Searching for Actions on the Hyperbole

Teng Long[1]*, Pascal Mettes[2], Heng Tao Shen[1]†, Cees Snoek[2]
University of Electronic Science and Technology of China[1], University of Amsterdam[2]

Abstract

In this paper, we introduce hierarchical action search. Starting from the observation that hierarchies are mostly ignored in the action literature, we retrieve not only individual actions but also relevant and related actions, given an action name or video example as input. We propose a hyperbolic action network, which is centered around a hyperbolic space shared by action hierarchies and videos. Our discriminative hyperbolic embedding projects actions on the shared space while jointly optimizing hypernym-hyponym relations between action pairs and a large margin separation between all actions. The projected actions serve as hyperbolic prototypes that we match with projected video representations. The result is a learned space where videos are positioned in entailment cones formed by different subtrees. To perform search in this space, we start from a query and increasingly enlarge its entailment cone to retrieve hierarchically relevant action videos. Experiments on three action datasets with new hierarchy annotations show the effectiveness of our approach for hierarchical action search by name and by video example, regardless of whether queried actions have been seen or not during training. Our implementation is available at https://github.com/Tenglon/hyperbolic_action

1. Introduction

This paper strives to search an action by its name or by an example video. A typical approach in the literature is to frame the retrieval as a recognition problem. Deep networks are trained to match videos to one-hot vectors of action classes [6, 18, 37], which can be used to rank videos using class scores. Others investigate action search by matching videos directly with embedded action names [20, 24, 27, 40] or by matching with a query video [31, 41]. While effective for searching individual actions, common amongst all these works is that hierarchical relations between actions are ignored. Implying the search is optimized for a single action rather than groups of related actions. Hence, search mistakes can be arbitrarily bad as each action is deemed equally dissimilar. To overcome such limitations, we add hierarchical relations to the action search.

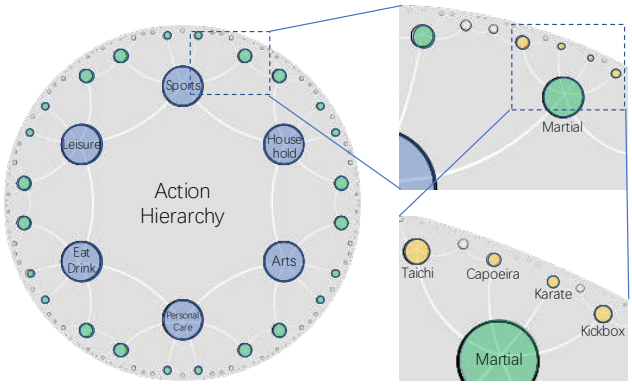


Figure 1: **Hyperboles for action search.** We propose hierarchical search of actions in videos by matching actions and videos in a shared hyperbolic space. In this way, we can search actions by their name or by video examples while abiding to the hierarchical structure of actions.

We are inspired by recent advances in hyperbolic embeddings for hierarchies [16, 29]. The hyperbole provides a natural space for tree hierarchies, as the disc area and circle length grow exponentially with the radius of the space [10, 29]. Hyperbolic spaces such as the Poincaré disk can in fact embed tree hierarchies with arbitrarily low distortion [10]. We seek to obtain a hyperbolic space that is shared by action hierarchies and videos, such that we can perform hierarchical search, see Figure 1. We extend hyperbolic embeddings with a large margin separation to project actions discriminatively on the hyperbole. We use the projected actions as hyperbolic prototypes and introduce a matching with projected video representations. Once trained, we show how to retrieve actions by increasingly enlarging the entailment cone of a query in the shared space.

We make three contributions in this work. First, we introduce discriminative hyperbolic embeddings to position action hierarchies on a hyperbole suitable for search. Second, we propose a matching function between projected actions and videos in the shared hyperbolic space. The pro-

*Work done while at University of Amsterdam.
†Corresponding author.

jected action locations are used as prototypes and we minimize the hyperbolic distance between videos and their corresponding action prototypes. Third, we demonstrate how to perform hierarchical action search by name and video example in the learned hyperbolic space. Experimentally, we find that our hyperbolic approach results in a hierarchically coherent action search, outperforming non-hierarchical approaches from video literature and hierarchical approaches from image literature.

2. Related work

2.1. Action search

In action search, there are in general two ways to provide queries, either by action name or by video example. For query by action name, a common direction is to match the names to detected objects [23, 34, 20, 24]. The match between action names and object detections is typically performed using word embeddings or by matching actions to object hierarchies. Recently, query by action name has also been investigated for searching video moments [3, 27] or video clips [26, 40]. Such a setup provides an effective formulation to retrieve videos from a textual query. The search is however focused on individual actions; it is not possible to search for groups of relevant actions, while mistakes in the search can be arbitrarily bad, because the hierarchical relation to other actions is ignored.

For query by video example, a core focus is on efficiently searching in video collections with a nearest neighbour search from an input video. Ciptadi *et al*. [9] propose movement-based histogram representations for video retrieval. Douze *et al*. [12] model movement with circulant temporal encoding and product quantization to enable a fast video search. Several works have also investigated hashing techniques for query by video example [31, 35, 41]. Common amongst these works is that the semantics of the search is focused on the action of the query video itself. We focus on hierarchical action search in videos, where we retrieve both individual actions, as well as relevant related actions.

2.2. Hierarchical image search

Within the vision literature, class hierarchies have been investigated for image recognition [7, 32] and image retrieval [11]. Beyond supervised recognition, several works have also shown the potential of hierarchies for recognizing unseen image categories [1, 2]. In these works however, hierarchies are either flattened to binary representations [1], used as a structured cost for standard losses [7], or the hierarchical scope is limited to classes from the same parent class [32]. Here, we seek to maintaining the full action hierarchy when searching for action videos.

Recently, Li *et al*. [22] proposed to perform hierarchical image recognition by first generating a three-level hierarchy of classes, followed by a softmax optimization for all three levels. Such a setup performs hierarchical recognition, but this setup is limited to fixed hierarchies. Our approach is suitable for hierarchies of any depth and for trees with varying depth levels. Barz and Denzler [4] embed the hierarchical relations of classes on the hypersphere for hierarchical retrieval. While such a setup improves over a retrieval that ignores hierarchies, we show in this work experimentally that the hyperbole provides a more suitable space for the problem of hierarchical action search.

A number of works have investigated recognition and search using prototypes with pair-wise semantic class similarities. For example, Mettes *et al*. [25] employ hyperspherical prototypes that are positioned based on uniform separation and word embedding similarities between classes. Similar approaches have been proposed in the Euclidean space [8, 21] or on learned manifolds [15]. In this work, we also treat classes as prototypes, but do so in a hyperbolic space, where we can incorporate hierarchical relations amongst all actions.

3. Hyperbolic action network

For the problem of hierarchical action search, we are given a set of action classes $\mathcal{A} = \{1, 2, \cdots, |\mathcal{A}|\}$ and their generic hypernym classes $\mathcal{H} = \{|\mathcal{A}|+1, |\mathcal{A}|+2, \cdots, |\mathcal{A}|+|\mathcal{H}|\}$ connected in a hierarchy. The hierarchy forms a tree of all the hypernym (parent) and hyponym (child) classes. Our goal is to search for actions, while abiding to the hierarchy when ranking the videos. To that end, we propose a hyperbolic action network, which projects actions and videos into a shared space $\mathbb{H}^n_c$, a n-dimensional hyperbolic space with curvature c, see Figure 2. In this paper, we specifically use the Poincaré disk $\mathbb{D}^n_c = \{\mathbf{x} \in \mathbb{R}^n \mid c||\mathbf{x}|| < 1\}$ for the shared space, in line with [29, 36]. We first detail how to position action hierarchies on the shared hyperbolic space in a discriminative manner. Second, we show how to map videos to the same space and how to perform the matching between action hierarchies and videos. Third, we propose hierarchical action search using the trained network.

3.1. Hyperbolic action embedding

The first step in our model is to embed $\mathcal{A} \cup \mathcal{H}$ into the shared space $\mathbb{D}^n_c$. The main idea is to project the hierarchy onto the hyperbole and use the positions of the actions in the hyperbolic space as class prototypes. More formally, we learn $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{|\mathcal{A}|+|\mathcal{H}|}\}$ to represent $\mathcal{A} \cup \mathcal{H}$. Where current works focus on preserving hypernym-hyponym relations in hyperbolic embeddings [36, 29, 16], we use the action positions in the hyperbolic space for the downstream task of search. Therefore, we propose a discriminative hyperbolic embedding for hierarchies, which balances hierarchical relations and large margin separation amongst classes.

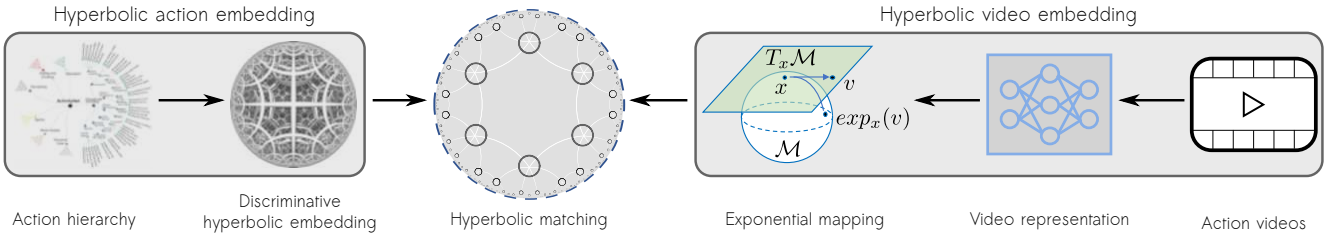


Figure 2: **Overview of the hyperbolic action network.** The actions in a hierarchy are projected on the shared hyperbolic space through a discriminative embedding to obtain action prototypes. Action videos are projected on the same space by feeding them to a 3D ConvNet, followed by an exponential map. We propose a matching function to align the hyperbolic action prototypes with the projected video representations, enabling hierarchical action search.

Let $\mathcal{P} = \{(\mathbf{u}, \mathbf{v}) | \mathbf{u} = h(\mathbf{v})\}$, with $h(\mathbf{v})$ the hypernym of $\mathbf{v}$, denote the positive hypernym pairs and $\mathcal{N} = \{(\mathbf{u}', \mathbf{v}') | \mathbf{u}' \neq h(\mathbf{v}')\}$ the negative pairs. We propose the following loss function to obtain a discriminative hyperbolic embedding:

$$\mathcal{L}_1(\mathcal{P}, \mathcal{N}, \mathbf{P}) = \mathcal{L}_H(\mathcal{P}, \mathcal{N}) + \lambda \cdot \mathcal{L}_S(\mathbf{P}). \quad (1)$$

The loss function consists of two parts. The first part is a hypernym-hyponym relation loss akin to [29]:

$$\mathcal{L}_H(\mathcal{P}, \mathcal{N}) = \sum_{(\mathbf{u},\mathbf{v}) \in \mathcal{P}} \log \left(\frac{e^{-d_c(\mathbf{u},\mathbf{v})}}{\sum_{(\mathbf{u},\mathbf{v}') \in \mathcal{N}} e^{-d_c(\mathbf{u},\mathbf{v}')}} \right). \quad (2)$$

The second part separate all non-hypernym classes:

$$\mathcal{L}_S(\mathbf{P}) = \mathbf{1}^T (\hat{\mathbf{P}} \hat{\mathbf{P}}^T - \mathbf{I}) \mathbf{1}, \quad (3)$$

where $\hat{\mathbf{P}}$ denotes the vector-wise ℓ_2-normalization of $\bar{\mathbf{P}}$, with $\bar{\mathbf{P}} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{|\mathcal{A}|}\}$ the prototypes for the non-hypernym actions. In the loss formulation above, $d_c(\cdot, \cdot)$ denotes the hyperbolic distance:

$$d_c(\mathbf{a}, \mathbf{b}) := \frac{2}{\sqrt{c}} \operatorname{arctanh} \left(\sqrt{c} \| -\mathbf{b} \oplus_c \mathbf{a} \| \right), \quad (4)$$

where $\oplus_c$ indicates the Möbius addition [38] in $\mathbb{D}_c^n$, *i.e.*:

$$\mathbf{a} \oplus_c \mathbf{b} := \frac{\left(1 + 2c\langle \mathbf{a}, \mathbf{b} \rangle + c\|\mathbf{b}\|^2\right) \mathbf{a} + \left(1 - c\|\mathbf{a}\|^2\right) \mathbf{b}}{1 + 2c\langle \mathbf{a}, \mathbf{b} \rangle + c^2 \|\mathbf{a}\|^2 \|\mathbf{b}\|^2}. \quad (5)$$

The proposed loss formulation extends standard hyperbolic embeddings with a discriminative loss that targets large margin separation. The main reason for this is that in search, we aim to be discriminative for the actions we are searching. A large margin separation enables this goal.

While our discriminative hyperbolic embedding results in tree-shaped regions on $\mathbb{D}_c^n$, there are no guarantees of entailment, *i.e.* of a partial order relationship that requires the region of each subtree to be fully covered by their parent tree. Therefore, following [16], we further update $\mathbf{P}$ with the following loss:

$$\mathcal{L}_2 = \sum_{h(\mathbf{u})=h(\mathbf{v})} E(\mathbf{u}, \mathbf{v}) + \sum_{h(\mathbf{u}')\neq h(\mathbf{v}')} \max\left(0, \gamma - E\left(\mathbf{u}', \mathbf{v}'\right)\right), \quad (6)$$

where $E(\mathbf{u}, \mathbf{v})$ measures the angle between $\mathbf{u}$ and $\mathbf{v}$. The first term of Equation (6) encourages $\mathbf{u}$ and $\mathbf{v}$ to point in a similar direction when they share a hypernym ($(h(\mathbf{u}) = h(\mathbf{v}))$. The second term pushes $\mathbf{u}'$ and $\mathbf{v}'$ away angularly if they don't share a hypernym ($h(\mathbf{u}') \neq h(\mathbf{v}')$). Variable γ denotes a margin factor that pushes $\mathbf{u}', \mathbf{v}'$ to be at least γ away. For full details of Equation (6), we refer to [16].

We first optimize our shared hyperbolic space using the loss in Equation (1). Afterwards, we refine the space using Equation (6). Since $\mathbf{P}$ resides in hyperbolic space, we optimize both losses using Riemannian gradient descent [5]:

$$\mathbf{P}_{t+1} = \mathbf{P}_t - \eta_t \nabla_R \mathcal{L}\left(\mathbf{P}_t\right), \quad (7)$$

with ∇_R the Riemannian gradient and η_t the learning rate.

3.2. Matching actions and videos

Hyperbolic video embedding. Second, we need to match videos to actions in the shared hyperbolic space. Let $v \in \mathbb{R}^{W \times H \times T \times 3}$ denote a video consisting of T frames. We first feed v to a 3D ConvNet $\mathbf{v} = \Psi(v; \theta) \in \mathbb{R}^D$ to obtain a D-dimensional video representation using network parameters θ. This function is in Euclidean space and can therefore not directly be matched with the hyperbolic action prototypes. We therefore project the video representation to hyperbolic space through the exponential map [17]:

$$\exp_{\mathbf{x}}^c(\mathbf{v}) = \mathbf{x} \oplus_c \left(\tanh\left(\sqrt{c} \frac{\lambda_x^c \|\mathbf{v}\|}{2} \right) \frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|} \right), \quad (8)$$

where $\mathbf{x}$ states the tangent point that connects tangent space $\mathcal{T}_{\mathbf{x}}\mathbb{D}_c^n$ to $\mathbb{D}_c^n$. Different values for $\mathbf{x}$ leads to different tangent spaces $\mathcal{T}_{\mathbf{x}}\mathbb{D}_c^n$. To eliminate ambiguities, we always set $\mathbf{x} = \mathbf{0}$. The exponential map allows us to project the Euclidean video representation onto the hyperbole in a differentiable network, which we will use for the final matching.

Hyperbolic prototype matching. In our model, the goal is to train 3D ConvNet Ψ to best match videos to hyperbolic action prototypes $\mathbf{P}$. Different from softmax cross-entropy on one-hot vectors, the de facto standard in action recognition networks, our optimization is supervised by $\mathbf{P} \in \mathbb{D}_c^n$.

We are given a training set of N samples, $\{(v_i, y_i)\}_{i=1}^N$, where $v_i \in \mathbb{R}^{W \times H \times T \times 3}$ denotes the i^{th} video sample, $y_i \in \mathcal{A}$ denotes the action label. We optimize the network by minimizing the negative log-likelihood:

$$J(\theta) = -\log p_\theta(y = k|v). \tag{9}$$

In this paper, we propose to define the likelihood itself as the softmax over the negative hyperbolic distance between the action and video embeddings in the shared hyperbolic space:

$$p_\theta(y = k|v) = \frac{\exp\left(-d_c\left(\Psi_e(v;\theta), \phi_c(k)\right)\right)}{\sum_{k'} \exp\left(-d_c\left(\Psi_e(v;\theta), \phi_c(k')\right)\right)}, \tag{10}$$

with

$$\Psi_e(v;\theta) = \exp_{\mathbf{x}}^c(\Psi(v;\theta)). \tag{11}$$

The proposed loss brings representations of a video close to the hyperbolic positions of the action prototypes in hyperbolic space. Similar in spirit to [25], we keep the action prototypes fixed after projection into the shared space.

3.3. Hierarchical action search

The hyperbolic action network aligns videos and action labels in a hierarchical manner on a shared hyperbolic space. In turn, this enables a hierarchical search for seen and unseen actions, along with their hierarchical siblings. The intuition behind this possibility is shown in Figure 3. By design of the hyperbolic space, the siblings of an action class fall under the same entailment cone. We propose two ways to perform hierarchical action search as a function of their entailment cones. These are *search by action name* and *search by video example*. For query $\mathbf{q} \in \mathbb{D}_c^n$, we calculate the distance to candidate sample $\mathbf{x}_i$ as:

$$d_{\mathbf{q}}(\mathbf{x}_i) = 1 - \cos(\mathbf{q}, \mathbf{x}_i). \tag{12}$$

Search is performed by computing the above distance function to all videos in a test set. By design of the network, the search starts from $\mathbf{q}$ and the search region grows angularly in all directions. This growing region can be interpreted as a cone expanding around $\mathbf{q}$.

Despite the simplicity of Equation (12), the use of the cosine distance for search directly matches with the construction of the action hierarchy in the hyperbolic space. Because action trees form entailment cones, the angle in hyperbolic space from an action to its sibling (action with same parent) is smaller than the angle from an action to an unrelated action. Hence, we only need to compute the cosine distance

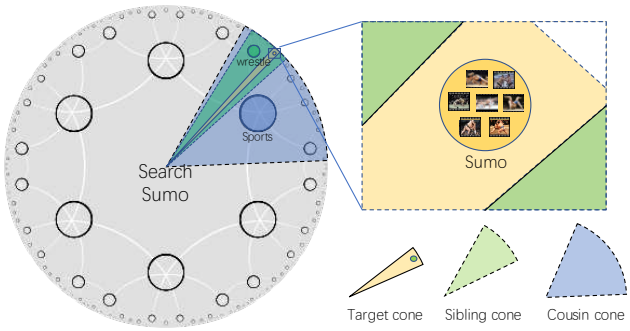


Figure 3: **Hierarchical action search.** For an action query, we start our search from the projection in the hyperbolic space and expand the entailment cone of the query in all directions. Upon enlarging the entailment cone, we first absorb videos from sibling action cones, then cousin cones. Only afterwards, we retrieve actions from other cones.

from a query to each candidate in a test set and rank based on the distances to obtain a search that abides to hierarchical action relations. We investigate two search settings.

Search by action name. In the first search setting, we perform a hierarchical search by starting from an action a in the hierarchy. Our setup allows for both a search for seen and unseen actions, where for the seen actions, labeled videos for action a have been used during hyperbolic alignment, in the latter not. Let S denote a set of videos in a candidate set. We perform a nearest neighbour search between the hyperbolic action embedding $\phi_c(a)$ and the hyperbolic video embedding $\Psi_e(\mathbf{s};\theta)\ \forall\ \mathbf{s} \in S$. All videos in S are ranked based on their cosine distance.

Search by video example. Given our general setup, it is also possible to hierarchically search for actions by providing query video $\mathbf{q}$. We perform a nearest neighbour search in S akin to the hierarchical search by action name.

4. Experimental setup

4.1. Hierarchical retrieval datasets

To enable hierarchical action search, we have revised three well-known action datasets: Activity-Net1.3 [13], Kinetics [6], and Moments-in-Time [28]. We include action hierarchies and action splits for unseen action search experiments. For the hierarchy revision, we follow the protocol of ActivityNet [13] and use the ATUS taxonomy[1]. The seen/unseen split follows the setup of [19] The statistics are shown in Table 1. All revised hierarchies and seen/unseen partitions are available in our implementation.

Hierarchical-ActivityNet. In ActivityNet [13], each untrimmed video consists of one or more action segments.

[1]U.S. Department of Labor. American time use survey. https://www.bls.gov/news.release/pdf/atus.pdf

Table 1: **Hierarchical action search datasets**, building on existing datasets. The actions in Hierarchical-ActivityNet and Hierarchical-Kinetics are at the third level of their respective hierarchy, while Hierarchical-Moments has actions at all levels.

	Source	Number of videos		Actions per level				Seen/unseen split	
		training	validation	1	2	3	4	seen	unseen
Hierarchical-ActivityNet	ActivityNet [13]	15,290	7,569	6	38	200	-	160	40
Hierarchical-Kinetics	mini-Kinetics [39]	77,117	4,897	5	33	200	-	-	-
Hierarchical-Moments	Moments-in-Time [28]	800,575	33,899	45	224	191	27	300	39

We trim the videos into clips based on the provided temporal annotations. The trimming yields ∼23K trimmed videos from 200 classes,∼15K for training and ∼8K for validation. We report our result on the validation set. ActivityNet comes with an action hierarchy, which we have slightly modified to make a more balanced tree and to remove a number of redundant hypernyms. Hierarchy annotations are performed at the action-level instead of the video-level, which makes the annotation burden light. For search experiments on unseen actions, we use 160 seen actions for training and 40 unseen actions for evaluation.

Hierarchical-Kinetics. Mini-Kinetics [39] contains ∼83K videos from 200 classes, ∼78K for training and ∼5K for validation. The official hierarchy has two layers, resulting in 33 parent nodes. We further add a hierarchical layer containing six grandparent nodes, along with slight modifications in the parent hierarchy akin to Hierarchical-ActivityNet. We do not perform zero-shot learning experiments for this dataset to avoid potential overlap with the pre-training of the 3D ConvNet on Kinetics [6].

Hierarchical-Moments. Moments-in-time [28] contains ∼1M clips from 339 classes. ∼800K for training and ∼34K for validation. Moments-in-time does not come with a hierarchy, but the class names are a subset of VerbNet [33]. We use VerbNet to provide an action hierarchy for this dataset, resulting in a tree with different depths for actions, ranging from two to four layers. For this dataset, we use 300 seen actions and 39 unseen actions for evaluation.

4.2. Implementation details

The action embedding is trained with Riemannian Adam [5] using a learning rate of 10^{-4} on a Nvidia GTX 1080TI. The video embedding network Ψ is a ResNeXt-C3D [18] pre-trained on Kinetics [6]. We obtain video representations through average pooling of groups of 16 frames. We use *geoopt*[2] for Riemannian optimization and PyTorch [30] for implementation.

4.3. Evaluation metrics

We split the data as follows: for search by video query, the pool set for each query is the validation set excluding the query. For search by name, the pool set is the entire validation set. For hierarchical search, we not only aim to retrieve the target class, we are also interested in videos from similar classes. These are sibling classes (shared parents, akin to Hascoet *et al*. [19].) and cousin classes (shared grandparents). For sibling retrieval and classification, we consider predictions that require at most two graph hops in the tree hierarchy as correct. For cousins, this expands to four hops. For retrieval, we report the (mean) Average Precision @ 50, for classification, we report the multi-class accuracy. The correct/incorrect labels depend on whether 0-hops (mAP), 2-hops (sibling-mAP), 4-hops (cousin-mAP) are included.

[2] https://github.com/geoopt/geoopt

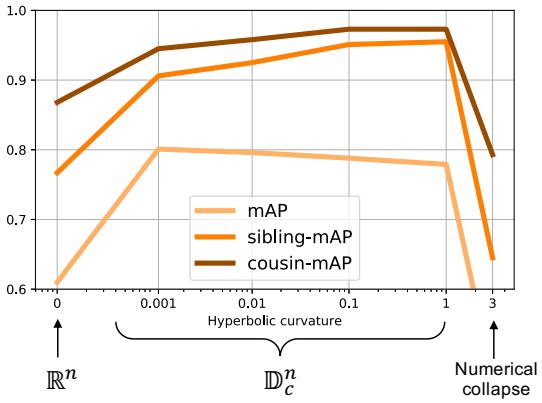


Figure 4: **Effect of hyperbolic curvature** on the hierarchical action search performance of Hierarchical-ActivityNet. We report sibling- and cousin-mAP, as well as standard mAP. For $c = 0$, the space collapses to a Euclidean space, which hurts the scores. For large curvatures, results drop due to numerical instabilities. We recommend a hyperbolic space with $0 < c \leq 1$.

5. Experiments

5.1. Ablation studies

Effect of hyperbolic curvature. The curvature c determines to what extent hyperbolic space $\mathbb{D}_c^n$ is distorted. The curvature can be any real number. For $c > 0$, we have a hyperbole, while for $c = 0$, we recover the Euclidean

Table 2: **Effect of hyperbolic dimensionality** on hierarchical action search accuracy. Our approach obtains high scores in low-dimensional space, outperforming standard softmax cross-entropy optimization using the same base network. Further improvements are obtained when enlarging the shared hyperbolic space.

	Dimensionality					
	5	10	20	50	100	200
mAP						
ResNeXt-C3D	-	-	-	-	-	0.728
This paper	0.671	0.760	0.774	0.785	0.787	**0.789**
sibling-mAP						
ResNeXt-C3D	-	-	-	-	-	0.889
This paper	0.712	0.924	0.947	0.949	**0.950**	0.948
cousin-mAP						
ResNeXt-C3D	-	-	-	-	-	0.945
This paper	0.783	0.955	0.969	**0.971**	**0.971**	0.970

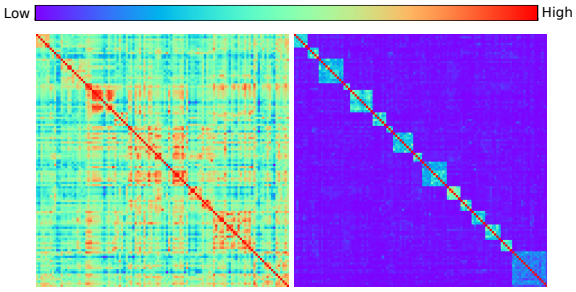
Table 3: **Effect of the discriminative loss** in the hyperbolic embedding on Hierarchical-ActivityNet. Ignoring discrimination ($\lambda = 0$) hurts mAP, while ignoring hierarchies ($\lambda = \infty$) hurts sibling-mAP and cousin-mAP. Weighting both losses equally performs best.

	Hierarchical-ActivityNet		
	mAP	S-mAP	C-mAP
$\lambda = 0$	0.777	0.950	0.971
$\lambda = 0.1$	0.779	0.949	0.969
$\lambda = 1$	0.789	0.948	0.970
$\lambda = 10$	0.793	0.940	0.966
$\lambda = \infty$	0.801	0.891	0.936

space. We demonstrate the effect of different choices for c for query by video in Figure 4. We report sibling-mAP and cousin-mAP, as well as standard mAP.

We observe that suboptimal results are obtained for the Euclidean space, exemplified by the lower scores at $c = 0$. For positive values of $0 < c \leq 1$, scores increase; hyperboles matter for hierarchical action search. For higher values of c, the results drop again, which is due to numerical instabilities in the hyperbolic embedding. Overall, we find that hyperbolic spaces are beneficial for hierarchical search and any value of $0 < c \leq 0.1$ results in a stable result across all three metrics. We therefore employ a fixed value of $c = 0.1$ throughout further experiments.

Effect of hyperbolic dimensionality. We again experiment on Hierarchical-ActivityNet and report all three mAP metrics. The results are shown in Table 2 for query by video. We find that across the metrics, a higher dimensionality results in higher scores. High scores can also be obtained using spaces with much fewer dimensions than classes. For comparison, we perform the search using soft-



(a) One-hot. (b) Hyperbolic (*ours*).

Figure 5: **Visualizing action similarities.** We show the average similarity of videos from a subset of actions in ActivityNet. Our hyperbolic approach learns similarities that adhere to the provided hierarchy. The one-hot baseline does not, resulting in lower scores for hierarchical action search.

max cross-entropy on the one-hot vectors, as is common in action recognition [39]. For the baseline, the dimensionality is always equal to the number of classes. Our high scores at low dimensionality highlight the potential of action hierarchies and encoding them on hyperboles. Throughout our experiments, we focus on action search quality, hence we use 200 dimensions in further experiments.

Effect of the discriminative loss. To embed action hierarchies on the hyperbolic space, our loss optimizes for both hypernym-hyponym relations and separation across action prototypes. In Table 3, we evaluate the effect of the discriminative embedding. When $\lambda = 0$, our embedding is equal to the Poincaré embeddings of Ganea *et al*. [16]. This embedding obtains high sibling- and cousin-mAP, since their approach targets the hierarchical relations only. When adding our discriminative component, the sibling-mAP and cousin-mAP scores are maintained, while standard mAP increases. This is a direct result of moving actions that are leaf nodes away from each other. When using only the discriminative loss ($\lambda = \infty$), we obtain the highest mAP, but at the expense of low sibling-mAP and cousin-mAP. Overall, balancing both losses equally is preferred.

Visualizing action similarities. To evaluate whether we are actually learning hierarchical relations, we show the average video similarities for a subset of the actions on Hierarchical-ActivityNet in Figure 5. We show the pairwise action similarities for our model and compare it to one trained with softmax cross-entropy on one-hot vectors. The confusion matrices show that for our approach, clear substructures emerge which align with the provided action hierarchy. The one-hot baseline discovers a small portion of these hierarchical relations, but also shows a high similarity to many other actions, even though these actions are semantically not related. As a result, the baseline is less likely to obtain high hierarchical action search scores.

Table 4: **Search by action name** comparison on three datasets. We outperform the baselines across datasets and metrics. Barz and Denzler [4] outperform other baselines, while our hyperbolic action network performs best overall.

		Hierarchical-ActivityNet			Hierarchical-Kinetics			Hierarchical-Moments		
	space	mAP	S-mAP	C-mAP	mAP	S-mAP	C-mAP	mAP	S-mAP	C-mAP
ResNextC3D [18]	Δ^n	0.728	0.889	0.945	0.712	0.879	0.918	0.208	0.258	0.438
DeViSE [14]	$\mathbb{R}^n$	0.689	0.870	0.935	0.672	0.859	0.906	0.170	0.216	0.406
Li *et al*. [22]	Δ^n	0.709	0.882	0.942	0.710	0.876	0.916	-	-	-
Mettes *et al*. [25]	$\mathbb{S}^{n-1}$	0.757	0.889	0.940	0.712	0.852	0.897	**0.311**	0.363	0.517
Barz and Denzler [4]	$\mathbb{S}^{n-1}$	0.718	0.907	0.956	0.695	0.898	0.935	0.302	0.364	0.533
This Paper	$\mathbb{D}_c^n$	**0.789**	**0.948**	**0.970**	**0.720**	**0.938**	**0.957**	**0.311**	**0.414**	**0.564**

5.2. Hierarchical action search applications

5.2.1 Search by action name

Setup. For search by action name, we start from action prototypes in the hyperbolic space. We compare to five baselines with different spaces for matching actions and videos. The first uses standard one-hot vectors on the simplex Δ^n with softmax cross-entropy [18]. We use the representations from the last fully-connected layer for retrieval, akin to our approach. We also compare to two baselines that position actions based on word embedding similarities in Euclidean space [14] and on the hypersphere [25]. Lastly, we compare to the recent hierarchical classification of Li *et al*. [22] and the hierarchical retrieval of Barz and Denzler [4]. All baselines use the same video network and settings as our approach. What separates them is their embedding space and how they position action classes in this space.

Results. The results in Table 4 show that across the three datasets, our approach is preferred. Softmax, DeViSE and Mettes *et al*. [25] ignore hierarchical relations amongst actions, resulting in sub-optimal scores. The hierarchical approach of Barz and Denzler [4] outperforms the other baselines, showing that hierarchies are beneficial. The approach of Li *et al*. [22] is restricted to Hierarchical-ActivityNet and Hierarchical-Kinetics, since the hierarchy of Hierarchical-Moments is unbalanced, which can not be handled in their approach. Our action network outperforms both, indicating the importance of hyperboles for action search. We provide qualitative results for three action queries in Figure 6.

5.2.2 Search by video example

Setup. This experiment employs the same datasets, metrics, and baselines as search by action name. Here we also include an internal baseline, using the Euclidean distance instead of the hyperbolic distance in our loss function.

Results. Results in Table 5 demonstrate our approach is preferred across datasets and metrics. With a Euclidean space in the loss we are on par with the baselines, switching to a hyperbolic geometry and loss results in the best overall scores. We provide qualitative results in Figure 7.

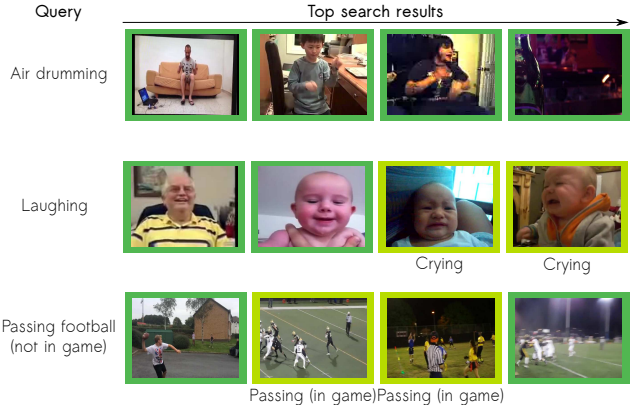


Figure 6: **Search by action name** results on Hierarchical-Kinetics. **Green** denotes the same action, **lime** sibling action. As we start from the action prototype, top results are likely to include the action of interest, as well as sibling actions that are visually similar.

5.2.3 Hierarchical action recognition

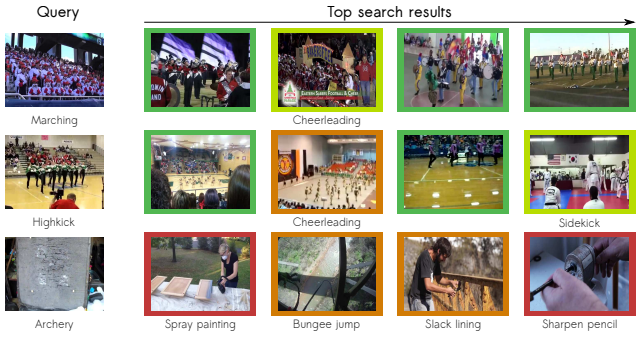
While optimized for hierarchical action search, our approach is also competitive for hierarchical action recognition. To show this, we provide accuracy and sibling-accuracy results on the ActivityNet and Mini-Kinetics datasets in Table 6. On both datasets, we obtain a consistent improvement over the baselines.

5.2.4 Zero-shot action search

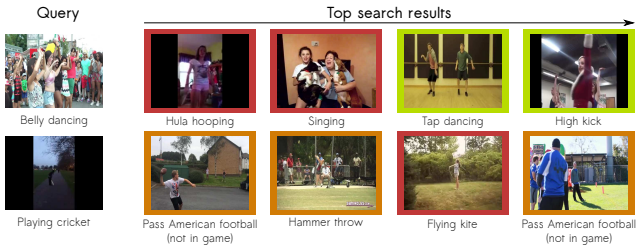
Finally, we investigate the potential of our hierarchical approach for zero-shot action search. Since, actions are already positioned on the hyperbole, it becomes possible to search by any action, regardless of whether training examples have been shown. To evaluate the effectiveness in the zero-shot setting, we perform a search by name, compared to three baselines. The first is the zero-shot approach of Zhang *et al*. [42], which transforms both videos and action word embeddings into a shared Euclidean space for matching. We also compare to the baselines of the supervised action search experiments [4, 22, 25].

Table 5: **Search by video example** comparison on all three datasets. For all datasets and metrics, our approach obtains favourable scores. This holds especially for the sibling-mAP and cousin-mAP, which benefit from hierarchical relations.

		Hierarchical-ActivityNet			**Hierarchical-Kinetics**			**Hierarchical-Moments**		
	space	mAP	S-mAP	C-mAP	mAP	S-mAP	C-mAP	mAP	S-mAP	C-mAP
ResNextC3D [18]	Δ^n	0.592	0.761	0.864	0.532	0.733	0.820	0.145	0.173	0.359
DeViSE [14]	$\mathbb{R}^n$	0.609	0.761	0.860	0.553	0.715	0.803	0.134	0.161	0.348
Li *et al.* [22]	Δ^n	0.583	0.760	0.865	0.552	0.753	0.833	-	-	-
Mettes *et al.* [25]	$\mathbb{S}^{n-1}$	0.587	0.760	0.864	0.551	0.754	0.835	0.142	0.172	0.358
Barz and Denzler [4]	$\mathbb{S}^{n-1}$	0.583	0.747	0.853	0.547	0.725	0.812	0.143	0.172	0.358
This Paper	$\mathbb{R}^n$	0.610	0.767	0.868	0.565	0.738	0.820	0.143	0.172	0.360
This Paper	$\mathbb{D}_c^n$	**0.678**	**0.843**	**0.908**	**0.593**	**0.824**	**0.880**	**0.163**	**0.201**	**0.381**



(a) Search results.



(b) Failure cases.

Figure 7: **Search by video example** results on Hierarchical-Kinetics. **Green** denotes the same action, **lime** sibling action, **orange** a cousin action, and **red** an irrelevant action. In (a), we show a success case, an ambiguous case where we still retrieve relevant actions, and a difficult case where the video content is different from its common setting. In (b), we highlight failures due to ambiguity in the query videos.

The zero-shot action search results are shown in Table 7. Compared to the zero-shot baseline of Zhang *et al.* [42], we perform better, especially in standard mAP and sibling mAP. Closest to our scores are the ones by Barz and Denzler [4]. On Hierarchical-ActivityNet, we outperform all baselines, while on Hierarchical-Moments, we are slightly preferred for mAP and sibling-mAP. We can conclude that our hyperbolic approach is effective for searching unseen actions, highlighting its generalization capabilities.

Table 6: **Hierarchical action recognition** comparison. While designed for hierarchical search, we also find moderate but consistent improvements for action recognition.

		ActivityNet		**Mini-Kinectics**	
	space	acc	S-acc	acc	S-acc
ResNext-C3D [18]	Δ^n	74.0	84.0	77.0	86.5
DeViSE [14]	$\mathbb{R}^n$	72.4	83.9	74.4	85.6
Li *et al.* [22]	Δ^n	74.1	85.3	77.0	86.5
Mettes *et al.* [25]	$\mathbb{S}^{n-1}$	73.8	83.7	76.0	86.2
Barz and Denzler [4]	$\mathbb{S}^{n-1}$	72.3	84.0	76.0	86.1
This Paper	$\mathbb{D}_c^n$	**75.1**	**85.8**	**77.7**	**87.5**

Table 7: **Zero-shot action search** comparison. On both datasets, we obtain the highest scores, especially for standard mAP and sibling-mAP.

	Hierarchical-ActivityNet			**Hierarchical-Moments**		
	mAP	S-mAP	C-mAP	mAP	S-mAP	C-mAP
Zhang *et al.* [42]	0.397	0.449	0.803	0.026	0.027	0.072
Li *et al.* [22]	0.389	0.461	0.821	-	-	-
Mettes *et al.* [25]	0.235	0.281	0.728	0.169	0.171	0.258
Barz and Denzler [4]	0.453	0.527	0.854	0.216	0.219	0.300
This Paper	**0.543**	**0.627**	**0.855**	**0.222**	**0.225**	**0.301**

6. Conclusion

In this work, we introduce hierarchical action search, where we seek not only the action of interest, but also actions that are related in a hierarchical manner. To that end, we propose a hyperbolic action network. Central in this network is a hyperbolic space that is shared by action hierarchies and videos. We project action hierarchies through a discriminative embedding that extends current embeddings with a large margin action separation. The obtained action embeddings form hyperbolic prototypes, which our loss function matches to projected video representations. Experiments on three video datasets, with added hierarchical annotations, show that our approach enables an effective hierarchical search by action name and by video example, outperforming alternative search approaches from the video and image literature.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE TPAMI*, 2015.

[2] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015.

[3] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017.

[4] Björn Barz and Joachim Denzler. Hierarchy-based image embeddings for semantic image retrieval. In *WACV*, 2019.

[5] Gary Becigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *ICLR*, 2019.

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[7] Ju Yong Chang and Kyoung Mu Lee. Large margin learning of hierarchical semantic similarity for image classification. *CVIU*, 132:3–11, 2015.

[8] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *CVPR*, 2018.

[9] Arridhana Ciptadi, Matthew S Goodwin, and James M Rehg. Movement pattern histogram for action recognition and retrieval. In *ECCV*, 2014.

[10] Christopher De Sa, Albert Gu, Christopher Ré, and Frederic Sala. Representation tradeoffs for hyperbolic embeddings. In *ICML*, 2018.

[11] Jia Deng, Alexander C Berg, and Li Fei-Fei. Hierarchical semantic indexing for large scale image retrieval. In *CVPR*, 2011.

[12] Matthijs Douze, Jérôme Revaud, Jakob Verbeek, Hervé Jégou, and Cordelia Schmid. Circulant temporal encoding for video retrieval and temporal alignment. *IJCV*, 119(3):291–306, 2016.

[13] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[14] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013.

[15] Zhenyong Fu, Tao Xiang, Elyor Kodirov, and Shaogang Gong. Zero-shot learning on semantic class prototype graph. *IEEE TPAMI*, 40(8):2009–2022, 2017.

[16] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *ICML*, 2018.

[17] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *NeurIPS*, 2018.

[18] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018.

[19] Tristan Hascoet, Yasuo Ariki, and Tetsuya Takiguchi. On zero-shot recognition of generic objects. In *CVPR*, 2019.

[20] Mihir Jain, Jan C van Gemert, Thomas Mensink, and Cees G M Snoek. Objects2action: Classifying and localizing actions without any video example. In *ICCV*, 2015.

[21] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *CVPR*, 2017.

[22] Aoxue Li, Tiange Luo, Zhiwu Lu, Tao Xiang, and Liwei Wang. Large-scale few-shot learning: Knowledge transfer with class hierarchy. In *CVPR*, 2019.

[23] Xirong Li, Dong Wang, Jianmin Li, and Bo Zhang. Video search in concept subspace: a text-like paradigm. In *CIVR*, 2007.

[24] Pascal Mettes and Cees GM Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *ICCV*, 2017.

[25] Pascal Mettes, Elise van der Pol, and Cees G M Snoek. Hyperspherical prototype networks. In *NeurIPS*, 2019.

[26] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[27] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *CVPR*, 2019.

[28] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, and Carl Vondrick. Moments in time dataset: one million videos for event understanding. *IEEE TPAMI*, 2019.

[29] Maximilian Nickel and Douwe Kiela. Poincaré Embeddings for Learning Hierarchical Representations. In *NeurIPS*, 2017.

[30] Adam Paszke et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*. 2019.

[31] Jie Qin, Li Liu, Mengyang Yu, Yunhong Wang, and Ling Shao. Fast action retrieval from videos via feature disaggregation. *CVIU*, 156, 2017.

[32] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*, 2011.

[33] Karin Kipper Schuler. *Verbnet: a broad-coverage, comprehensive verb lexicon*. PhD thesis, University of Pennsylvania, 2005.

[34] Cees G M Snoek and Marcel Worring. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2(4):215–322, 2009.

[35] Jingkuan Song, Hanwang Zhang, Xiangpeng Li, Lianli Gao, Meng Wang, and Richang Hong. Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE TIP*, 27(7):3210–3221, 2018.

[36] Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. Poincare glove: Hyperbolic word embeddings. In *ICLR*, 2019.

[37] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[38] Abraham A Ungar. The hyperbolic square and mobius transformations. *Banach Journal of Mathematical Analysis*, 2007.

[39] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018.

[40] Huijuan Xu, Kun He, Bryan A Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*, 2019.

[41] Guangnan Ye, Dong Liu, Jun Wang, and Shih-Fu Chang. Large-scale video hashing via structure learning. In *ICCV*, 2013.

[42] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2017.