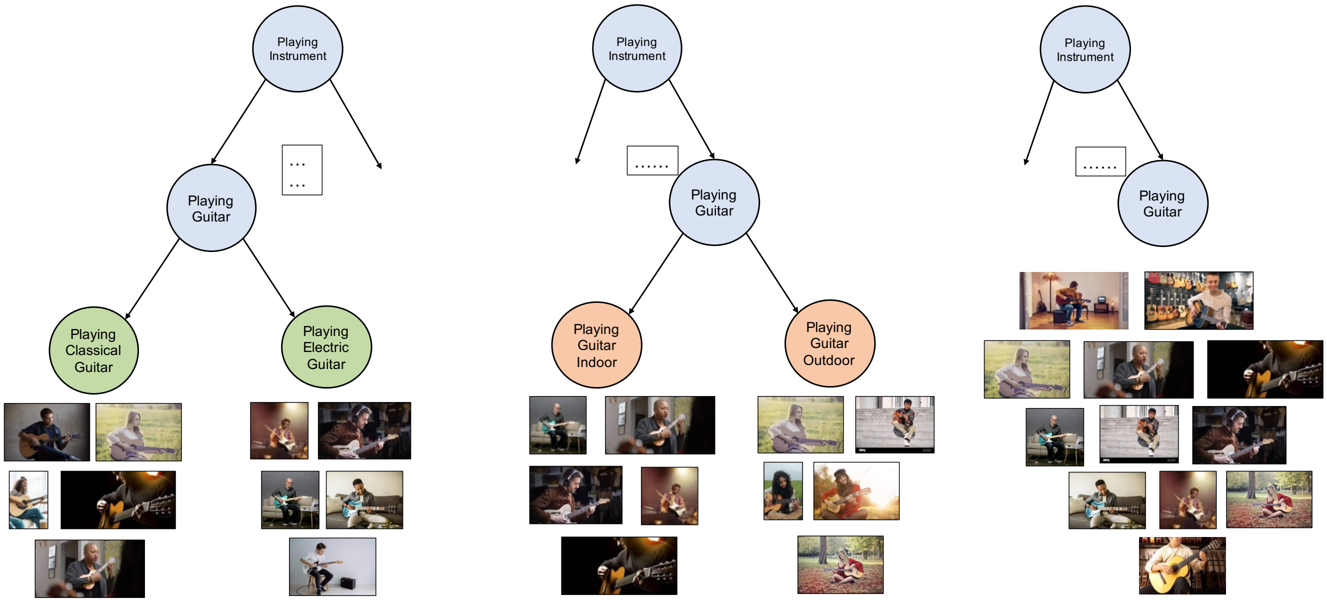

Similarity score among hierarchical clusters

Similarity score among plain clusters

Figure 3. Visualization of the similarity scores between clusters for **(Top)** hierarchical clusters and **(Bottom)** plain clusters. The hierarchical clusters have a well-defined structure whereas the plain clusters appear nearly arbitrary. We construct the hierarchy in hyperbolic space based on the similarities between the hierarchical clusters.

We qualitatively show the hierarchy discovered in Figure 4. Based on this visualization we can observe that we are able to discover fine-grained categories, subdividing semantic categories into narrow categories based on modality-specific information. This shows the potential of our approach to discover task-specific fine-grained hierarchies. For instance, for the audio hierarchy we are able to discover different sub-categories given the same appearance, *e.g.* electrical guitar, classical guitar, and acoustic guitar all belong to the same class "playing guitar". Fine-grained hierarchies are beneficial as previous self-supervision methods, including DeepCluster [9] and SeLa [44], have shown that over-segmentation benefits the training process, especially when revealing sub-categories structures that are not manually annotated. Crucially, obtaining a similar level of fine-grained annotation through human annotation would be very costly and time intensive.

We are able to achieve the lowest cost among all the hierarchical clustering methods. In particular, we are able to outperform the discrete baselines because we are updating the position of embeddings while for discrete baselines the



(a) Discovered Audio Hierarchy (b) Discovered Visual Hierarchy (c) Human Annotation

Figure 4. Visualisation of discovered hierarchies within the VGGSound dataset. Based on this visualization we can observe that we discover fine-grained hierarchies which are modality-specific. **(Left)** The audio hierarchy, where different guitar types are discovered for guitar playing. **(Middle)** The visual hierarchy, where different scenarios for guitar playing are discovered. **(Right)** Human annotation, which consists of only two levels of hierarchy, it would require significant extra expense and effort to further subdivide these into fine-grained classes.

Table 3. The hierarchy quality of our proposed model expressed in terms of Dasgupta Cost. The unit is multiplied by 10^{14}, and the running time is calculated after convergence. Methods with a † failed to converge due to computational cost.

Hierarchical Model	Dasgupta Cost↓	Running Time↓
Complete Link	8.01	∼10min
Single Link	6.69	∼10min
UFit [16]	7.72	∼4h
HypHC [12]†	7.66	>4h
Ours	**6.62**	**∼2min**

embeddings do not move anymore once the data similarity is computed. We also beat the continuous method UFit by a large margin. It is not surprising that HypHC outperforms UFit, as UFit's optimization is not directly related to Dasgupta Cost. However, our method, which can be regarded as an approximated version of HypHC, still outperforms both methods at a much faster running speed.

5.3. Downstream Task

We tested the pre-trained backbone on the downstream task of activity recognition. We tested the models on the KineticsSound dataset in three different settings: audio recognition, vision recognition, and multi-modal recognition. The results are illustrated in Table 4.

Table 4. The downstream activity recognition performance on the KineticsSound dataset. We separately evaluate the performance in a audio-only, visual-only, and audio-visual setting. Only L^3 and our method are able to operate across all three settings, with our method showing superior performance.

	👁	👂	Both
Deep Cluster [9]	69.68	-	-
Deep Cluster [9]	-	49.92	-
SeLa [44]	71.53	-	-
L^3 [4]	68.76	51.13	70.95
XDC [3]	-	-	71.27
SelaVi [5]	-	-	73.51
GDT [36]	-	-	73.85
Ours	71.69	51.90	74.10

Baseline methods. We compare against the four baselines that are most related to us. L^3 [4] is the first paper regarding audio-visual self-supervision, we adopted their input simplification strategy. **DeepCluster** [9] introduced clustering-based pseudo-label guided self-supervision. **XDC** [3] uses audio-visual cross-modal pseudo-labels for self-supervision, our method also follows this strategy. **Self-labelling** [44] for uni-modality and **SeLaVi** [5] for audio-visual fusion uses Sinkhorn clustering [17].

Only two of the methods are able to perform activity recognition across all modality settings, L^3 and our method. Notably, in all three settings, we improve over L^3. Additionally, even when compared to more specialized methods which only operate in a single setting we still outperform them by a small margin. This demonstrates the strength and flexibility of our approach, as we are able to operate across all three settings, and perform well in all of them.

5.4. Evaluation Metrics

Clustering Quality. Metrics such as normalized mutual information (NMI), adjusted rand index (ARI), and mean purity (denoted as p_{mean}) are commonly used to evaluate clustering quality. **NMI** measures the similarity between the clustering results and the ground truth labels, where a value of 1 indicates a perfect match and 0 indicates no similarity. **ARI**, on the other hand, measures the agreement between two sets of clustering results, adjusted for chance agreement, with a value of 1 indicating perfect agreement and 0 indicating agreement no better than chance. **Mean purity** is a measure of cluster quality in clustering analysis. It measures how pure each cluster is, based on the distribution of ground truth labels among the data points in each cluster. Its value ranges from 0 to 1, with higher values indicating that the clusters are purer and that most of the data points in each cluster belong to the same ground truth label. We refer to the formal definition of these metrics in Supplementary Materials.

Hierarchy Quality. The Dasgupta cost [18] measures the total cost of the clustering solution, where the cost is defined as the sum of the distances between each point and its closest cluster center.

5.5. Effect of Hyperbolic space

Hyperbolic space is well-suited for hierarchies because it can embed a tree without information loss [34], while for Euclidean space there is a loss even in the infinite dimensional case. All prior methods in Table 2 perform clustering in Euclidean space, to verify the benefits of hyperbolic we add ablations in Table 5 for different manifolds.

Table 5. Effect of using different manifold $\mathcal{M}$ (simplex Δ, Euclidean $\mathbb{R}$, and hyperbolic $\mathbb{D}$) on Kinetics-Sound for multi-modal setting. The * and + indicate that the method was modified to $\mathbb{R}$ or $\mathbb{D}$ space respectively for self-supervision.

Model	$\mathcal{M}$	Dimensionality	NMI↑	ARI↑
SeLaVi [5]	Δ	100	36.4	20.2
SeLaVi* [5]	$\mathbb{R}$	512	36.4	20.1
SeLaVi+ [5]	$\mathbb{D}$	512	36.1	19.8
Ours*	$\mathbb{R}$	512	36.3	20.0
Ours	$\mathbb{D}$	50	**36.9**	**20.4**

We construct three extra baselines: SeLaVi* on Euclidean space $\mathbb{R}$, SelaVi+ on hyperbolic space $\mathbb{D}$, and Ours* on $\mathbb{R}$. We can observe that Euclidean indeed requires higher dimensionality and that the benefits of hyperbolic disappear when not using hierarchy (i.e., SeLaVi+).

5.6. Ablation Study

As prior hierarchical clustering methods are limited by their lack of scalability, we propose to overcome this issue by dividing the clustering process into three steps: Pre-clustering (3.1), Post-clustering (3.2), and SSL (3.3), where the final clustering is only obtained after the SSL step. To demonstrate the effectiveness of this three step process, we perform an ablation in Table 6. For these results, SelaVi [5] can be regarded as using only 3.1 (Pre-clustering), and HypHC [12] (equipped with hierarchy) as only using 3.2 (Post-clustering). When using neither pre-clustering nor post-clustering the network is a L^3 initialized network. For only post-clustering (row 2), the large number of samples inhibits convergence, leading to a deteriorated performance that is even lower than L^3. When only pre-clustering (row 3) the performance is good, but the model is incapable of discovering hierarchy. Our approach allows us to obtain good performance while also discovering hierarchy.

Table 6. Ablation results on Kinetics-Sound for multi-modal setting. ⁘ indicates Pre-clustering, ⁂ indicates Post-clustering.

Model	⁘	⁂	NMI↑	ARI↑
L^3	✗	✗	26.5	9.8
HypHC [12]	✗	✓	20.6	6.7
SeLaVi [5]	✓	✗	36.4	20.2
Ours	✓	✓	**36.9**	**20.4**

6. Conclusion

Limitations. Similar to Sinkhorn [17], Hyperbolic hierarchical clustering usually requires high numerical precision, as the data points projected to the edge of $\mathbb{D}^n$ tends to have an infinite far distance, leading to performance loss when we use half-precision training or mixed precision training. We suggest using double precision in the hyperbolic distance calculation, which only leads to a marginal increase in memory consumption. Hierarchical hyperbolic clustering optimization runs slower than Sinkhorn clustering and FAISS [26] Kmeans, but it is still applicable in self-supervised context as 1) the running time is very short compared to the time it takes to compute features for the training set and 2) we only have to redo the clustering after a few epochs.

In this paper we proposed a new method for hierarchical clustering, overcoming the limitations of existing methods related to scalability and lack of continuous representation. Our findings show that we can obtain high-quality clusters in a fraction of the runtime, and qualitatively we show that we are able to discover semantically coherent hierarchies for multiple modalities. We evaluate the hierarchies on a cross-modal self-supervised task and obtain competitive performance with prior methods, demonstrating the strength of hyperbolic hierarchical representations obtained by our proposed clustering method. We see hierarchical pseudo-labels in continuous hyperbolic space as a natural extension of discrete pseudo-labels and we expect that this approach will be able to benefit a wide range of tasks.

Acknowledgements

The research reported in this paper is performed as part of the VisXP project, supported by ClickNL for the top sector creative industry.

References

[1] Triantafyllos Afouras, Yuki M. Asano, Francois Fagan, Andrea Vedaldi, and Florian Metze. Self-supervised object detection from audio-visual correspondence. In *CVPR*, 2022. 2

[2] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. VATT: Transformers for Multimodal Self-Supervised Learning from Raw Video, Audio and Text. In *NeurIPS*, volume 34, 2021. 2, 5

[3] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-Supervised Learning by Cross-Modal Audio-Video Clustering. In *NeurIPS*, 2020. 1, 2, 6, 7

[4] Relja Arandjelovic and Andrew Zisserman. Look, Listen and Learn. In *ICCV*, 2017. 2, 5, 7

[5] Yuki Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*, 2020. 2, 5, 6, 7, 8

[6] Björn Barz and Joachim Denzler. Hierarchy-Based Image Embeddings for Semantic Image Retrieval. In *WACV*, 2019. 1

[7] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A. Lord. Making Better Mistakes: Leveraging Class Hierarchies With Deep Networks. In *CVPR*, 2020. 1

[8] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *CVPR*, 2015. 1, 2

[9] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV*, 2018. 1, 2, 3, 5, 6, 7

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*, 2020. 1, 2, 4

[11] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*, 2017. 1, 2, 4

[12] Ines Chami, Albert Gu, Vaggos Chatziafratis, and Christopher Ré. From Trees to Continuous Embeddings and Back: Hyperbolic Hierarchical Clustering. In *NeurIPS*, 2020. 1, 2, 3, 4, 6, 7, 8

[13] Brian Chen, Andrew Rouditchenko, Kevin Duarte, Hilde Kuehne, Samuel Thomas, Angie Boggust, Rameswar Panda, Brian Kingsbury, Rogerio Feris, David Harwath, James Glass, Michael Picheny, and Shih-Fu Chang. Multimodal Clustering Networks for Self-Supervised Learning From Unlabeled Videos. In *ICCV*, 2021. 2

[14] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A Large-Scale Audio-Visual Dataset. In *ICASSP*, 2020. 1, 2, 5

[15] Tianshui Chen, Wenxi Wu, Yuefang Gao, Le Dong, Xiaonan Luo, and Liang Lin. Fine-Grained Representation Learning and Recognition by Exploiting Hierarchical Semantic Embedding. In *ACM Multimedia*, 2018. 1, 2

[16] Giovanni Chierchia and Benjamin Perret. Ultrametric Fitting by Gradient Descent. In *NeurIPS*, 2019. 6, 7

[17] Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *NeurIPS*, 2013. 2, 3, 8

[18] Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In *STOC*, 2016. 1, 2, 6, 8

[19] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28, 1980. 5

[20] Ankit Dhall, Anastasia Makarova, Octavian Ganea, Dario Pavllo, Michael Greeff, and Andreas Krause. Hierarchical Image Classification Using Entailment Cone Embeddings. In *CVPR Workshops*, 2020. 1

[21] Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. In *ICML*, 2018. 1

[22] Trevor Hastie, Jerome Friedman, and Robert Tibshirani. Unsupervised Learning. In *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001. 2, 6

[23] Di Hu, Rui Qian, Minyue Jiang, Xiao Tan, Shilei Wen, Errui Ding, Weiyao Lin, and Dejing Dou. Discriminative Sounding Objects Localization via Self-supervised Audiovisual Matching. In *NeurIPS*, volume 33, 2020. 2

[24] Vincent Tao Hu, David W. Zhang, Yuki M. Asano, Gertjan J. Burghouts, and Cees G. M. Snoek. Self-guided diffusion models. In *CVPR*, 2023. 2

[25] Xu Ji, Andrea Vedaldi, and Joao Henriques. Invariant Information Clustering for Unsupervised Image Classification and Segmentation. In *ICCV*, 2019. 1, 2, 5

[26] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7, 2021. 8

[27] Salar Hosseini Khorasgani, Yuxuan Chen, and Florian Shkurti. SLIC: Self-Supervised Learning With Iterative Clustering for Human Action Videos. In *CVPR*, 2022. 1

[28] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative Learning of Audio and Video Models from Self-Supervised Synchronization. In *NeurIPS*, volume 31, 2018. 2

[29] Suren Kumar and Rui Zheng. Hierarchical Category Detector for Clothing Recognition From Visual Data. In *ICCV Workshops*, 2017. 2

[30] Liulei Li, Tianfei Zhou, Wenguan Wang, Jianwu Li, and Yi Yang. Deep Hierarchical Semantic Segmentation. In *CVPR*, 2022. 1, 2

[31] Li-Jia Li, Chong Wang, Yongwhan Lim, David M. Blei, and Li Fei-Fei. Building and using a semantivisual image hierarchy. In *CVPR*, 2010. 1

[32] Teng Long, Pascal Mettes, Heng Tao Shen, and Cees GM Snoek. Searching for actions on the hyperbole. In *CVPR*, 2020. 5

[33] Nicholas Monath, Manzil Zaheer, Daniel Silva, Andrew McCallum, and Amr Ahmed. Gradient-based Hierarchical Clustering using Continuous Representations of Trees in Hyperbolic Space. In *SIGKDD*, 2019. 1, 2

[34] Maximillian Nickel and Douwe Kiela. Poincaré Embeddings for Learning Hierarchical Representations. In *NeurIPS*, 2017. 1, 4

[35] Sungwon Park, Sungwon Han, Sundong Kim, Danu Kim, Sungkyu Park, Seunghoon Hong, and Meeyoung Cha. Improving Unsupervised Image Clustering With Robust Learning. In *CVPR*, 2021. 1

[36] Mandela Patrick, Yuki M. Asano, Polina Kuznetsova, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. On Compositions of Transformations in Contrastive Self-Supervised Learning. In *ICCV*, 2021. 7

[37] Joel W Robbin and Dietmar A Salamon. INTRODUCTION TO DIFFERENTIAL GEOMETRY. *ETH, Lecture Notes, preliminary version.*, 2011. 1

[38] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Re. Representation Tradeoffs for Hyperbolic Embeddings. In *ICML*, 2018. 1, 3

[39] Josef Sivic, Bryan C. Russell, Andrew Zisserman, William T. Freeman, and Alexei A. Efros. Unsupervised discovery of visual object class hierarchies. In *CVPR*, 2008. 1

[40] Abraham A. Ungar. The hyperbolic square and Mobius transformations. *Banach Journal of Mathematical Analysis*, 1, 2007. 3

[41] Francisco Rivera Valverde, Juana Valeria Hurtado, and Abhinav Valada. There Is More Than Meets the Eye: Self-Supervised Multi-Object Detection and Tracking With Sound by Distilling Multimodal Knowledge. In *CVPR*, 2021. 2

[42] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. SCAN: Learning to Classify Images Without Labels. In *ECCV*, 2020. 1

[43] Pengwan Yang, Cees GM Snoek, and Yuki M Asano. Self-ordering point clouds. In *ICCV*, 2023. 2

[44] Asano YM., Rupprecht C., and Vedaldi A. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020. 1, 2, 3, 5, 6, 7