SAPIENZA
UNIVERSITÀ DI ROMA

Faculty of Information Engineering, Informatics, and Statistics
Department of Computer Science

Doctor of Philosophy in Computer Science
(XXXVII cycle)

Learning with Uncertainty via Hyperbolic Neural Networks

Advisor
Prof. Fabio Galasso

Candidate
Paolo Mandica

Academic Year MMXXIV–MMXXV

La borsa di dottorato è stata cofinanziata con risorse del Programma Operativo Nazionale Ricerca e Innovazione 2014-2020, risorse FSE REACT-EU Azione IV.4 "Dottorati e contratti di ricerca su tematiche dell'innovazione" e Azione IV.5 "Dottorati su tematiche Green"

SAPIENZA
UNIVERSITÀ DI ROMA

Learning with Uncertainty via Hyperbolic Neural Networks

Faculty of Information Engineering, Informatics, and Statistics
Department of Computer Science
Doctor of Philosophy in Computer Science (XXXVII cycle)

Paolo Mandica
ID number 1898788

Advisor
Prof. Fabio Galasso

Academic Year 2024/2025

Thesis defended on 15 January 2025
in front of a Board of Examiners composed by:

Prof. Simone Calderara (chairman)

Prof. Antonio Piccinno

Prof. Giovanna Varni

Learning with Uncertainty via Hyperbolic Neural Networks
PhD Thesis. Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: paolo.mandica@uniroma1.it

La borsa di dottorato è stata cofinanziata con risorse del Programma Operativo Nazionale Ricerca e Innovazione 2014-2020, risorse FSE REACT-EU Azione IV.4 “Dottorati e contratti di ricerca su tematiche dell’innovazione” e Azione IV.5 “Dottorati su tematiche Green”

Abstract

This thesis explores the application of hyperbolic geometry and hyperbolic neural networks across various domains, with a focus on leveraging uncertainty estimation to improve the learning process and performance in complex tasks. We begin with a brief introduction to hyperbolic neural networks, providing the theoretical foundation and key concepts that underpin our subsequent research. This work then spans three main areas: self-supervised representation learning for skeleton-based action recognition, active domain adaptation for semantic segmentation, and multimodal large language models.

First, this thesis investigates self-supervised learning in the context of skeleton-based action recognition, where effective representation learning remains challenging due to the hierarchical nature of human motion data. We introduce hyperbolic neural networks to address this challenge through uncertainty-aware learning, developing a novel Hyperbolic Self-Paced learning model (HYSP). This approach leverages the hyperbolic radius as an uncertainty metric to adaptively pace the learning process, scaling the gradient determined by each sample by the norm of the hyperbolic embedding. When evaluated on standard action recognition benchmarks, HYSP demonstrates superior performance while eliminating the need for computationally expensive negative mining procedures.

Next, we explore active learning for semantic segmentation under domain shift, where efficient label acquisition is crucial for adapting to new environments while keeping labeling costs down. For this challenge, we develop a hyperbolic approach named HALO (Hyperbolic Active Learning Optimization), which interprets the hyperbolic radius as an indicator of data scarcity. By combining the hyperbolic radius with prediction entropy, we obtain an estimator of epistemic uncertainty, which we use for selective annotation of pixels in the image. HALO achieves state-of-the-art results on domain adaptation benchmarks while requiring only a small fraction of target labels, surpassing even fully supervised domain adaptation methods.

Finally, this thesis examines large-scale vision-language modeling, where uncertainty estimation becomes particularly challenging due to the scale and multimodal nature of the data. By developing a novel training strategy for a hyperbolic version of BLIP-2, we demonstrate that hyperbolic learning can be successfully scaled to billion-parameter architectures without compromising stability or performance. Our

approach achieves results comparable to its Euclidean counterpart while providing meaningful uncertainty estimates thanks to hyperbolic embeddings, offering a new perspective on uncertainty quantification in large multimodal models.

Throughout these studies, we demonstrate that learning in hyperbolic space offers unique advantages in estimating uncertainty and improving model performance and efficiency across diverse machine learning tasks. This work contributes to the broader understanding of hyperbolic neural networks and their potential to advance the field of deep learning.

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Fabio Galasso, who has been both a mentor and a leader, guiding me through the most academically meaningful years of my life and inspiring me to pursue excellence while challenging me to achieve my full potential. My most heartfelt thanks goes to my mom, sister, and dad for their unwavering support and encouragement. Even during my lowest moments, when my mood was down, and my willpower faltered, you always gave me the strength to pursue my goals without hesitation. I am also deeply grateful for the presence of my dog, Hela, who kept me company and brought a smile to my face every single day, giving me unconditional love without asking for anything in return. Lastly, I extend my deepest gratitude to all my friends and colleagues, as well as everyone with whom I had the pleasure of sharing both the sunny and cloudy days of this adventurous journey.

I would also like to express my gratitude for the financial and collaborative support that has made this Ph.D. journey possible. I am sincerely appreciative of the resources and insights provided by:

- *Programma Operativo Nazionale Ricerca e Innovazione 2014-2020, risorse FSE REACT-EU Azione IV.5 "Dottorati su tematiche Green"*
- *Panasonic Corporation and the Panasonic PRDCA-AI Lab in Mountain View, California. Thank you for the invaluable support and enriching research discussions throughout these years of collaboration.*
- *Nvidia AI Technology Center (NVAITC) for the provision of computing resources, and to Giuseppe Fiameni for his expert guidance, which ensured their optimal use.*
- *University of California, Berkeley, in particular Trevor Darrell and the BAIR research lab. Thank you for welcoming me as a visiting researcher and for the rewarding collaborative opportunities.*
- *The PNRR MUR project PE0000013-FAIR.*
- *Regione Lazio (Italy), through the PO FSE 2014-2020 program.*
- *Sapienza University, through the CENTS grant (RG123188B3EF6A80).*

Contents

Publications

The research reported in this thesis has contributed to the following publications:

- [41] Franco, Luca and Mandica, Paolo et al. “Hyperbolic Self-paced Learning for Self-supervised Skeleton-based Action Representations”. In: *The Eleventh International Conference on Learning Representations (ICLR)*. 2023.
- [40] Franco, Luca and Mandica, Paolo et al. “Hyperbolic Active Learning for Semantic Segmentation under Domain Shift”. In: *Forty-first International Conference on Machine Learning (ICML)*. 2024.
- [88] Mandica, Paolo and Franco, Luca et al. “Hyperbolic Learning with Multimodal Large Language Models”. In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2024.

List of Figures

List of Tables

Chapter 1

Introduction

Over the past decade, deep learning has revolutionized the field of artificial intelligence, achieving remarkable results across a wide range of applications, from computer vision to natural language processing. As we push the boundaries of traditional neural networks, new challenges arise that demand innovative approaches. Among recent advancements in AI research, hyperbolic neural networks have emerged as a compelling solution to a range of real-world challenges.

Euclidean geometry has long been the default framework for most machine learning models. However, many real-world data structures – such as hierarchies, taxonomies, and complex relational networks – possess an inherent non-Euclidean nature. This mismatch between the geometry of the data and the model's representational space can lead to suboptimal performance and inefficient use of parameters. Hyperbolic spaces [12], characterized by their negative curvature, offer a compelling alternative with unique properties that make them particularly well-suited for representing complex, hierarchical data structures. Unlike Euclidean space, which grows polynomially, hyperbolic spaces grow exponentially, allowing for more efficient embedding of tree-like data structures and capturing of fine-grained relationships between data points.

The potential of hyperbolic neural networks extends beyond mere representation efficiency. Recent research [5, 19, 37, 39] has shown that hyperbolic embeddings can provide valuable insights into data uncertainty and complexity, opening new avenues for improving model robustness, developing new learning strategies, and multimodal integration. Some of these studies [37, 5, 130] focus on training hyperbolic neural

networks for the intrinsic advantages with hierarchical data, only to show post-training the association of the radius with some feature of the data. Others [19, 39] instead, harness the potential of the hyperbolic radius directly at training time, showing how an active use of it can improve performance and interpretability of these models. This thesis explores the application of hyperbolic geometry across several key areas of machine learning, focusing on the utility and meaning of the hyperbolic radius, and demonstrating its potential to enhance performance, efficiency, and interpretability in complex tasks.

We begin with a brief introduction to the fundamental concepts of hyperbolic geometry and hyperbolic neural networks, establishing the theoretical foundation for the research that follows. This groundwork is essential for understanding the distinct advantages hyperbolic geometry provides in estimating uncertainty and capturing hierarchical relationships. Building on this foundation, the thesis is organized around three main research areas: (1) self-supervised representation learning for skeleton-based action recognition, (2) active domain adaptation for semantic segmentation, and (3) multimodal vision-language models.

The first task we take on is the investigation of self-supervised learning for skeleton-based action recognition, where capturing effective representations is challenging due to the hierarchical nature of human motion data. To address this, we propose a novel Hyperbolic Self-Paced learning approach, which we dub HYSP. HYSP leverages hyperbolic embeddings in a self-supervised setting, using the hyperbolic radius as an uncertainty measure to dynamically pace the learning process. By adaptively adjusting each sample's influence on training based on its uncertainty, HYSP enables a more efficient learning progression, whereby samples with lower uncertainty are prioritized with larger gradient scaling. HYSP uses data augmentations to generate two views of each sample, training by matching one (the "online" view) to the other (the "target" view). Hyperbolic uncertainty is obtained as a natural byproduct of the hyperbolic network – thus incurring no additional computational cost compared to Euclidean counterparts. When evaluated on three major skeleton-based action recognition benchmarks (PKU-MMD I, NTU-60, and NTU-120), HYSP demonstrates state-of-the-art performance, excelling on PKU-MMD I and outperforming competitors on two of three downstream tasks in NTU-60 and NTU-120. Additionally, HYSP eliminates the need for computationally expensive negative mining by using only positive pairs, further improving efficiency.

Another critical challenge in modern machine learning is the need for large amounts of labeled data. This requirement can be particularly burdensome in domains such as semantic segmentation, where pixel-level annotations are time-consuming and expensive to obtain. Active learning strategies aim to mitigate this issue by intelligently selecting the most informative samples for labeling. However, existing methods [145, 135, 119, 118, 95] often fail to capture the full complexity of the data distribution, resulting in suboptimal sample selection. Building on insights from HYSP, which revealed a meaningful correlation between hyperbolic radius and sample uncertainty, we developed HALO (Hyperbolic Active Learning Optimization), a novel method for pixel-level active learning in semantic segmentation under domain shift. HALO interprets the hyperbolic radius as a proxy for data scarcity, proposing a data acquisition strategy based on epistemic uncertainty. The hyperbolic radius, complemented by the widely-adopted prediction entropy, effectively approximates epistemic uncertainty, leading to an intuitive selection of data point that are the least known by the model. We conduct extensive experimental analysis on two synthetic-to-real benchmarks, GTAV $\rightarrow$ Cityscapes and SYNTHIA $\rightarrow$ Cityscapes, and further test HALO on the Cityscapes $\rightarrow$ ACDC benchmark to evaluate domain adaptation under adverse weather conditions. HALO sets a new state-of-the-art in active learning for semantic segmentation under domain shift, outperforming traditional supervised domain adaptation with only a fraction of labels (1%). HALO's efficacy across both convolutional and attention-based backbones underscores its versatility, with important implications for reducing labeling costs and enabling more efficient data annotation in real-world applications.

As the need for larger and increasingly powerful models grows, and given the lack of studies on the application of hyperbolic neural networks in this domain, we chose to explore the integration of hyperbolic embeddings into large-scale vision-language models, specifically adapting the BLIP-2 architecture [75]. Hyperbolic embeddings have shown strong potential for capturing hierarchical relationships and uncertainty in tasks like action recognition, image segmentation and active learning; however, their use in vision-language models remains largely unexplored. A notable exception, MERU [35], demonstrates the viability of hyperbolic embeddings in large models, albeit limited to hundreds of millions of parameters with the CLIP ViT-large architecture. Our work scales multimodal hyperbolic models by orders of magnitude, integrating hyperbolic representations into an architecture with billions of parameters, while tackling the inherent training complexities. Despite challenges,

we developed a novel training strategy that preserves BLIP-2's performance level in Euclidean space, while providing meaningful uncertainty estimates from hyperbolic embeddings. This contribution advances the application of hyperbolic geometry in state-of-the-art multimodal AI systems, setting the stage for broader adoption of hyperbolic representations in large-scale vision-language models.

In summary, this thesis explores hyperbolic neural networks's potential to address key challenges in machine learning. Our work not only pushes the state-of-the-art in specific tasks but also advances the broader understanding of non-Euclidean geometries in AI and contributes to a growing discourse on geometry's role in artificial intelligence. By applying hyperbolic neural networks across self-supervised learning, active learning, and multimodal integration, we aim to inspire further exploration in domains that benefit from uncertainty quantification and hierarchical representations. Hyperbolic spaces, with their exponential growth properties, present novel avenues for efficient, interpretable, and robust AI systems that align more closely with human-like reasoning on complex data structures.

Chapter 2

A Brief Introduction to Hyperbolic Neural Networks

In recent years, the field of deep learning has seen a growing interest in non-Euclidean geometries, particularly hyperbolic geometry, as an alternative to the more conventional Euclidean framework. This shift is driven by the understanding that many types of data – especially those with hierarchical or tree-like structures – are inherently difficult to represent effectively in Euclidean spaces. Hyperbolic geometry, with its unique properties of exponential expansion and natural embedding of hierarchical structures, has emerged as a compelling solution for modeling such complex data.

Hyperbolic Neural Networks (HNNs) represent a significant shift in neural network architecture and learning paradigms. By operating in hyperbolic space, HNNs offer a novel way to learn and represent data with inherent hierarchies or graph-like structures. The key advantage of HNNs lies in their ability to capture and maintain the intricate relationships within data that often become distorted or lost in Euclidean spaces. This capability has profound implications for a wide range of applications, from natural language processing and computer vision to graph analysis and beyond.

2.1 Hyperbolic Geometry

This section provides a focused overview of the key formulas and concepts in hyperbolic geometry essential for understanding the hyperbolic neural networks discussed in subsequent chapters. Rather than offering an exhaustive exploration, we present the fundamental mathematical principles underpinning these advanced neural architectures. Our goal is to establish a solid foundation that facilitates a deeper understanding of hyperbolic neural networks and their applications. For a more comprehensive explanation of hyperbolic spaces, readers are encouraged to refer to the works of Ganea et al. [45] and Cannon et al. [12].

2.1.1 The Poincaré Ball Model

Hyperbolic space can be represented through various isometric models, each tailored to specific applications. Among these, the Poincaré ball model has gained prominence in machine learning applications due to its advantageous properties. As this model serves as the foundation for our work in the subsequent chapters, we will focus our discussion on the Poincaré ball representation of hyperbolic space.

The Poincaré ball model is defined as the pair $(\mathbb{D}_c^N, g^{\mathbb{D}_c})$, where $\mathbb{D}_c^N$ is the manifold and $g^{\mathbb{D}_c}$ is the associated Riemannian metric:

$$\mathbb{D}_c^N = \{x \in \mathbb{R}^N : c\|x\| < 1\} \tag{2.1}$$

$$g_x^{\mathbb{D}_c} = (\lambda_x^c)^2 g^{\mathbb{E}} \tag{2.2}$$

Here, $c > 0$ represents the curvature of the hyperbolic space, $\|\cdot\|$ denotes the standard Euclidean L^2-norm, $\lambda_x^c = \frac{2}{1-c\|x\|^2}$ is the conformal factor, and $g^{\mathbb{E}} = \mathbb{I}^N$ is the Euclidean metric tensor, which is conformal to the hyperbolic one $g_x^{\mathbb{D}_c}$.

In the Poincaré model, geodesics are curves that minimize the distance between any two points, analogous to straight lines in Euclidean geometry. These geodesics appear either as arcs of circles that intersect the boundary circle orthogonally, or as diameters of the disk.

Figure 2.1 illustrates and describes the main properties of the Poincaré model

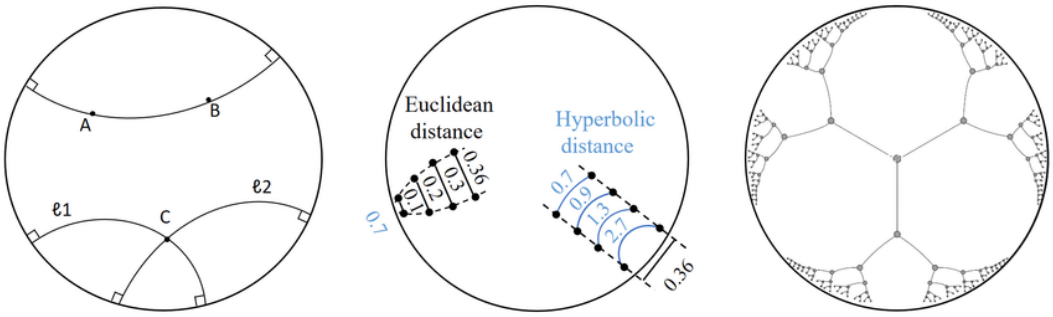

Figure 2.1. Visualization of the Poincaré disk model demonstrating key properties of hyperbolic geometry (reproduced from [103]). **(left)** Illustration of parallel geodesics: points A and B are connected by a unique geodesic; there exist infinite geodesics (e.g., ℓ_1 and ℓ_2) that do not intersect with the original one AB, demonstrating the violation of Euclid's parallel postulate. **(center)** Comparison between Euclidean and hyperbolic distances between pairs of points, highlighting the metric distortion inherent to the model. **(right)** Embedding of a tree structure, showcasing how the exponential growth of space from center to boundary in the Poincaré disk naturally accommodates hierarchical data structures.

in 2 dimensions (i.e., Poincaré disk model).

Exponential Map

The exponential map [45] is a fundamental operation in hyperbolic geometry, serving as a bridge between Euclidean and hyperbolic spaces. It allows us to project vectors from the tangent space of a point in the Poincaré ball (which is essentially a Euclidean space) onto the hyperbolic manifold itself.

Mathematically, given a feature vector $v \in \mathbb{R}^N$ extracted in Euclidean space, we project it into the Poincaré ball using the exponential map defined as:

$$\exp_x^c(v) = x \oplus_c \left(\frac{v}{\sqrt{c}\|v\|} \tanh\left(\sqrt{c} \frac{\lambda_x^c \|v\|}{2} \right) \right) \tag{2.3}$$

where $x \in \mathbb{D}_c^N$ is the anchor point and $\oplus_c$ is the Möbius addition (later defined in Sec. 2.1.1).

If the origin, i.e. $\mathbf{0}$, is chosen as anchor point, then the exponential map becomes:

$$\exp_{\mathbf{0}}^{c}(v) = \frac{v}{\sqrt{c}\|v\|}\tanh(\sqrt{c}\|v\|) \tag{2.4}$$

Logarithmic Map

The logarithmic map is essentially the inverse operation of the exponential map, and it allows for the projection back from hyperbolic to Euclidean space. It is defined as:

$$\log_{x}^{c}(v) = \frac{2}{\sqrt{c}\lambda_{x}^{c}}\tanh^{-1}(\sqrt{c}\|-x\oplus_{c}y\|)\frac{-x\oplus_{c}y}{\|-x\oplus_{c}y\|} \tag{2.5}$$

Again, if the origin, i.e. $\mathbf{0}$, is chosen as anchor point, then the logarithmic map simplifies into:

$$\log_{\mathbf{0}}^{c}(v) = \tanh^{-1}(\sqrt{c}\|y\|)\frac{y}{\sqrt{c}\|y\|} \tag{2.6}$$

Möbius Addition

Möbius addition is a fundamental operation in hyperbolic geometry, serving as the hyperbolic counterpart to vector addition in Euclidean space. In Euclidean geometry, adding two vectors results in a new vector with a length and direction determined by the sum of the originals. However, hyperbolic space's non-Euclidean nature, where parallel lines diverge and geodesics differ from straight lines, necessitates a different approach. Möbius addition accounts for these geometric peculiarities, ensuring that the result of combining two points in the Poincaré ball remains within the ball, respecting its boundary and curvature. Note that, unlike Euclidean addition, Möbius addition is non-commutative.

The Möbius addition $\oplus_c$ of two hyperbolic vectors $h, w \in \mathbb{D}_c^N$ is defined as:

$$h\oplus_{c}w = \frac{(1+2c\langle h,w\rangle + c\|w\|^2)h + (1-c\|h\|^2)w}{1+2c\langle h,w\rangle + c^2\|h\|^2\|w\|^2} \tag{2.7}$$

When $c = 0$, the operation reverts to the Euclidean addition of two vectors in $\mathbb{R}^n$.

Poincaré Distance

The distance between two vectors in the Poincaré ball model is the length of the geodesic connecting them. In hyperbolic geometry, a geodesic is the shortest path between two points, analogous to a straight line in Euclidean geometry.

We mathematically define the distance d_c between two vectors $x, y \in \mathbb{D}_c^N$ on the Poincaré ball as:

$$d_c(x, y) = \frac{2}{\sqrt{c}} \tanh^{-1} \left(\sqrt{c} \| -x \oplus_c y \| \right) \tag{2.8}$$

The Poincaré distance reflects the unique geometry of hyperbolic space, resulting in a perception of distance that differs significantly from that in Euclidean space. A distinctive feature of this metric is its behavior near the boundary of the Poincaré ball: as points approach the edge, the hyperbolic distance between them increases exponentially, even when their Euclidean distance remains small (see Fig. 2.1 (center)). This phenomenon arises from the exponential expansion of volume as one moves from the origin to the border of the ball, which is a direct consequence of the negative curvature of the hyperbolic space.

Hyperbolic Radius

In the Poincaré ball model, points are located within an $N-$dimensional ball of radius $1/\sqrt{c}$, centered at the origin, where $c > 0$ is the curvature of the hyperbolic space. The hyperbolic radius of an embedding $h \in \mathbb{D}_c^N$ is defined as its Poincaré distance from the origin:

$$d_c(h, 0) = \frac{2}{\sqrt{c}} \tanh^{-1} \left(\sqrt{c} \| h \| \right) \tag{2.9}$$

2.1.2 Fréchet Mean

The Fréchet mean is a generalization of the arithmetic mean to non-Euclidean spaces, such as the hyperbolic space. In the context of hyperbolic geometry, the Fréchet mean represents the point that minimizes the sum of squared distances to a given set of points. For a set of points $x_1, ..., x_M \in \mathbb{D}_c^N$, the Fréchet mean μ is

defined as:

$$\mu = \arg\min_{\mu \in \mathbb{D}_c^N} \sum_{i=1}^{M} d_c^2(x_i, \mu) \tag{2.10}$$

Unlike in Euclidean space, where the arithmetic mean provides a simple closed-form solution, the Fréchet mean in hyperbolic space typically requires recursive algorithms to be computed.

Riemannian Variance

For a set of hyperbolic vectors $x_1, ..., x_M \in \mathbb{D}_c^N$, we can define the Riemannian variance:

$$\sigma^2 = \frac{1}{M} \sum_{i=1}^{M} d_c^2(x_i, \mu) \tag{2.11}$$

where μ is the Fréchet mean, which minimizes the Riemannian variance. While μ cannot be computed in closed form, it can be approximated using recursive algorithms.

2.2 Hyperbolic Neural Networks

Hyperbolic Neural Networks (HNNs) represent a novel paradigm in deep learning, designed to leverage the unique properties of hyperbolic geometry. These networks are particularly adept at capturing hierarchical and tree-like structures inherent in many types of data, offering advantages over traditional Euclidean neural networks in certain domains.

2.2.1 Architecture and Operation

Hyperbolic neural networks typically employ a hybrid architecture, integrating elements from both Euclidean and hyperbolic geometries. While recent years have seen a shift towards fully-hyperbolic neural networks [125, 7], these models are still in their infancy. Their effectiveness has been demonstrated primarily on smaller datasets, limiting their applicability to more complex tasks. Given the sophisticated challenges addressed in this thesis, we opted for larger, more established

network architectures that have not yet been fully converted to hyperbolic space. These networks maintain their foundational structure in Euclidean space while incorporating hyperbolic components strategically. The general framework of these hybrid hyperbolic neural networks can be outlined as follows:

1. **Euclidean Feature Extraction**: The initial layers of an HNN often consist of standard Euclidean neural network components (e.g., convolutional layers in image processing tasks). These layers are responsible for extracting relevant features from the input data in Euclidean space.

2. **Hyperbolic Projection**: The transition from Euclidean to hyperbolic space occurs in the latter part of the network. This is typically achieved by replacing the final layer(s) of a traditional neural network with a hyperbolic projection layer. The exponential map (expmap) operation, as defined in Equation 2.4, is used to project the Euclidean features into the Poincaré ball model of hyperbolic space.

3. **Hyperbolic Operations**: Once in hyperbolic space, subsequent operations are performed using hyperbolic analogues of common neural network components, such as hyperbolic feed-forward layers or hyperbolic convolutional layers, depending on the specific architecture and task.

4. **Classification in Hyperbolic Space**: For classification tasks, the final layer often employs Hyperbolic Multinomial Logistic Regression (HyperMLR). This method, as described in Equations 2.12-2.14, uses gyroplanes in hyperbolic space to separate different classes.

2.2.2 Advantages and Considerations

The primary advantage of HNNs lies in their ability to efficiently represent hierarchical structures. The negative curvature of hyperbolic space allows for exponential growth in volume as one moves away from the origin, which naturally accommodates tree-like and hierarchical data structures.

Key benefits include:

- **Improved Representation of Hierarchies**: HNNs can capture complex hierarchical relationships more efficiently than their Euclidean counterparts,

potentially leading to better performance on tasks involving hierarchical or graph-structured data.

- **Compact Embeddings**: The properties of hyperbolic space often allow for more compact representations of complex data structures, potentially reducing the dimensionality required for effective modeling.

- **Enhanced Generalization**: In some cases, the inductive bias introduced by hyperbolic geometry can lead to improved generalization, particularly for data with inherent hierarchical structure.

However, working with HNNs also presents certain challenges, mainly revolving around computational complexity and numerical (in)stability. The first challenge arises from the inherent higher computational cost of operations in hyperbolic space, especially compared to their Euclidean counterparts. A clear example is the vector addition, which in hyperbolic space is the Möbius addition (see Sec. 2.7), that requires much more gpu memory for the calculation compared to the Euclidean one. Secondly, HNNs suffer from numerical instability, particularly when working with points near the boundary of the Poincaré ball due to the higher possibility of having exploding gradient. This problem has been tackled in many different ways, such as using feature clipping [55], curriculum learning [41], or carefully initializing the network parameters [126].

2.2.3 Hyperbolic Multinomial Logistic Regression (MLR)

For classification tasks in hyperbolic space, Hyperbolic Multinomial Logistic Regression [45, 5] is typically used. Given an image feature $z_i \in \mathbb{R}^N$ projected onto the Poincaré ball as $h_i = \exp_x^c(z_i) \in \mathbb{D}_c^N$, we classify it using a set of hyperplanes (gyroplanes) H_y^c for each class y:

$$H_y^c = \{h_i \in \mathbb{D}_c^N, \langle -p_y \oplus_c h_i, w_y \rangle\} \tag{2.12}$$

where p_y is the gyroplane offset and w_y is the orientation for class y. The distance between an embedding h_i and the gyroplane H_y^c is given by:

$$d(h_i, H_y^c) = \frac{1}{\sqrt{c}} \sinh^{-1}\left(\frac{2\sqrt{c}\langle -p_y \oplus_c h_i, w_y \rangle}{(1 - c\| - p_y \oplus_c h_i\|^2)\|w_y\|}\right) \tag{2.13}$$

The likelihood of a class y given an embedding h_i is then defined as:

$$p(\hat{y}_i = y|h_i) \propto \exp(\zeta_y(h_i)) \tag{2.14}$$

where $\zeta_y(h_i) = \lambda^c_{p_y}\|w_y\|d(h_i, H^c_y)$ is the logit for class y.

Chapter 3

Hyperbolic Self-paced Learning for Self-supervised Skeleton-based Action Representations

3.1 Overview

Starting from the seminal work of [72], the machine learning community has started looking at self-paced learning, i.e. determining the ideal sample order, from easy to complex, to improve the model performance. Self-paced learning has been adopted so far for weakly-supervised learning [86, 139, 113], or where some initial knowledge is available, e.g. from a source model, in unsupervised domain adaption [86]. Self-paced approaches use the label (or pseudo-label) confidence to select easier samples and train on those first. However labels are not available in self-supervised learning (SSL) [24, 58, 50, 27], where the supervision comes from the data structure itself, i.e. from the sample embeddings.

We propose HYSP, the first HYperbolic Self-Paced learning model for SSL. In HYSP, the self-pacing confidence is provided by the hyperbolic uncertainty [45, 117] of each data sample. In more details, we adopt the Poincaré Ball model [130, 45, 67, 37] and define the certainty of each sample as its embedding radius. The hyperbolic uncertainty is a property of each data sample in hyperbolic space, and it is therefore

available while training with SSL algorithms.

HYSP stems from the belief that the uncertainty of samples matures during the SSL training and that more certain ones should drive the training more prominently, with a larger pace, at each stage of training. In fact, hyperbolic uncertainty is trained end-to-end and it matures as the training proceeds, i.e. data samples become more certain. We consider the task of human action recognition, which has drawn growing attention [122, 76, 52, 29, 69] due to its vast range of applications, including surveillance, behavior analysis, assisted living and human-computer interaction, while being skeletons convenient light-weight representations, privacy preserving and generalizable beyond people appearance [147, 78].

HYSP builds on top of a recent self-supervised approach, SkeletonCLR [76], for training skeleton-based action representations. HYSP generates two views for the input samples by data augmentations [58, 24, 14, 50, 27, 76], which are then processed with two Siamese networks, to produce two sample representations: an *online* and a *target*. The training proceeds by tasking the former to match the latter, being both of them *positives*, i.e. two *views* of the same sample. HYSP only considers positives during training and it requires curriculum learning. The latter stems from the vanishing gradients of hyperbolic embeddings upon initialization, due to their initial overconfidence (high radius) [56]. So initially, we only consider angles, which coincides with starting from the conformal Euclidean optimization. The former is because matching two embeddings in hyperbolic implies matching their uncertainty too, which appears ill-posed for negatives from different samples, i.e. uncertainty is specific of each sample at each stage of training. This bypasses the complex and computationally-demanding procedures of negative mining, which contrastive techniques require[1]. Both aspects are discussed in detail in Sec. 3.3.2.

We evaluate HYSP on three most recent and widely-adopted action recognition datasets, NTU-60, NTU-120 and PKU-MMD I. Following standard protocols, we pre-train with SSL, then transfer to a downstream skeleton-base action classification task. HYSP outperforms the state-of-the-art on PKU-MMD I, as well as on 2 out of 3 downstream tasks on NTU-60 and NTU-120.

[1]Similarly to BYOL [50], HYSP is not a contrastive technique, as it only adopts positives.

3.2 Related Work

HYSP embraces work from four research fields, for the first time, as we detail in this section: self-paced learning, hyperbolic neural networks, self-supervision and skeleton-based action recognition.

3.2.1 Self-paced Learning

Self-paced learning (SPL), initially introduced by [72] is an extension of curriculum learning [9] which automatically orders examples during training based on their difficulty. Methods for self-paced learning can be roughly divided into two categories. One set of methods [64, 142] employ it for fully supervised problems. For example, [64, 142] employ it for image classification. [64] enhance SPL by considering the diversity of the training examples together with their hardness to select samples. [142] studies SPL when training with limited time budget and noisy data.
The second category of methods employ it for weakly or semi supervised learning. Methods in [113, 156] adopt it for weakly supervised object detection, where [113] iteratively selects the most reliable images and bounding boxes, while [156] uses saliency detection for self-pacing. Methods in [102, 139] employ SPL for semi-supervised segmentation. [102] adds a regularization term in the loss to learn importance weights jointly with network parameters. [139] considers the prediction uncertainty and uses a generalized Jensen Shannon Divergence loss. Both categories require the notion of classes and not apply to SSL frameworks, where sample embeddings are only available.

3.2.2 Hyperbolic Neural Networks

Hyperbolic representation learning in deep neural networks gained momentum after the pioneering work hyperNNs [45], which proposes hyperbolic counterparts for the classical (Euclidean) fully-connected layers, multinomial logistic regression and RNNs. Other representative work has introduced hyperbolic convolution neural networks [117], hyperbolic graph neural networks [83, 18], and hyperbolic attention networks [51]. Two distinct aspects have motivated hyperbolic geometry: their capability to encode hierarchies and tree-like structures [16, 67], also inspiring [104] for skeleton-based action recognition; and their notion of uncertainty, encoded by

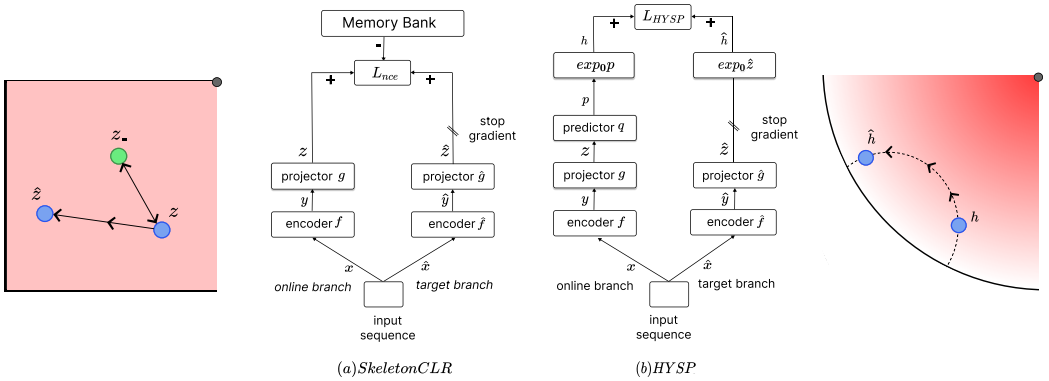

Figure 3.1. (a) SkeletonCLR is a contrastive learning approach based on pushing the embeddings of two views x and $\hat{x}$ of the same sample (*positives*) close to each other, while repulsing from embeddings of other samples (*negatives*). In the represented Euclidean space, on the left side, the attracting/repulsive force is a straight line. (b) The proposed HYSP maps the embeddings into a hyperbolic space, where the sample uncertainty determines the learning pace. The hyperbolic Poincaré ball, on the right side, illustrates the embeddings, where the radius indicates the uncertainty. The attractive force proceeds on a circle, orthogonal to the boundary circle. So the force grows polynomially with the radius. In HYSP, learning is paced by the uncertainty, which the model learns end-to-end during SSL.

the embedding radius [130, 5], which we leverage here. More recently, hyperbolic geometry has been used with self-supervision [148, 130, 37]. Particularly, [130] and [37] let hierarchical actions and image categories emerge from the unlabelled data using RNNs and Vision Transformers, respectively. Differently, our HYSP uses GCNs and it applies to skeleton data. Moreover, unlike all prior works, HYSP leverages hyperbolic uncertainty to drive the training by letting more certain pairs steer the training more.

3.2.3 Self-supervised Learning

Self-supervised learning aims to learn discriminative feature representations from unlabeled data. The supervisory signal is typically derived from the input using certain *pretext* tasks and it is used to pre-train the encoder, which is then transferred to the downstream task of interest. Among all the proposed methods [96, 157, 47, 13, 14], a particularly effective one is the contrastive learning, introduced by SimCLR [24]. Contrastive learning brings different views of the same image closer

('positive pairs'), and pushes apart a large number of different ones ('negative pairs'). MoCo [58, 28] partially alleviates the requirement of a large batch size by using a memory-bank of negatives. BYOL [50] pioneers on training with positive pairs only. Without the use of negatives, the risk of collapsing to trivial solutions is avoided by means of a predictor in one of the Siamese networks, and a momentum-based encoder with stop-gradient in the other. SimSiam [27] further simplifies the design by removing the momentum encoder. Here we draw on the BYOL design and adopt positive-only pairs for skeleton-based action recognition. We experimentally show that including negatives harm training self-supervised hyperbolic neural networks.

3.2.4 Skeleton-based Action Recognition

Most recent work in (supervised) skeleton-based action recognition [29, 30, 138] adopts the ST-GCN model of [150] for its simplicity and performance. We also use ST-GCN. The self-supervised skeleton-based action recognition methods can be broadly divided into two categories. The first category uses encoder-decoder architectures to reconstruct the input [159, 128, 98, 151, 129]. The most representative example is AE-L [98], which uses Laplacian regularization to enforce representations aware of the skeletal geometry. The other category of methods [78, 131, 76, 107] uses contrastive learning with negative samples. The most representative method is SkeletonCLR [76] which uses momentum encoder with memory-augmented contrastive learning, similar to MoCo. More recent techniques, CrosSCLR [76] and AimCLR [52], are based on SkeletonCLR. The former extends it by leveraging multiple views of the skeleton to mine more positives; the latter by introducing extreme augmentations, features dropout, and nearest neighbors mining for extra positives. We also build on SkeletonCLR, but we turn it into positive-only (BYOL), because only uncertainty between positives is justifiable (cf. Sec. 3.3.2). BYOL for skeleton-based SSL has first been introduced by [93]. Differently from them, HYSP introduces self-paced SSL learning with hyperbolic uncertainty.

3.3 Method

We start by motivating hyperbolic self-paced learning, then we describe the baseline SkeletonCLR algorithm and the proposed HYSP. We design HYSP according to the principle that more certain samples should drive the learning process more

predominantly. Self-pacing is determined by the sample target embedding, which the sample online embeddings are trained to match. Provided the same distance between target and online values, HYSP gradients are larger for more certain targets[2]. Since in SSL no prior information is given about the samples, uncertainty needs to be estimated end-to-end, alongside the main objective. This is realized in HYSP by means of a hyperbolic mapping and distance, combined with the positives-only design of BYOL [50] and a curriculum learning [9] which gradually transitions from the Euclidean to the uncertainty-aware hyperbolic space [130, 67].

3.3.1 Background

A few most recent techniques [52, 76] for self-supervised skeleton representation learning adopt SkeletonCLR [76]. That is a contrastive learning approach, which forces two augmented views of an action instance (positives) to be embedded closely, while repulsing different instance embeddings (negatives).
Let us consider Fig. 3.1*(a)*. The positives x and $\hat{x}$ are two different augmentations of the same action sequence, e.g. subject to shear and temporal crop. The samples are encoded into $y = f(x, \theta)$ and $\hat{y} = \hat{f}(\hat{x}, \hat{\theta})$ by two Siamese networks, an online f and a target $\hat{f}$ encoder with parameters θ and $\hat{\theta}$ respectively. Finally the embeddings z and $\hat{z}$ are obtained by two Siamese projector MLPs, g and $\hat{g}$.
SkeletonCLR also uses negative samples which are embeddings of non-positive instances, queued in a memory bank after each iteration, and dequeued according to a first-in-first-out principle. Positives and negatives are employed within the Noise Contrastive Estimation loss [97]:

$$L_{nce} = -\log \frac{\exp(z \cdot \hat{z}/\tau)}{\exp(z \cdot \hat{z}/\tau) + \sum_{i=1}^{M} \exp(z \cdot m_i/\tau)} \tag{3.1}$$

where $z \cdot \hat{z}$ is the normalised dot product (cosine similarity), m_i is a negative sample from the queue and τ is the softmax temperature hyper-parameter.
Two aspects are peculiar of SkeletonCLR, important for its robustness and performance: the momentum encoding for $\hat{f}$ and the stop-gradient on the corresponding branch. In fact, the parameters of the target encoder $\hat{f}$ are not updated by back-propagation but by an exponential moving average from the online encoder, i.e $\hat{\theta} \leftarrow \alpha\hat{\theta} + (1 - \alpha)\theta$, where α is the momentum coefficient. A similar notation and

[2]Gradients generally depend on the distance between the estimation (here online) and the reference (here target) ground truth, as a function of the varying estimation. In self-paced learning, references vary during training.

procedure is adopted for $\hat{g}$. Finally, stop-gradient enables the model to only update the online branch, keeping the other as the reference "target".

3.3.2 Hyperbolic self-paced learning

Basing on SkeletonCLR, we propose to learn by comparing the online h and target $\hat{h}$ representations in hyperbolic embedding space. We do so by mapping the outputs of both branches into the Poincaré ball model centered at $\mathbf{0}$ by an *exponential map* [45] (cf. Eq. 2.4. h and $\hat{h}$ are compared by means of the Poincaré distance (Eq. 2.8), which can be also written as follows:

$$L_{poin}(h, \hat{h}) = \cosh^{-1}\left(1 + 2\frac{\|h - \hat{h}\|^2}{(1 - \|h\|^2)(1 - \|\hat{h}\|^2)}\right) \tag{3.2}$$

with $\|h\|$ and $\|\hat{h}\|$ the radii of h and $\hat{h}$ in the Poincaré ball.
Following [130][3], we define the hyperbolic uncertainty of an embedding $\hat{h}$ as:

$$u_{\hat{h}} = 1 - \|\hat{h}\| \tag{3.3}$$

Note that the hyperbolic space volume increases with the radius, and the Poincaré distance is exponentially larger for embeddings closer to the hyperbolic ball edge, i.e. the matching of *certain* embeddings is penalized exponentially more. Upon training, this yields larger uncertainty for more ambiguous actions, cf. Sec. 3.4.5 and Appendix A. Next, we detail the self-paced effect on learning.

Uncertainty for Self-paced Learning.

Learning proceeds by minimizing the Poincaré distance via stochastic Riemannian gradient descent [10]. This is based on the Riemannian gradient of Eq. 3.2, computed w.r.t. the online hyperbolic embedding h, which the optimization procedure pushes to match the target $\hat{h}$:

$$\nabla L_{poin}(h, \hat{h}) = \frac{(1 - \|h\|^2)^2}{2\sqrt{(1 - \|h\|^2)(1 - \|\hat{h}\|^2) + \|h - \hat{h}\|^2}}\left(\frac{h - \hat{h}}{\|h - \hat{h}\|} + \frac{h\|h - \hat{h}\|}{1 - \|h\|^2}\right) \tag{3.4}$$

[3]The definition is explicitly adopted in their code. This choice differs from [67] which uses the Poincaré distance, i.e. $u_{\hat{h}} = 1 - d_c(\hat{h}, 0)$. However both definitions provide similar rankings of uncertainty, which matters here. We regard this difference as a calibration issue, which we do not investigate.

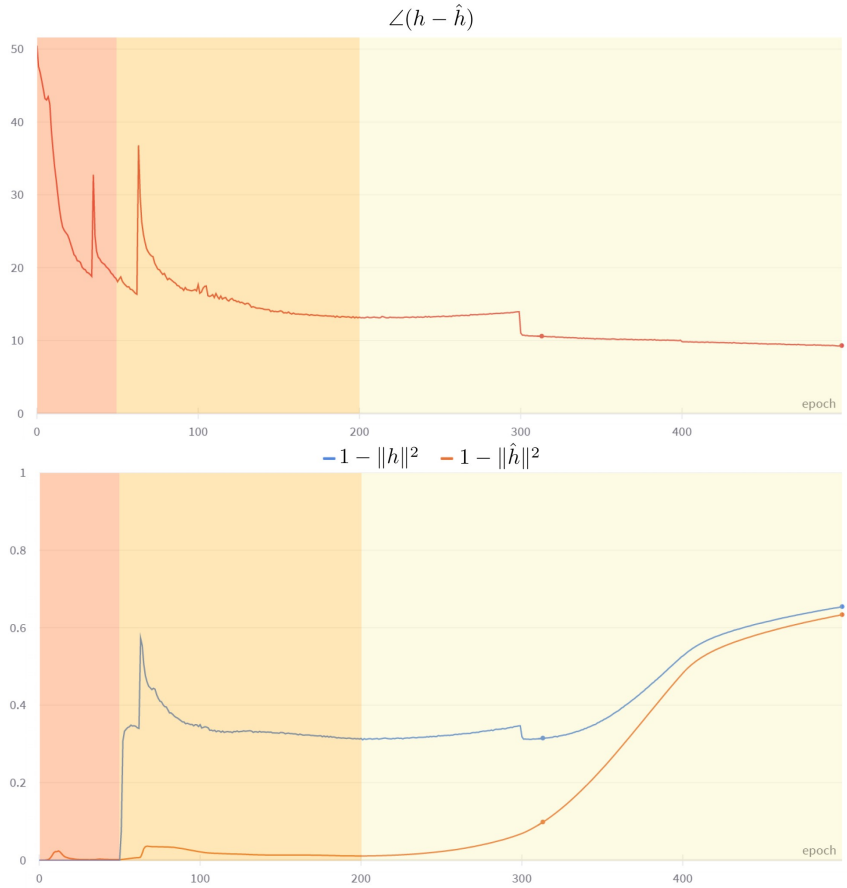

Figure 3.2. Analysis of the parts of eq. 3.4, namely (*top*) the angle between h and $\hat{h}$ and (*bottom*) the quantity $(1 - \|h\|^2)$, monotonic with the embedding uncertainty $(1 - \|h\|)$, during the training phases, i.e. initial (orange), intermediate (yellow), and final (light yellow). See discussion in Sec. 3.3.2.

Learning is *self-paced*, because the gradient changes according to the certainty of the target $\hat{h}$, i.e. the larger their radius $\|\hat{h}\|$, the more certain $\hat{h}$, and the stronger are the gradients $\nabla L_{poin}(h, \hat{h})$, irrespective of h. We further discuss the training dynamics of Eq. 3.4. Let us consider Fig. 3.2, plotting for each training epoch the angle difference between h and $\hat{h}$, namely $\angle(h - \hat{h})$, and respectively the quantities $1 - \|h\|^2$ and $1 - \|\hat{h}\|^2$:

- At the beginning of training (Fig. 3.2 orange section), following random initialization, the embeddings are at the edge of the Poincaré ball $(1 - \|h\|^2, 1 - \|\hat{h}\|^2 \approx 0)$. Assuming that the representations of the two views be still not close $(\|h - \hat{h}\| > 0)$, the gradient $\nabla L_{poin}(h, \hat{h})$ vanishes [56]. The risk of stall is avoided by means of curriculum learning (see Sec. 3.3.2), which considers

only their angle for the first 50 epochs;

- At intermediate training phases (Fig. 3.2 yellow section), the radius of h is also optimized. h departs from the ball edge ($1 - \|h\|^2 > 0$ in Fig. 3.2 *bottom*) and it moves towards $\hat{h}$, marching along the geodesic ($\|h - \hat{h}\| > 0$). At this time, learning is self-paced, i.e. the importance of each sample is weighted according to the certainty of the target view: the higher the radius of $\hat{h}$, the larger the gradient. Note that the online embedding h is pushed towards the ball edge, since $\hat{h}$ is momentum-updated and it moves more slowly away from the ball edge, where it was initialized. This prevents the embeddings from collapsing to trivial solutions during epochs 50-200;
- The training ends (Fig. 3.2 light yellow section) when $\hat{h}$ also departs from the ball edge, due to the updates of the model encoder, brought up by h with momentum. Both h and $\hat{h}$ draws to the ball origin, collapsing into the trivial null solution.

Next, we address two aspects of the hyperbolic optimization: **i.** the use of negatives is not well-defined in the hyperbolic space and, in fact, leveraging them deteriorates the model performance; **ii.** the exponential penalization of the Poincaré distance results in gradients which are too small at early stages, causing vanishing gradient. We address these challenges by adopting relatively positive pairs only [50] and by curriculum learning [9], as we detail next.

Positive-only Learning

The Poincaré distance is minimized when embeddings h and $\hat{h}$ match, both in terms of angle (cosine distance) and radius (uncertainty). Matching uncertainty of two embeddings only really makes sense for positive pairs. In fact, two views of the same sample may well be required to be known to the same degree. However, repulsing negatives in terms uncertainty is not really justifiable, as the sample uncertainty is detached from the sample identity. This has not been considered by prior work [37], but it matters experimentally, as we show in Sec. 3.4.4.
We leverage BYOL [50] and learn from positive pairs only. Referring to Fig. 3.1*(b)*, we add a predictor head q to the online branch, to yield the predicted output embedding p. Next, p is mapped into the hyperbolic space to get the embedding z which is pushed to match the target embedding $\hat{z}$, reference from the stop-gradient

Table 3.1. Results of linear, semi-supervised and finetuning protocols on NTU-60 and NTU-120.

	Linear eval.				Semi-sup. (10%)		Finetune				Additional Techniques				
	NTU-60		NTU-120		NTU-60		NTU-60		NTU-120				Extra	Extra	
Method	*xsub*	*xview*	*xsub*	*xset*	*xsub*	*xview*	*xsub*	*xview*	*xsub*	*xset*	*3s*	*Neg.*	*Aug.*	*Pos.*	*ME*
P&C *[128]*	50.7	76.3	42.7	41.7	-	-	-	-	-	-					
MS^{2}L *[79]*	52.6	-	-	-	65.2	-	78.6	-	-	-		✓			
AS-CAL *[107]*	58.5	64.8	48.6	49.2	-	-	-	-	-	-		✓			
SkeletonCLR *[76]*	68.3	76.4	56.8	55.9	66.9	67.6	80.5	90.3	75.4	75.9		✓			
MCC *[129]*	-	-	-	-	55.6	59.9	83.0	89.7	79.4	80.8		✓			
AimCLR *[52]*	74.3	79.7	63.4	63.4	-	-	-	-	-	-		✓	✓	✓	
ISC *[131]*	76.3	**85.2**	**67.9**	**67.1**	65.9	72.5	-	-	-	-		✓	✓		✓
HYSP *(ours)*	**78.2**	82.6	61.8	64.6	**76.2**	**80.4**	**86.5**	**93.5**	**81.4**	**82.0**			✓		
3s-ST-GCN	-	-	-	-	-	-	85.2	91.4	77.2	77.1	✓				
3s-SkeletonCLR *[76]*	75.0	79.8	-	-	-	-	-	-	-	-	✓	✓			
3s-Colorization *[151]*	75.2	83.1	-	-	71.7	78.9	88.0	94.9	-	-	✓				
3s-CrosSCLR *[76]*	77.8	83.4	67.9	66.7	74.4	77.8	86.2	92.5	80.5	80.4	✓	✓		✓	
3s-AimCLR *[52]*	78.9	83.8	**68.2**	**68.8**	78.2	81.6	86.9	92.8	80.1	80.9	✓	✓	✓	✓	
3s-HYSP *(ours)*	**79.1**	**85.2**	64.5	67.3	**80.5**	**85.4**	**89.1**	**95.2**	**84.5**	**86.3**	✓		✓		

branch, using Eq. 3.2. Note that this simplifies the model and training, as it removes the need for a memory bank and complex negative mining procedures.

Curriculum Learning

The model random initialization yields starting (high-dimensional) embeddings h and $\hat{h}$ with random directions and high norms, which results in vanishing gradients, cf. [56] and the former discussion on training dynamics. We draw inspiration from curriculum learning [9] and propose to initially optimize for the embedding angles only, neglecting uncertainties. The initial training loss is therefore the cosine distance of h and $\hat{h}$. Since the hyperbolic space is conformal with the Euclidean, the hyperbolic and Euclidean cosine distances coincide:

$$L_{cos}(h, \hat{h}) = \frac{h \cdot \hat{h}}{\|h\|\|\hat{h}\|} \tag{3.5}$$

The curriculum procedure may be written as the following convex combination of losses:

$$L_{\text{HYSP}}(h, \hat{h}) = \alpha \; L_{poin}(h, \hat{h}) + (1 - \alpha) \; L_{cos}(h, \hat{h}) \tag{3.6}$$

whereby the weighting term α makes the optimization to smoothly transition from the angle-only to the full hyperbolic objective leveraging angles and uncertainty, after an initial stage. (See Eq. 3.7 in Sec. 3.4.2).

Table 3.2. Results of linear, semi-supervised and finetuning protocols on PKU-MMD I

	Linear eval.				Semi-sup. (10%)				Finetune			
Method	*Joint*	*Bone*	*Motion*	*3s*	*Joint*	*Bone*	*Motion*	*3s*	*Joint*	*Bone*	*Motion*	*3s*
MS^{2}L *[79]*	64.9	-	-	-	70.3	-	-	-	85.2	-	-	-
SkeletonCLR *[76]*	80.9	72.6	63.4	85.3	-	-	-	-	-	-	-	-
ISC *[131]*	80.9	-	-	-	72.1	-	-	-	-	-	-	-
AimCLR *[52]*	83.4	82.0	**72.0**	87.8	-	-	-	86.1	-	-	-	-
HYSP *(ours)*	**83.8**	**87.2**	70.5	**88.8**	**85.0**	**87.0**	**77.8**	**88.7**	**94.0**	**94.9**	**91.2**	**96.2**

3.4 Experiments and Results

Here we present the reference benchmarks (Sec. 3.4.1) and the comparison with the baseline and the state-of-the-art (Sec. 3.4.3); finally we present the ablation study (Sec. 3.4.4) and insights into the learnt skeleton-based action representations (Sec. 3.4.5).

3.4.1 Datasets and Metrics

We consider three established datasets:

NTU RGB+D 60 Dataset [116]. This contains 56,578 video sequences divided into 60 action classes, captured with three concurrent Kinect V2 cameras from 40 distinct subjects. The dataset follows two evaluation protocols: cross-subject (*xsub*), where the subjects are split evenly into train and test sets, and cross-view (*xview*), where the samples of one camera are used for testing while the others for training.

NTU RGB+D 120 Dataset [82]. This is an extension of the NTU RGB+D 60 dataset with additional 60 action classes for a total of 120 classes and 113,945 video sequences. The samples have been captured in 32 different setups from 106 distinct subjects. The dataset has two evaluation protocols: cross-subject (*xsub*), same as NTU-60, and cross-setup (*xset*), where the samples with even setup ids are used for training, and those with odd ids are used for testing.

PKU-MMD [31]. This dataset comprises 1,076 long video sequences across 51 action categories, captured from 66 subjects using the Kinect v2 sensor across three camera views. The evaluation follows a cross-subject protocol, where the subjects are divided into a training group of 57 subjects and a testing group of 9 subjects. The dataset is divided into two subsets: PKU-MMD I and PKU-MMD

II, containing nearly 20,000 and 7,000 action instances, respectively. Both subsets present challenges for action recognition due to the large number of action classes and relatively small training sets. PKU-MMD II is particularly difficult due to greater variation in camera views, leading to increased skeleton noise.

The performance of the encoder f is assessed following the same protocols as [76]:

Linear Evaluation Protocol. A linear classifier composed by a fully-connected layer and softmax is trained supervisedly on top of the frozen pre-trained encoder.
Semi-supervised Protocol. Encoder and linear classifier are finetuned with 10% of the labeled data.
Finetune Protocol. Encoder and linear classifier are finetuned using the whole labeled dataset.
Transfer Learning Protocol. Following [131], we leverage the models pre-trained on the NTU-60 and PKU-MMD I datasets and we finetune them (pre-trained encoder and a classifier jointly) on PKU-MMD II.

3.4.2 Implementation Details

Here we provide some details about the data pipeline and the whole training and testing process.

Data pre-processing Following [76], we pre-process each skeleton sequence by removing all invalid frames and resizing it to the length of 50 frames by linear interpolation. We use two sets of augmentations, normal and extreme. *Normal augmentations* include *Shear* [107] as the spatial and *Crop* [120] as the temporal augmentation. *Extreme augmentations* [52] include four spatial: *Shear, Spatial Flip, Rotate, Axis Mask*, two temporal: *Crop, Temporal Flip*, and two spatio-temporal augmentations: *Gaussian Noise and Gaussian Blur*. Normal augmentations are applied on the target, while extreme ones are applied on the online sequences.

Self-supervised Pre-training The encoder f is ST-GCN [155] with output dimension 1024. Following BYOL [50], the projector and predictor MLPs are linear layers with dimension 1024, followed by batch normalization, ReLU and a final linear layer with dimension 1024. The model is trained with batch size 512 and learning

rate *lr* 0.2 in combination with RiemannianSGD [71] optimizer with momentum 0.9 and weight decay 0.0001. For curriculum learning, across all experiments, we set $e_1 = 50$ and $e_2 = 100$ in Eq. 3.7.

$$\alpha(e) = \begin{cases} 0 & \text{if } e \leq e_1 \\ \frac{e-e_1}{e_2-e_1} & \text{if } e_1 < e < e_2 \\ 1 & \text{if } e \geq e_2 \end{cases} \tag{3.7}$$

Training on 4 Nvidia Tesla A100 GPUs takes approximately 8 hours.

Evaluation In downstream evaluation, the model is trained for 100 epochs using SGD optimizer with momentum 0.9 and weight decay 0. For the linear evaluation, the base *lr* of the classifier is set to 10, dropped by a factor 10 at epochs 60 and 80, keeping the encoder frozen. For semi-supervised and fine-tuned evaluation, the base *lr* of the encoder and classifier is set to 0.1, dropped by a factor 10 at epochs 60 and 80. Exponential mapping is not adopted in evaluation phase.

3.4.3 Comparison to the State of the Art

For all experiments, we used the following setup: ST-GCN [155] as encoder, RiemannianSGD [71] as optimizer, BYOL [50] (instead of MoCo) as self-supervised learning framework, data augmentation pipeline from [52] (See Sec. 3.4.2 for all the implementation details).

NTU-60/120. In Table 3.1, we gather results of most recent SSL skeleton-based action recognition methods on NTU-60 and NTU-120 datasets for all evaluation protocols. We mark in the table the additional engineering techniques that the methods adopt: (*3s*) three stream, using joint, bone and motion combined; (*Neg.*) negative samples with or without memory bank; (*Extra Aug.*) extra extreme augmentations compared to *shear* and *crop*; (*Extra Pos.*) extra positive pairs, typically via nearest-neighbor or cross-view mining; (*ME*) multiple encoders, such as GCN and BiGRU. The upper section of the table reports methods that leverage only information from one stream (joint, bone or motion), while the methods in the lower section use 3-stream information.
As shown in the table, on NTU-60 and NTU-120 datasets using linear evaluation, HYSP outperforms the baseline SkeletonCLR and performs competitive compared to other state-of-the-art approaches. Next, we evaluate HYSP on the NTU-60

Table 3.3. Results of transfer learning on PKU-MMD II, comparing the proposed HYSP against more recent techniques (performance scores other than HYSP are reported from [131]). The evaluation protocol follows [131]: after training an encoder on the source dataset (NTU-60 and PKU-MMD I), the pre-trained encoder and classifier are jointly fine-tuned on a target dataset (PKU-MMD II) for action classification.

	Transfer Learning (PKU-MMD II)	
Method	*PKU-MMD I*	*NTU-60*
S+P *[159]*	43.6	44.8
MS^{2}L *[79]*	44.1	45.8
ISC *[131]*	45.1	45.9
HYSP *(ours)*	**50.7**	**46.3**

dataset using semi-supervised evaluation. As shown, HYSP outperforms the baseline SkeletonCLR by a large margin in both *xsub* and *xview*. Furthermore, under this evaluation, HYSP surpasses all previous approaches, setting a new state-of-the-art. Finally, we finetune HYSP on both NTU-60 and NTU-120 datasets and compare its performance to the relevant approaches. As shown in the table, with the one stream information, HYSP outperforms the baseline SkeletonCLR as well as all competing approaches.

Next, we combine information from all 3 streams (joint, bone, motion) and compare results to the relevant methods in the lower section of Table 3.1. As shown, on NTU-60 dataset using linear evaluation, our 3s-HYSP outperforms the baseline 3s-SkeletonCLR and performs competitive to the current best method 3s-AimCLR. Furthermore, with semi-supervised and fine-tuned evaluation, our proposed 3s-HYSP sets a new state-of-the-art on both the NTU-60 and NTU-120 datasets, surpassing the current best 3s-AimCLR.

PKU-MMD I. In Table 3.2, we compare HYSP to other recent approaches on the PKU MMD I dataset. The trends confirm the quality of HYSP, which achieves state-of-the-art results under all three evaluation protocols.

PKU-MMD II. HYSP outperforms all competing methods in the transfer learning protocol on the PKU-MMD II. Results are reported in Table 3.3, where HYSP surpasses the previous state-of-the-art method by +5.6% when pre-trained on PKU-MMD I and +0.4% if the pre-training dataset is NTU-60.

Table 3.4. Ablation study on NTU-60 (*xview*). Top-1 accuracy (%) is reported.

HYSP	*w/* neg.	*w/o* hyper.	*w/o* curr. learn.	*w/* 3-stream ensemble	hyper. downstream
82.6	69.7	76.9	73.9	85.2	75.5

3.4.4 Ablation Study

Building up HYSP from SkeletonCLR is challenging, since all model parts are required for the hyperbolic self-paced model to converge and perform well. We detail this in Table 3.4 using the linear evaluation on NTU-60 *xview*:

w/ neg. Considering the additional repulsive force of negatives (i.e, replacing BYOL with MoCo) turns HYSP into a contrastive learning technique. Its performance drops by 12.9%, from 82.6 to 69.7. We explain this as due to negative repulsion being ill-posed when using uncertainty, cf. Sec.3.3.2.

w/o hyper. Removing the hyperbolic mapping, which implicitly takes away the self-pacing learning, causes a significant drop in performance, by 5.7%. In this case the loss is the cosine distance, and each target embedding weights therefore equally (no self-pacing). This speaks for the importance of the proposed hyperbolic self-pacing.

w/o curr. learn. Adopting hyperbolic self-pacing from the very start of training makes the model more unstable and leads to lower performance in evaluation (73.9%), re-stating the need to only consider angles at the initial stages.

w/ 3-stream ensemble. Ensembling the 3-stream information from joint, motion and bone, HYSP improves to 85.2%.

Note that all downstream tasks take the SSL-trained encoder f, add to it a linear classifier and train with cross-entropy, as an Euclidean network. Since HYSP pre-trains in hyperbolic space, it is natural to ask whether hyperbolic helps downstream too. A hyperbolic downstream task (linear classifier followed by exponential mapping) yields 75.5%. This is still comparable to other approaches, but below the Euclidean downstream transfer. We see two reasons for this: **i.** in the downstream tasks the GT is given, without uncertainty; **ii.** hyperbolic self-paced learning is most important in pre-training, i.e. the model has supposedly surpassed the information bottleneck [1] upon concluding pre-training, for which a self-paced finetuning is not anymore as effective.

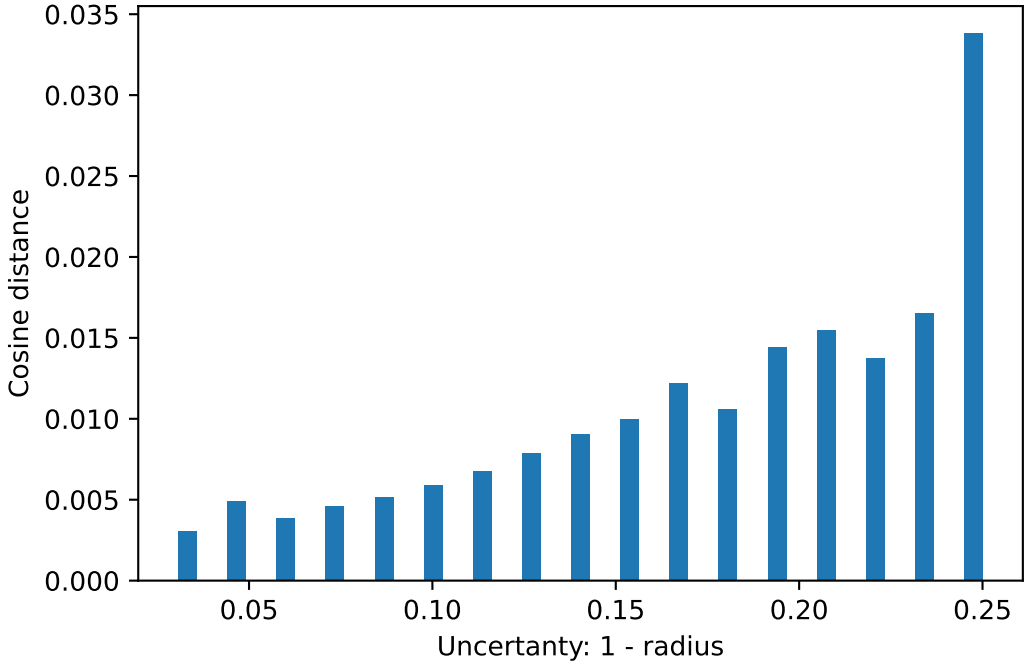

Figure 3.3. Bar plot of the average cosine distance among positive pairs for specific intervals of uncertainty. The plot shows how cosine distance between embeddings and prediction uncertainty after the pre-text task are highly correlated.

3.4.5 More on Hyperbolic Uncertainty

We analyze the hyperbolic uncertainty upon the SSL self-paced pre-training, we qualitatively relate it to the classes, and illustrate motion pictograms, on the NTU-60 dataset.

Uncertainty as a measure of difficulty. We show in Fig. 3.3, the average cosine distance between positive pairs against different intervals of the average uncertainty $(1 - \|h\|)$. The plot shows a clear trend, revealing that the model attributes higher uncertainty to samples which are likely more difficult, i.e. to samples with larger cosine distance.

Inter-class variability. We investigate how uncertainty varies among the 60 action classes, by ranking them using median radius of the class sample embeddings. We observe that: i) actions characterized by very small movements e.g writing, lie at the very bottom of the list, with the smallest radii; ii) actions that are comparatively more specific but still with some ambiguous movements (e.g. touch-back) are almost in the middle, with relatively larger radii; iii) the most peculiar and understandable actions,

(a) Writing (Radius 0.7712)

(b) Touch back (Radius 0.8108)

(c) Standing up (Radius 0.8538)

Figure 3.4. Visualizations of skeleton samples. Uncertainty indicates the difficulty in classifying specific actions if they are characterized by less or more peculiar movements, (a) and (b), or unambiguous actions (c).

either simpler (standing up) or characterized by larger unambiguous movements (handshake), lie at the top of the list with large radii. (See Sec. 3.5 for the complete list).

Sample action pictograms Fig. 3.4 shows pictograms samples from three selected action classes and their class median hyperbolic uncertainty: writing (3.4a), touch back (3.4b), and standing up (3.4c). "Standing up" features a more evident motion and correspondingly the largest radius; by contrast, "writing" shows little motion and the lowest radius.

3.5 Additional Analyses and Visualizations

Here we provide additional visualizations and analysis of the proposed model HYSP. First, we present an analysis of the hyperbolic uncertainty, showing how it varies among different action classes; secondly, we illustrate how it varies within the same action class with sample pictograms; then, we illustrate the relation between action variability and prediction error. Finally, we show through some extra ablation studies what are the effects of batch size and higher/lower uncertainty samples in training on the final performance.

Inter-class variability In support of the discussion in Sec. 3.4.5, in Fig. 3.5 we present the complete list of the 60 action classes from the NTU-60 dataset, ordered according to the median radius of their embeddings[4]. Note that the radius serves as a valid ranking metric for the median action uncertainty, however this work has not explored the calibration of the actual uncertainty value (cf. the discussion on limitations in Sec. 5). Note in the figure how the radius ranks well the level of peculiarity and unambiguous motion in the sequence, e.g. the pictured "Pushing" action (with radius 0.9697) is easier to recognize than "Shake head" (with radius 0.7181).

In Fig. 3.5, actions with smaller movements (e.g. writing, eat meal, playing with phone, or brushing teeth) have smallest radii, while actions with comparatively wider movements (e.g. touch-back, hopping or wearing on glasses) are almost in the middle with relatively larger radii. Finally, actions with the most peculiar and

[4]The median is adopted (instead of the mean) because it is a more robust estimator, less affected by outlier values of radii.

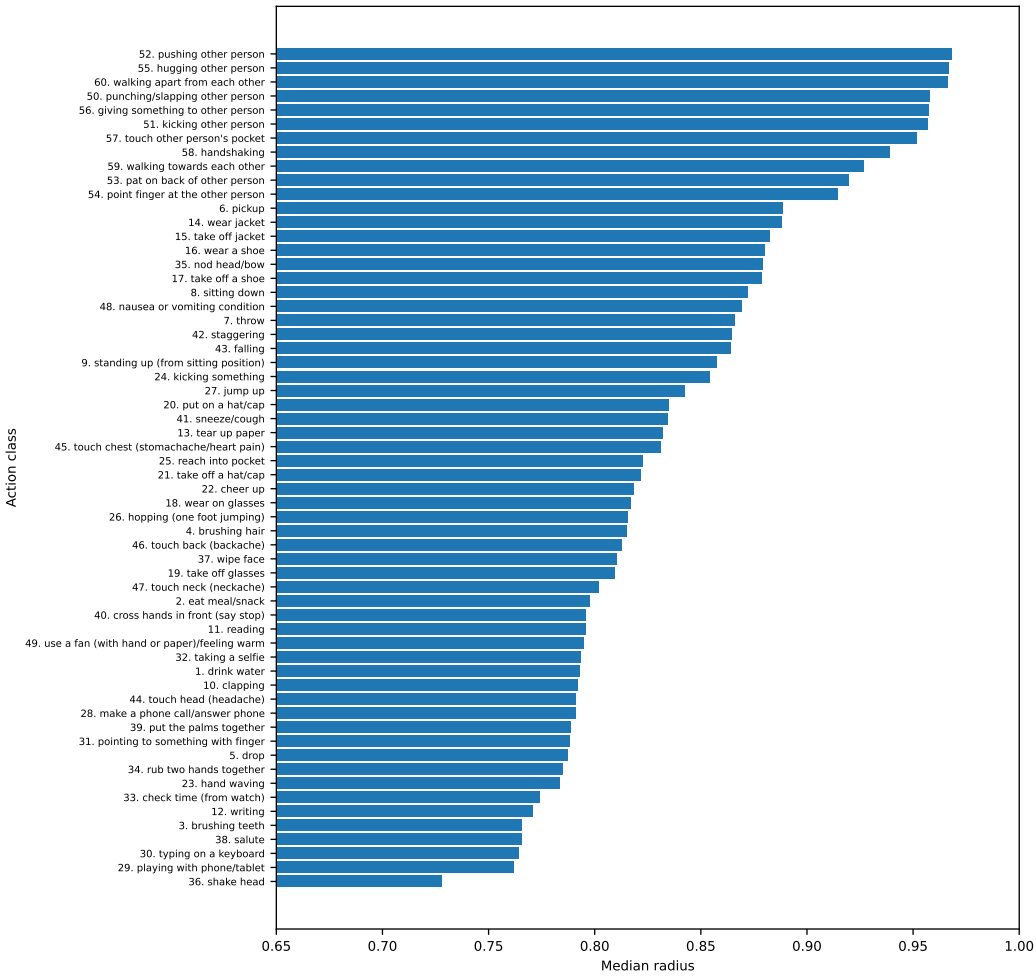

Figure 3.5. Median radius per action class is an indicator of the understandability of each action class (vector graphics, please zoom-in for better readability).

(a) Pushing (Radius 0.9697) **(b)** Hugging (Radius 0.9672) **(c)** Walking apart (Radius 0.9692)

(d) Take off hat (Radius 0.8143) **(e)** Cheer up (Radius 0.8091) **(f)** Standing up (Radius 0.8538)

(g) Playing with phone (Radius 0.7679) **(h)** Writing (Radius 0.7721) **(i)** Shake head (Radius 0.7181)

Figure 3.6. Visualization of inter-class variability through skeleton sequences with high (a,b,c), average (d,e,f), and low (g,h,i) median radius.

unambiguous movements, either simpler (standing up, throw) or involving two people (handshake, walking apart), lie at the top of the list with large radii. Fig. 3.6 shows the pictograms of some selected action classes. For each action, we have selected a representative sequence (with its radius close to the median of the class).

Intra-class variability We complement the above paragraph with qualitative examples of intra-class variability, i.e. how the action embedding radius (as a proxy to its uncertainty) varies for sequences within the same class.

As illustrated in Fig. 3.7, with reference to actions "Touch neck" (a, b, c) and "Staggering" (d, e, f), some sequences are more ambiguous (3.7a, 3.7d), some are a little bit clearer (3.7b, 3.7e) and, finally, others are more specific and easier to represent by the model (3.7c, 3.7f).

- 0. drink water.
- 1. eat meal/snack.
- 2. brushing teeth.
- 3. brushing hair.
- 4. drop.
- 5. pickup.
- 6. throw.
- 7. sitting down.
- 8. standing up (from sitting position).
- 9. clapping.
- 10. reading.
- 11. writing.
- 12. tear up paper.
- 13. wear jacket.
- 14. take off jacket.
- 15. wear a shoe.
- 16. take off a shoe.
- 17. wear on glasses.
- 18. take off glasses.
- 19. put on a hat/-cap.
- 20. take off a hat/-cap.
- 21. cheer up.

(a) Touch neck (Radius 0.722) **(b)** Touch neck (Radius 0.7976) **(c)** Touch neck (Radius 0.9554)

(d) Staggering (Radius 0.7725) **(e)** Staggering (Radius 0.873) **(f)** Staggering (Radius 0.9754)

Figure 3.7. Visualization of intra-class variability through skeleton sequences of the same class having low (a,d), average (b,e), and high (c,f) radius.

- 22. hand waving.
- 23. kicking something.
- 24. reach into pocket.
- 25. hopping (one foot jumping).
- 26. jump up.
- 27. make a phone call/answer phone.
- 28. playing with phone/tablet.
- 29. typing on a keyboard.
- 30. pointing to something with finger.
- 31. taking a selfie.
- 32. check time (from watch).
- 33. rub two hands together.
- 34. nod head/bow.
- 35. shake head.
- 36. wipe face.
- 37. salute.
- 38. put the palms together.
- 39. cross hands in front (say stop).
- 30. sneeze/cough.
- 41. staggering.
- 42. falling.
- 43. touch head (headache).
- 44. touch chest (stomachache/heart pain).
- 45. touch back (backache).
- 46. touch neck (neckache).
- 47. nausea or vomiting condition.
- 48. use a fan (with hand or paper)/feeling warm.
- 49. punching/slapping other person.
- 50. kicking other person.
- 51. pushing other person.
- 52. pat on back of other person.
- 53. point finger at the other person.
- 54. hugging other person.
- 55. giving something to other person.
- 56. touch other person's pocket.
- 57. handshaking.
- 58. walking towards each other.
- 59. walking apart from each other.

Relation between intra-class variability and prediction error We complement the study on action variability and uncertainty by relating those to the prediction error. In Fig. 3.8, we report the confusion matrix, whereby actions have been arranged by the median radius of the class embeddings. So, actions in the rows and columns are sorted according to their median radii (increasing left-to-right and top-to-bottom), using the values from Fig. 3.5. One observes that lower radii (bottom right corner) yield larger prediction errors among the actions. This is the case for actions such as writing (#11) and brushing teeth (#2), i.e. these are more ambiguous/uncertain because they are characterized by less distinct motion. (The action IDs are illustrated after the confusion matrix.)

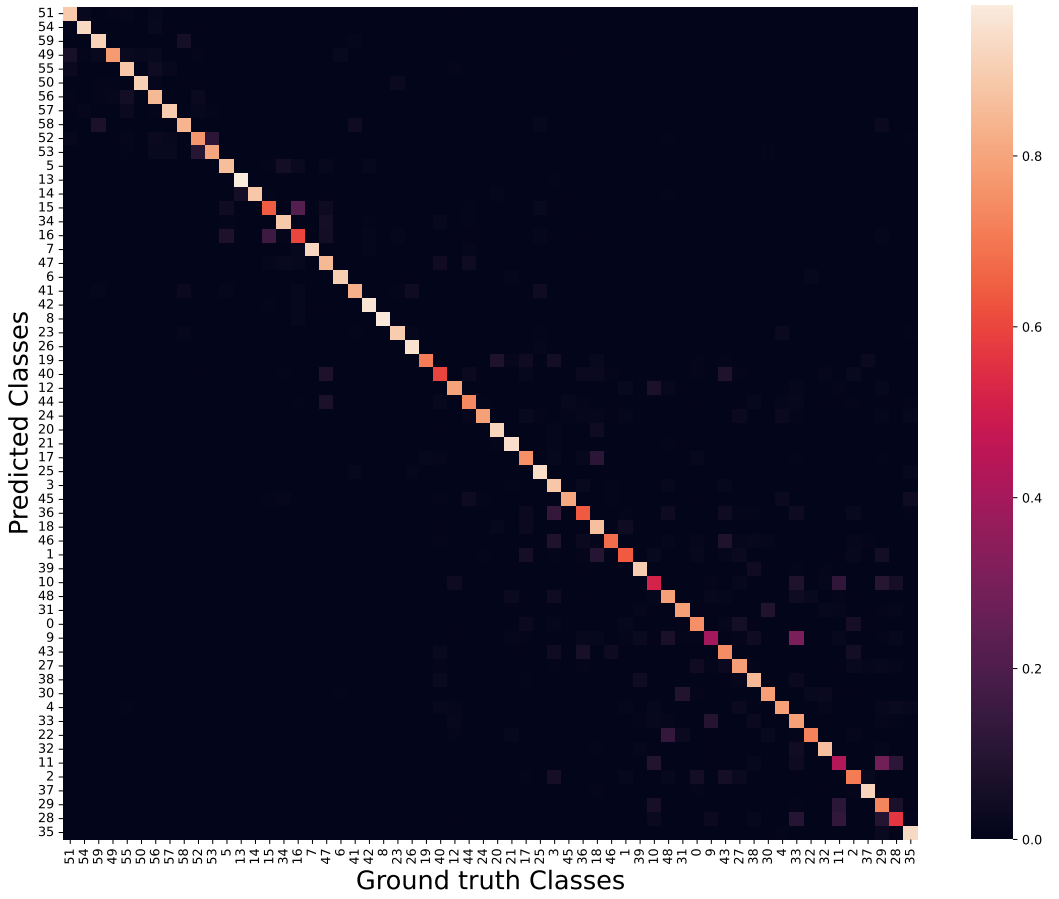

Figure 3.8. Confusion matrix among the actions of NTU60 (*xview*, single stream input, linear test protocol, top-1 accuracy) arranged by decreasing order of the median radii of their class sample embeddings (values from Fig. 3.5). Actions with lower radii (bottom-right corner) result in larger prediction errors. Those actions (IDs listed above) are also more ambiguous and characterized by less distinct motions.

Additional insights into the self-paced learning In Fig. 3.9, we provide further insights into Eq. 3.4, the Riemannian gradient of the Poincaré loss, by illustrating the relation between the gradient magnitude and the uncertainty of the target-view embedding $(1 - \|\hat{h}\|)$. We consider the model checkpoint for epoch 100, which belongs to the second (the main) stage of training (the analysis of other epochs has produced similar statistics). Then we compute aggregate histogram statistics across all the training samples. The plot of Fig. 3.9 confirms the analysis of Sec. 3.3.2: the lower the uncertainty of the target view embedding $\hat{h}$ (left side of the plot), the larger the gradient. As a result, learning is self-paced because the importance of each sample is weighted, via the respective gradient magnitude,

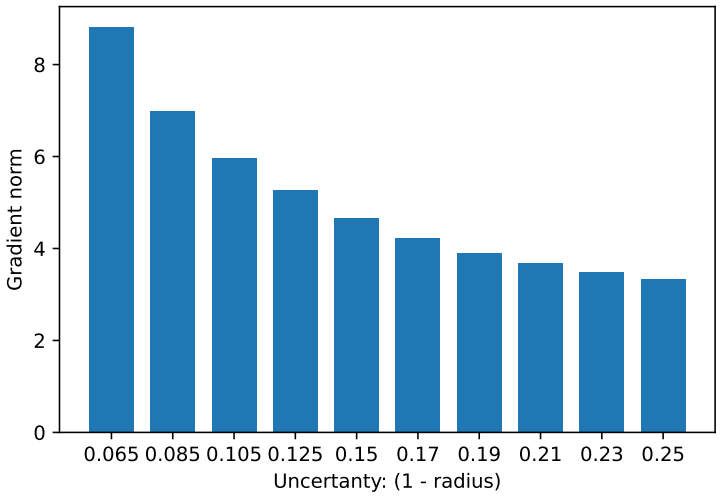

Figure 3.9. Bar plot of the average Riemannian gradient norm for specific intervals of uncertainty. The plot experimentally explains the intuition behind Eq. 5 described in Sec 3.2.1. The gradient norm and prediction uncertainty after the pre-text task are highly correlated. Learning them is *self-paced*, because lower the uncertainty of $\hat{h}$ and the stronger is the gradient.

Table 3.5. Top-1 accuracy varying batch size for the linear protocol. The dataset used in this table is NTU-60 xview.

Batch size	512	256	128	64	**32**
Top-1 Acc.	78.3	79.9	81.2	82.2	**82.6**

according to the level of certainty of the target-view embedding.

Effect of batch size on linear protocol Here we report an interesting finding relating to the batch-size hyper-parameter when testing with the linear protocol. We find that a lower batch size (32) results in a considerable performance increase with respect to a larger one (512). For the NTU-60, *xview*-subset, the top-1 accuracy grows from 78.3 to 82.6, resulting in a performance increase of +4.3%. For this test, Table 3.5 reports performances of various batch sizes.

Effect of higher/lower uncertainty samples on training We experiment to compare the performance variation between using harder positives only (i.e. the samples with higher uncertainty) vs. easier positives only (those samples with lower uncertainty). For the sake of comparison, we select an equal number of hard and

easy positives, namely the first and second half of the total samples, after sorting them in decreasing order of uncertainty (as estimated upon the HYSP pre-training on all samples). It turns out that learning with the harder examples yields the top-1 accuracy of 70.9%, while the easier ones yield 74.5%. It is expected that both results are well below the performance of the model trained on the full dataset (82.6% with the joint stream), because of using half the amount of data. As for the performance discrepancy, we speculate that easier samples have an edge on the half-sized training set, while harder samples better support when larger amount of data is available. Let us also add that, as expected, the two new trainings have different durations: that one with harder samples converges and overfits earlier (at epochs 400 and 500 respectively) than with easier (epochs 400 and 700 correspondingly).

3.6 Discussion

This work has proposed the first hyperbolic self-paced model HYSP, which re-interprets self-pacing with self-regulating estimates of the uncertainty for each sample. Both the model design and the self-paced training are the result of thorough considerations on the hyperbolic space. As current limitations, we have not explored the calibration of uncertainty, nor tested HYSP for the image classification task, because modeling has been tailored to the skeleton-based action recognition task and the baseline SkeletonCLR, which yields comparatively faster development cycles. Exploring the overfitting of HYSP at the latest stage of training lies for future work. This is now an active research topic [37, 57, 56] for hyperbolic spaces.

Chapter 4

Hyperbolic Active Learning for Semantic Segmentation under Domain Shift

4.1 From HYSP to HALO: Expanding Hyperbolic Applications

The promising results achieved by harnessing the hyperbolic radius in HYSP's learning process naturally led us to explore its potential in other domains. We saw an opportunity to apply this concept to the challenging field of semantic segmentation under domain shift. This insight gave birth to HALO (Hyperbolic Active Learning Optimization), where we reimagined the hyperbolic radius as a proxy for data scarcity. By combining this novel interpretation with prediction entropy, HALO introduces an innovative approach to estimating epistemic uncertainty for active learning in pixel-level tasks. This evolution from skeleton-based action recognition to semantic segmentation not only showcases the adaptability of hyperbolic embeddings across diverse computer vision challenges but also illustrates how a fundamental concept can be creatively repurposed to address distinct machine learning problems.

4.2 Overview

Dense prediction tasks, such as semantic segmentation (SS), are important in applications such as self-driving cars, manufacturing, and medicine. However, these tasks necessitate pixel-wise annotations, which can incur substantial costs and time inefficiencies [32]. Previous methods [145, 135, 119, 118, 95] have addressed this labeling challenge via domain adaptation, capitalizing on large source datasets for pre-training and domain-adapting with few target annotations [8]. Most recently, active domain adaptation (ADA) has emerged as an effective strategy, i.e. annotating only a small set of target pixels in successive labelling rounds [95]. State-of-the-art (SOTA) ADA [119, 140, 145] relies on the entropy of predicted pseudo-labels, which they define as prediction uncertainty, as the core strategy for active learning (AL) data acquisition. In fact, prediction uncertainty correlates well with the likelihood of pixel classification mistakes, but it is only one of the factors of the overall model uncertainty, as we argue in this work.

Following extensive literature [34, 66, 62, 133], we distinguish aleatoric and epistemic uncertainty, and we propose the second for the data acquisition strategy in active learning. Epistemic uncertainty is an indicator of the state of knowledge about the task. This uncertainty stems not only from inaccuracies, as identified by prediction uncertainty, but also from the information the model has been exposed to thus far, including the amount of data considered. In the domain adaptation task, the domain gap arises from the model's lack of understanding of the new domain data, akin to the definition of epistemic uncertainty. Building upon this intuition, we propose HALO (Hyperbolic Active Learning Optimization), a novel approach for active domain adaptation, where we introduce the use of epistemic uncertainty into the data acquisition strategy. Our in-depth analysis shows that the hyperbolic radius effectively estimates data scarcity, revealing it as a key component in the estimation of epistemic uncertainty. The combination of the radius with a complementary information signal such as prediction entropy offers a comprehensive estimate of epistemic uncertainty.

Interpreting the hyperbolic radius as a proxy to data scarcity diverges from known interpretations in the hyperbolic literature. The SOTA hyperbolic SS model [5] trains with class hierarchies, which they manually define. As a result, their hyperbolic radius represents the parent-to-child hierarchical relations in the Poincaré ball. We adopt [5] without enforcing hierarchical labels and we find that hierarchical

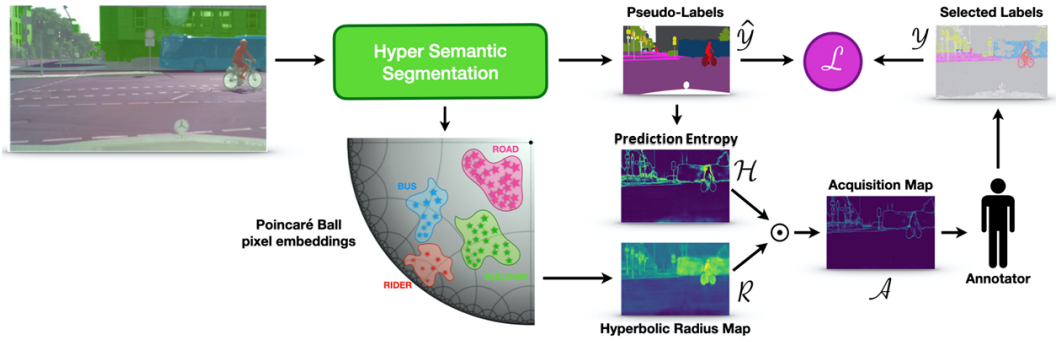

Figure 4.1. Overview of HALO. Pixels are encoded into the hyperbolic Poincaré ball and classified in the pseudo-label $\hat{\mathcal{Y}}$. The hyperbolic radius of the pixel embeddings defines the new hyperbolic score map $\mathcal{R}$. The prediction entropy $\mathcal{H}$ is extracted as the entropy of the softmax probabilities. Combining $\mathcal{R}$ and $\mathcal{H}$ we define the data acquisition score map $\mathcal{A}$, which is used to query new labels $\mathcal{Y}$.

relationships do not emerge naturally in our case. For instance, in HALO, classes such as *road* and *building* are closer to the center of the ball, while *person* and *rider* have larger radii. This class arrangement contradicts the interpretation of the hyperbolic radius as a proxy for uncertainty, which emerged from metric learning hyperbolic studies [37, 41]. In our context, larger radii indicate larger data scarcity, therefore less certainty, which is in contrast with [41]'s interpretation. Thus, our interpretation of the hyperbolic radius as a proxy for data scarcity does not align with neither of the existing interpretations in the case of hierarchy-free hyperbolic SS. Consider HALO's pipeline illustrated in Fig. 4.1 and the circular sector representing the Poincaré ball, where pixels from various classes are mapped. The hyperbolic model assigns a higher radius to classes that appear less frequently in the dataset (e.g., *rider*), and a lower radius to classes which are more frequent (e.g., *road*). In Sec. 4.5, we show how this novel interpretation of the hyperbolic radius arises bottom-up from data statistics.

We demonstrate the effectiveness of our approach through extensive benchmarking on well-established datasets for SS, including ADA from GTAV to Cityscapes, SYNTHIA to Cityscapes, and additional testing on Cityscapes to ACDC under adverse weather conditions. HALO sets a new SOTA on all the benchmarks (+3.3% on GTA→CS, +4.2% on SYNTHIA→CS, and +2.9% on CS→ACDC). Moreover, this is the first AL method that surpasses the supervised domain adaptation baseline using only a small portion of labels (+2.6% on GTA→CS with 5% budget). Our paper also introduces a novel technique to enhance the stability of hyperbolic training,

referred to as *Hyperbolic Feature Reweighting* (HFR), cf. Sec. 4.6. Our code will be released.

In summary, our contributions include: 1) Presenting a novel interpretation of the hyperbolic radius as a proxy for data scarcity and its relationship with epistemic uncertainty; 2) Introducing hyperbolic neural networks in AL and a novel pixel-based data acquisition score based on the hyperbolic radius; 3) Validating both the concept and the algorithm through a comprehensive analysis, achieving a new state-of-the-art performance across all the considered ADA benchmarks for SS. Our method surpasses for the fist time in AL the supervised DA performance using only a small percentage of labels.

4.3 Related Work

4.3.1 Hyperbolic Representation Learning (HRL)

Hyperbolic geometry has been extensively used to capture embeddings of tree-like structures [94, 17] with low distortion [112, 114]. Since the seminal work of [45] on Hyperbolic Neural Networks (HNN), approaches have successfully combined hyperbolic geometry with model architectures ranging from convolutional [117] to attention-based [51], including graph neural networks [84, 18] and, most recently, vision transformers [37]. There are two leading interpretations of the hyperbolic radius in hyperbolic space: as a measure of the prediction uncertainty [20, 37, 41, 39] or as the hierarchical parent-to-child relation [94, 132, 130, 37, 5]. Our work builds on the SOTA hyperbolic semantic segmentation method of [5], which enforces hierarchical labels and training objectives. However, when training without manually injected hierarchical labels, as we do, the hierarchical interpretation does not apply. Although a correlation between the hyperbolic radius and an uncertainty measure has been noted, a comprehensive understanding of this relationship is still lacking. In order to further research in this direction, we provide an investigation that examines the relationship between the hyperbolic radius, data scarcity, and epistemic uncertainty, aiming to shed light on this association. Furthermore, HALO's acquisition score is tailored for semantic segmentation, as it computes the hyperbolic radius for each pixel embedding. Hyperbolic neural networks have shown comparable performance to Euclidean models in semantic segmentation [5], enabling fair comparisons. However, this equivalence does not extend to other tasks, where hyperbolic neural

networks have not achieved similar performance (image classification) or are yet to be developed (object detection).

4.3.2 Active Learning (AL)

The number of annotations required for dense tasks such as semantic segmentation can be costly and time-consuming. Active learning balances the labeling efforts and performance, selecting the most informative pixels in successive learning rounds. Strategies for active learning are based on uncertainty sampling [44, 136, 137], diversity sampling [4, 70, 115, 141] or a combination of both [124, 144, 105, 145]. For the case of AL in semantic segmentation, EqualAL [48] incorporates the self-supervisory signal of self-consistency to mitigate the overfitting of scenarios with limited labeled training data. Labor [119] selects the most representative pixels within the generation of an inconsistency mask. PixelPick [118] prioritizes the identification of specific pixels or regions over labeling the entire image. [92] explores the effect of data distribution, semi-supervised learning, and labeling budgets. We are the first to leverage the hyperbolic radius as a proxy for the most informative pixels to label next.

4.3.3 Active Domain Adaptation (ADA)

Domain Adaptation (DA) involves learning from a source data distribution and transferring that knowledge to a target dataset with a different distribution. Recent advancements in DA for semantic segmentation have utilized unsupervised (UDA) [59, 135, 152, 87, 89, 85] and semi-supervised (SSDA) [42, 110, 121, 65] learning techniques. However, challenges such as noise and label bias still pose limitations on the performance of DA methods. Active Domain Adaptation (ADA) aims to reduce the disparity between source and target domains by actively selecting informative data points from the target domain [127, 43, 123, 119], which are subsequently labeled by human annotators. In semantic segmentation, [95] propose a multi-anchor strategy to mitigate the distortion between the source and target distributions. The recent study of [145] shows the advantages of region-based selection in terms of region impurity and prediction uncertainty scores, compared to pixel-based approaches. By contrast, we show that selecting pixels just from class boundaries limits performance, as they are not necessarily the most informative, as

we confirm with an oracular study. Instead, we show that the hyperbolic radius, in conjunction with prediction entropy, effectively approximates epistemic uncertainty, thereby serving as a successful objective for label acquisition.

4.3.4 Uncertainty

The notion of uncertainty has gained increasing attention in machine learning (ML) research in recent years, primarily due to its growing practical significance in real-world applications. Consequently, numerous studies have developed approaches for uncertainty quantification in ML [66, 15, 143, 91]. Within the existing literature, two distinct sources of uncertainty are commonly acknowledged: aleatoric and epistemic [38, 60]. Aleatoric uncertainty stems from the inherent randomness and variability within the data, while epistemic uncertainty arises from a lack of knowledge or data. As a result, epistemic uncertainty can theoretically be reduced with supplementary information, while aleatoric uncertainty remains non-reducible. Several methodologies have proposed techniques for quantifying both aleatoric and epistemic uncertainty [34, 66, 62, 133]. Following the approach of [34], both total uncertainty and aleatoric uncertainty can be approximated via model ensemble [74], deriving epistemic uncertainty as the difference between the two. In our study, for the first time, we distinguish two leading complementary causes for epistemic uncertainty: prediction error and data scarcity.

4.4 Background

In this section, we introduce the background for our work. We begin by discussing Hyperbolic Neural Networks and Hyperbolic Semantic Segmentation, moving then to Active Domain Adaptation, which form the basis of our approach.

4.4.1 Hyperbolic Neural Networks and Semantic Segmentation

We operate in the Poincaré ball hyperbolic manifold, as defined in Section 2.1. We define it as the pair $(\mathbb{D}_c^N, g^{\mathbb{D}_c})$ where $\mathbb{D}_c^N = \{x \in \mathbb{R}^N : c\|x\| < 1\}$ is the manifold and $g_x^{\mathbb{D}_c} = (\lambda_x^c)^2 g^{\mathbb{E}}$ is the associated Riemannian metric, $-c$ is the curvature, $\lambda_x^c = \frac{2}{1-c\|x\|^2}$ is the conformal factor and $g^{\mathbb{E}} = \mathbb{I}^N$ is the Euclidean metric tensor.

Hyperbolic neural networks first extract a feature vector v in Euclidean space, which is subsequently projected into the Poincaré ball via exponential map (cf. Eq. 2.3).

We define the hyperbolic radius of the embedding $h \in \mathbb{D}_c^N$ as the Poincaré distance (Eq. 2.8) of h from the origin of the ball (cf. Sec. 2.1.1 and Eq. 2.9).

For the hyperbolic semantic segmentation, we adopt the work by [5], which stands as the first to showcase performance comparable to that of Euclidean networks. Segmentation is performed using hyperbolic multinomial logistic regression (HyperMLR; see Sec. 2.2.3 and Eq. 2.14) [45].

4.4.2 ADA for Semantic Segmentation

The task of ADA aims to transfer knowledge from a source labeled dataset $\mathcal{S} = (X_s, Y_s)$ to a target unlabeled dataset $\mathcal{T} = (X_t, Y_t)$, where X represents an image and Y the corresponding annotation map. Y_s is given, Y_t is initially the empty set $\emptyset$. Adhering to the ADA protocol [145, 140, 119], target annotations are incrementally added in rounds, subject to a predefined budget, upon querying an annotator. Each pixel is assigned a priority score using a predefined acquisition map $\mathcal{A}$. Labels are added to Y_t in each AL round by selecting pixels from $\mathcal{A}$ with higher scores, in accordance with the budget. Each AL round is divided into two phases. In the first phase, the segmentation model undergoes end-to-end training, with back-propagation incorporating estimates $\hat{Y}_s$ and $\hat{Y}_t$ from the per-pixel cross-entropy loss $\mathcal{L}(\hat{Y}_s, \hat{Y}_t, Y_s, Y_t)$. The second phase consists in acquiring new target labels according to the acquisition score $\mathcal{A}$ and the predefined budget.

In Sec. 4.5, we assume to have pre-trained the hyperbolic image segmenter of [5] on the source dataset GTAV [108] and to have domain-adapted it to the target dataset Cityscapes [32] through 5 rounds of AL with a total budget of 5%. The following analyses consider the radii of the hyperbolic pixel embeddings and the prediction entropy, for which statistics are computed on the Cityscapes validation set.

4.5 Hyperbolic Radius and Epistemic Uncertainty

In Sec. 4.5.1 we interpret the emerging properties of hyperbolic radius, and we compare with the interpretations in literature in Sec. 4.5.2.

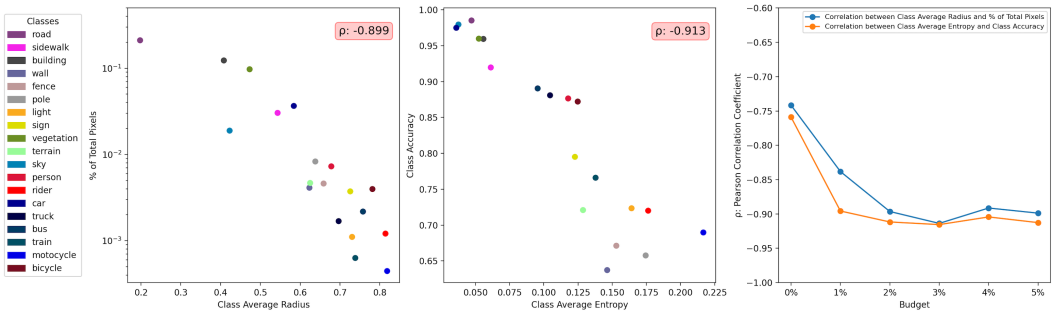

Figure 4.2. (left) Plot of class average radius vs. the percentage of total pixels in the target dataset; (center) Plot of the class average entropy vs. class accuracy; (right) Plot of labeling budget vs. correlation between class average radius and percentage of total pixels (blue) and between class average entropy and class accuracy (orange).

4.5.1 Emerging properties of the hyperbolic radius

What does the hyperbolic radius represent? Fig. 4.2 (left) shows the correlation between the average class hyperbolic radius and the percentage of pixel labels for each class relative to the total number of pixels in the dataset. The correlation is substantial ($\rho = -0.899$), so classes with larger hyperbolic radii such as *motocycle* are rarer in the target dataset, while at lower hyperbolic radii we have more frequent classes such as *road*. In conclusion, larger hyperbolic radii indicate which classes the model has been exposed less so far in the training.

Understanding the role of the prediction entropy Prior active learning literature [144, 105, 145] agree on the utility of prediction entropy, i.e. the entropy of the prediction scores, in the data acquisition strategy. In Fig. 4.2 (center) we report the correlation between the class average entropy and the class accuracy, whose resulting value is a strong correlation of $\rho = -0.913$. In conclusion, prediction entropy appears to be a good indicator for classes with low accuracy. In HALO, we combine prediction entropy with the newly proposed hyperbolic radius.

How does learning the hyperbolic manifold proceed? Fig. 4.2 (right) illustrates the evolution, across active learning rounds, of the correlation between prediction entropy and accuracy (orange), and the correlation between the hyperbolic radius and the percentage of pixels in the target dataset (blue). Both correlations exhibit a growing trend in module, eventually saturating at high values. In conclusion, as the training progresses, both the hyperbolic radius and the prediction entropy become better estimators for data scarcity and prediction error.

Relation with the epistemic uncertainty Following the work of [34], we quantify the epistemic uncertainty as the difference between total and aleatoric uncertainty. The total uncertainty is estimated by computing the entropy of the predictive posterior distribution $U_t(\boldsymbol{x}) = \mathcal{H}[p(y|\boldsymbol{x})]$. This formulation encompasses the epistemic uncertainty regarding the network parameters $\boldsymbol{\theta}$. To compute it, we first measure the aleatoric uncertainty as $U_a(\boldsymbol{x}) = E_{p(\boldsymbol{\theta}|\mathcal{D})}\mathcal{H}[p(y|\boldsymbol{\theta}, \boldsymbol{x})]$ and then we derive the epistemic uncertainty as the difference $U_e(\boldsymbol{x}) = U_t(\boldsymbol{x}) - U_a(\boldsymbol{x})$. The model ensemble approach [74] offers an effective means to approximate the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ using a finite ensemble of models $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M$. We can approximate the total uncertainty as

$$\tilde{U}_t(\boldsymbol{x}) = -\sum_{y\in\mathcal{Y}} \left(\frac{1}{M}\sum_{m=1}^{M} p(y|\boldsymbol{\theta}_m, \boldsymbol{x})\right) log_2 \left(\frac{1}{M}\sum_{m=1}^{M} p(y|\boldsymbol{\theta}_m, \boldsymbol{x})\right) \tag{4.1}$$

and similarly the aleatoric uncertainty

$$\tilde{U}_a(\boldsymbol{x}) = -\frac{1}{M}\sum_{m=1}^{M}\sum_{y\in\mathcal{Y}} p(y|\boldsymbol{\theta}_m, \boldsymbol{x})\ log_2\ p(y|\boldsymbol{\theta}_m, \boldsymbol{x}). \tag{4.2}$$

Finally the epistemic uncertainty is approximated by the difference $\tilde{U}_e(\boldsymbol{x}) = \tilde{U}_t(\boldsymbol{x}) - \tilde{U}_a(\boldsymbol{x})$.

The correlation between the epistemic uncertainty and the hyperbolic radius results in a value of $\rho = 0.769$, while the correlation between the epistemic uncertainty and the prediction entropy is $\rho = 0.789$. Supported by the fact that the correlation between the hyperbolic radius and the prediction entropy is moderate ($\rho = 0.658$), we conclude that they encode complementary signals for the uncertainty description, respectively data scarcity and prediction error. In fact, the correlation between their product and epistemic uncertainty results in an even higher value ($\rho = 0.824$). Building upon this observation, we establish the acquisition score as the product of these two metrics (see Sec. 4.6.2).

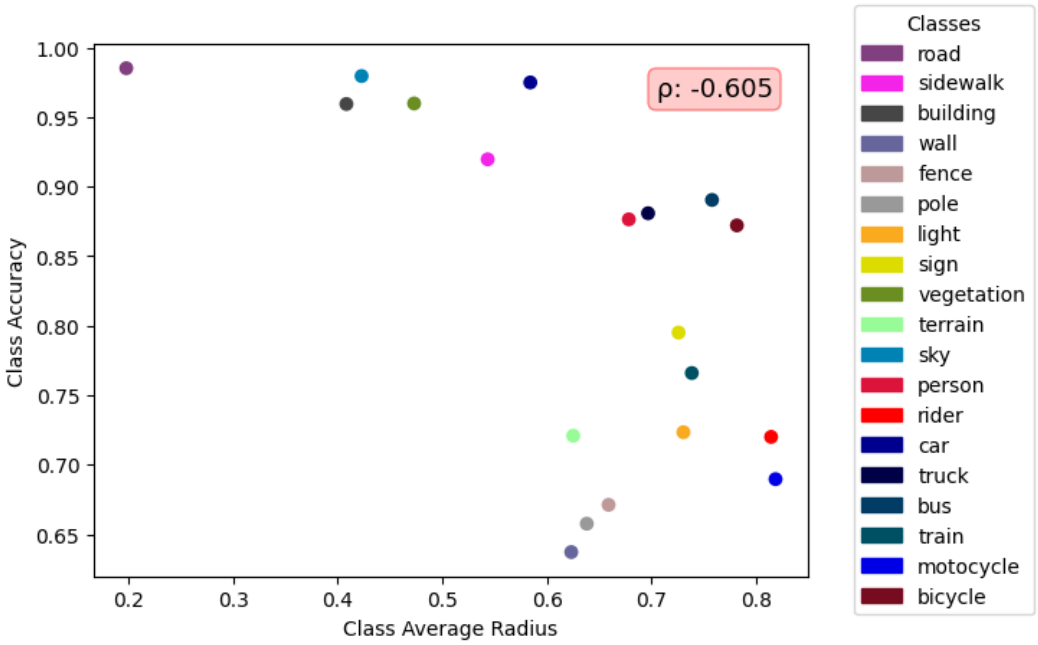

Figure 4.3. Plot of the class average radius vs. class accuracy.

4.5.2 Comparing interpretations of the hyperbolic radius

It emerges from our analysis that larger radii are assigned to classes that have higher data scarcity. Earlier work has explained the hyperbolic radius in terms of uncertainty or hierarchies. Approaches from the former [41, 39] suggest that larger hyperbolic radii indicate more certain and unambiguous samples in terms of classification accuracy. In our case, the correlation between the hyperbolic radius and class accuracy, as depicted in Fig. 4.3, is moderate ($\rho = -0.605$). However, this value is considerably lower than the correlation between the hyperbolic radius and the percentage of pixels. Hence, the radius serves as a more effective indicator of data scarcity (see appendix 4.8.1 for additional analysis). Another difference with the studies in favor of the uncertainty interpretation lies in the definition of the uncertainty as $1 - \text{radius}$, typical of hyperbolic metric learning-based approaches [41, 39]. In those, a larger radius leads to an exponentially larger matching penalty due to the employed Poincaré distance, effectively making the radius inversely proportional to the errors, as those studies show.

Elsewhere, the interpretation of the hyperbolic radius aligns with a hierarchical explanation [94, 132, 130, 37, 5]. These methods involve hierarchical datasets, hierarchical labeling, and classification objective functions. Hierarchies naturally

align with the growing volume in the Poincaré ball, resulting in children nodes from different parents being mapped further from each other than from their parents. Learning under hierarchical constraints results in leaf classes closer to the edge of the ball, and transitions between them traverse their parent nodes at lower hyperbolic radii. Our hyperbolic segmentation approach differs from prior hyperbolic works [5, 41, 39, 37] as we employ hyperbolic multinomial logistic regression without the incorporation of hierarchical labels or losses based on the Poincaré distance. These differences drive our intuition to utilize the hyperbolic radius as an estimator of data scarcity, thereby incorporating it into the final data acquisition score in the active learning process.

4.6 Hyperbolic Active Learning Optimization (HALO)

In this section, first we introduce the proposed HALO pipeline (Sec. 4.6.1), then we detail the novel acquisition strategy (Sec. 4.6.2). Finally, we present our proposition for fixing the hyperbolic training instability (Sec. 4.6.3).

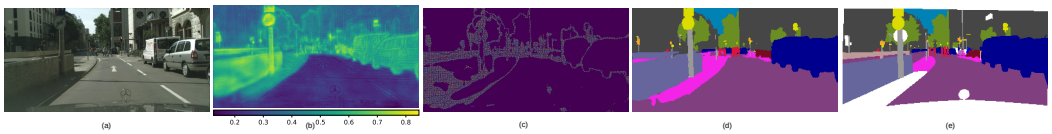

Figure 4.4. (a) Original image; (b) Radius map depicting the hyperbolic radii of pixel embeddings; (c) Pixels (yellow) that have been selected for data acquisition. See Sec. 4.6 for details; (d) HALO prediction; (e) Ground Truth annotations. Zoom in for the details.

4.6.1 HALO pipeline

Let us consider Fig. 4.1. Our AL strategy consists in assigning an acquisition score $\mathcal{A}$ to each pixel, based on the combination of hyperbolic radius and prediction entropy. We estimate the hyperbolic radius $\mathcal{R}$ from pixel embeddings (as detailed in Sec. 4.5 and illustrated in Fig. 4.4b). Concurrently, predicted classification probabilities are used to compute the prediction entropy $\mathcal{H}$, inspired by prior works [101, 118, 136, 137, 145]. New labels are subsequently chosen based on the acquisition score $\mathcal{A}$ and integrated into the training set (see selected pixels in Fig. 4.4c). Note that the new labels are both at the boundaries and within, in areas with the largest inaccuracies (compare Fig. 4.4d and 4.4e).

4.6.2 Novel data acquisition strategy

The acquisition score of each pixel in an image is formulated as the element-wise multiplication of the hyperbolic radius map $\mathcal{R}$ and the prediction entropy map $\mathcal{H}$, i.e. $\mathcal{A} = \mathcal{R} \odot \mathcal{H}$. The radius is computed as the distance of the hyperbolic pixel embedding (i, j) from the center of the Poincaré ball $\mathcal{R}^{(i,j)} = d(h_{i,j}, 0)$ (cf. Eq. 2.9). The prediction entropy $\mathcal{H}^{(i,j)} = -\sum_{c=1}^{C} P_{i,j,c}\, log P_{i,j,c}$ is estimated as the entropy of the softmax probability array $P_{i,j,c}$ associated with the pixel (i, j) and the classes $c \in \{1, ..., C\}$. The acquisition score $\mathcal{A}$ serves as a surrogate indicator for the epistemic uncertainty of each pixel and determines which ones are presented to the human annotator for labeling, to augment the target label set Y_t.

4.6.3 Robust hyperbolic learning with feature reweighting

HNNs are prone to instability during training because of the unique topology of the Poincaré ball. More precisely, when embeddings approach the boundary, the occurrence of vanishing gradients can impede the learning process. Several solutions have been proposed in the literature to address this problem. [53] achieves robustness by clipping the largest values of the radii, [41] makes it by curriculum learning, and [126] needs to carefully initialize the hyperbolic network parameters. However, these approaches yield sub-optimal performances or are not compatible with our use case. Therefore, we introduce the *Hyperbolic Feature Reweighting (HFR)* module, designed to enhance training stability by reweighting features, prior to their projection onto the Poincaré ball. Given the feature map $Z \in \mathbb{R}^{\tilde{H} \times \tilde{W}}$ generated by the encoder, we compute the weights $L = \mathrm{HFR}(Z) \in \mathbb{R}^{\tilde{H} \times \tilde{W}}$ and use them to rescale each entry of the normalized feature map, yielding $\tilde{Z} = \frac{Z}{|Z|} \odot L$, where $|Z| = \sum_{k=1}^{\tilde{H}\tilde{W}} z_{ij}$ and $\odot$ denotes the element-wise multiplication. Intuitively, reweighting prevents embeddings from getting too close to the boundaries, where the distances tend to infinity. Our proposed HFR module is end-to-end trained and it enables the model to dynamically adapt through the various stages of training, improving its robustness.

Table 4.1. Comparison of mIoU results for different methods on the (a) GTAV→Cityscapes, (b) SYNTHIA→Cityscapes, and (C) Cityscapes→ACDC benchmarks. Methods marked with ♯ are based on DeepLab-v3+ [23], the ones with † on SegFormer-B4 [146], whereas all the others use DeepLab-v2 [22].

Method	Budget	road	side.	buil.	wall	fence	pole	light	sign	veg.	terr.	sky	pers.	rider	car	truck	bus	train	motor.	bike	mIoU	mIoU*
(a) GTAV → Cityscapes																						
LabOR [119]	2.2%	96.6	77.0	89.6	47.8	50.7	**48.0**	**56.6**	63.5	89.5	57.8	91.6	72.0	47.3	91.7	62.1	61.9	48.9	47.9	65.3	66.6	-
RIPU [145]	2.2%	96.5	74.1	89.7	53.1	51.0	43.8	53.4	62.2	90.0	57.6	92.6	73.0	53.0	**92.8**	73.8	**78.5**	62.0	**55.6**	70.0	69.6	-
HALO (ours)	2.2%	**97.5**	**79.9**	**90.2**	**55.6**	**51.5**	45.3	56.2	**66.2**	**90.2**	**58.6**	**92.8**	**73.3**	**53.5**	92.6	**76.9**	76.2	**64.2**	55.2	**70.1**	**70.8**	-
AADA♯ [127]	5%	92.2	59.9	87.3	36.4	45.7	46.1	50.6	59.5	88.3	44.0	90.2	69.7	38.2	90.0	55.3	45.1	32.0	32.6	62.9	59.3	-
MADA♯ [95]	5%	95.1	69.8	88.5	43.3	48.7	45.7	53.3	59.2	89.1	46.7	91.5	73.9	50.1	91.2	60.6	56.9	48.4	51.6	68.7	64.9	-
D²ADA♯ [140]	5%	97.0	77.8	90.0	46.0	55.0	52.7	58.7	65.8	90.4	58.9	92.1	75.7	54.4	92.3	69.0	78.0	68.5	59.1	72.3	71.3	-
RIPU♯ [145]	5%	97.0	77.3	90.4	**54.6**	53.2	47.7	55.9	64.1	90.2	59.2	93.2	75.0	54.8	92.7	73.0	79.7	68.9	55.5	70.3	71.2	-
HALO♯ (ours)	5%	**97.6**	**81.0**	**91.4**	53.7	**54.9**	**56.7**	**62.9**	**72.1**	**91.4**	**60.5**	**94.1**	**78.0**	**57.3**	**94.0**	**81.4**	**84.7**	**70.1**	**60.0**	**73.3**	**74.5**	-
HALO† (ours)	5%	**98.2**	**85.4**	**92.5**	**62.5**	**61.6**	**58.3**	**67.7**	**74.9**	**92.2**	**65.1**	**94.7**	**79.9**	**60.8**	**94.6**	**84.1**	**85.4**	**83.6**	**61.2**	**75.5**	**77.8**	-
Eucl. Supervised DA♯	100%	97.4	77.9	91.1	54.9	53.7	51.9	57.9	64.7	91.1	57.8	93.2	74.7	54.8	93.6	76.4	79.3	67.8	55.6	71.3	71.9	-
Hyper. Supervised DA♯	100%	97.6	81.2	90.7	49.9	53.2	53.5	58.0	67.2	91.0	59.1	93.9	74.2	52.6	93.1	76.4	81.0	67.0	55.0	70.8	71.9	-
(b) SYNTHIA → Cityscapes																						
RIPU [145]	2.2%	96.8	76.6	89.6	45.0	47.7	45.0	53.0	62.5	90.6	-	**92.7**	73.0	52.9	93.1	-	80.5	-	52.4	70.1	70.1	75.7
HALO (ours)	2.2%	**97.5**	**81.7**	**90.5**	**52.8**	**52.8**	**45.6**	**57.3**	**67.1**	**91.2**	-	92.6	**74.5**	**54.9**	**93.3**	-	**81.6**	-	**55.2**	**71.1**	**72.5**	**77.6**
AADA♯ [127]	5%	91.3	57.6	86.9	37.6	48.3	45.0	50.4	58.5	88.2	-	90.3	69.4	37.9	89.9	-	44.5	-	32.8	62.5	61.9	66.2
MADA♯ [95]	5%	96.5	74.6	88.8	45.9	43.8	46.7	52.4	60.5	89.7	-	92.2	74.1	51.2	90.9	-	60.3	-	52.4	69.4	68.1	73.3
D²ADA♯ [140]	5%	96.7	76.8	90.3	48.7	51.1	54.2	58.3	68.0	90.4	-	93.4	77.4	56.4	92.5	-	77.5	-	58.9	73.3	72.7	77.7
RIPU♯ [145]	5%	97.0	78.9	89.9	47.2	50.7	48.5	55.2	63.9	91.1	-	93.0	74.4	54.1	92.9	-	79.9	-	55.3	71.0	71.4	76.7
HALO♯ (ours)	5%	**97.5**	**81.5**	**91.5**	**56.5**	**52.7**	**57.0**	**63.2**	**72.9**	**92.0**	-	**94.4**	**77.8**	**57.4**	**94.4**	-	**86.1**	-	**60.5**	**73.5**	**75.6**	**80.2**
HALO† (ours)	5%	**98.3**	**86.5**	**92.6**	**61.0**	**61.5**	**60.6**	**67.6**	**76.2**	**93.2**	-	**94.6**	**80.8**	**58.9**	**95.0**	-	85.1	-	**62.7**	**75.6**	**78.1**	**82.1**
Eucl. Supervised DA♯	100%	97.5	81.4	90.9	48.5	51.3	53.6	59.4	68.1	91.7	-	93.4	75.6	51.9	93.2	-	75.6	-	52.0	71.2	72.2	77.1
Hyper. Supervised DA♯	100%	97.7	82.2	90.3	53.0	48.8	51.7	56.0	66.1	91.4	-	94.2	75.0	51.5	93.4	-	82.1	-	52.8	70.2	72.3	77.1
(c) Cityscapes → ACDC																						
RIPU [145]	2.2%	91.4	69.5	83.8	**52.7**	41.6	52.8	66.4	54.2	**85.1**	47.5	**94.7**	54.5	21.8	**85.5**	**58.7**	58.8	**76.9**	41.4	**45.9**	62.3	-
HALO	2.2%	**92.6**	**71.3**	**84.5**	51.3	**43.1**	**53.5**	**67.2**	**57.6**	85.1	**49.5**	94.5	**57.2**	**28.6**	84.1	53.3	**76.0**	66.9	**44.1**	41.4	**63.2**	-
RIPU♯ [145]	5%	**92.7**	**72.5**	84.7	53.1	44.8	56.7	69.1	58.9	85.9	46.9	**95.3**	57.2	24.3	84.5	**61.4**	59.4	79.0	36.9	43.6	63.5	-
HALO♯	5%	92.6	72.2	**84.8**	**54.9**	**47.7**	**59.5**	**71.5**	**61.1**	**86.1**	**49.5**	95.2	**60.7**	**30.6**	**85.8**	58.4	**73.8**	**82.0**	**41.6**	**53.2**	**66.4**	-
HALO†	5%	**95.2**	**79.8**	**88.2**	**60.2**	**51.1**	**64.1**	**78.2**	**65.6**	**87.9**	**55.7**	**95.5**	**66.3**	20.7	**88.9**	**82.2**	**89.3**	**87.9**	**50.4**	**59.0**	**71.9**	-

4.7 Experiments and Results

In this section, we describe the benchmarks and we perform a comparative evaluation against the SOTA (Sec. 4.7.1). We conduct ablation studies on the components of HALO and additional analyses in Sec. 4.7.2 and 4.7.3. The implementation follows [145] (details in Appendix 4.9.1).

Datasets For pre-training, we utilize GTAV [108] and SYNTHIA [109] synthetic datasets, each comprising 24,966 and 9,000 densely annotated images, with 19 and 16 classes, respectively. For ADA training and evaluation, we employ real-world target datasets, specifically **Cityscapes** (CS) train and val sets or **ACDC** train and test sets, both categorized into the same 19 classes. CS [32] consists of 2,975 training samples and 500 validation samples. **ACDC** [111] comprises 4,006 images captured

under adverse conditions (fog, nighttime, rain, snow) to maximize the complexity and diversity of the scenes.

Training protocol The model undergoes a pre-training on either GTAV or SYNTHIA source synthetic datasets. Subsequently, the model is domain adapted using both the source and the target datasets. Our hyperbolic radius-based acquisition method is used to select pixels to be labeled in five evenly spaced rounds during training, with either 2.2% or 5% of the total labels. Our model is additionally trained under adverse weather conditions, using CS and ACDC as the source and target datasets, respectively, in line with [61] and [11]. The ADA performances in Table 4.1 are also compared with the corresponding supervised domain adaptation baselines (Supervised DA). Supervised DA refers to the process where the adaptation to a target dataset involves using all of its labels (i.e., 100%) for the whole training, in contrast to active domain adaptation which uses a smaller fraction (e.g., 2.2% or 5%) of labels.

Evaluation metrics To assess the effectiveness of the models, the mean Intersection-over-Union (mIoU) metric is computed on the target validation set. For GTAV→CS and CS→ACDC, the mIoU is calculated on the shared 19 classes, whereas for SYNTHIA→CS two mIoU values are reported, one on the 13 common classes (mIoU*) and another on the 16 common classes (mIoU).

4.7.1 Comparison with the state-of-the-art

In Table 4.1a, we present the results of our method and the most recent ADA approaches on the GTAV→CS benchmark. HALO outperforms the current state-of-the-art methods [145, 140] using both 2.2% (+1.2% mIoU) and 5% (+3.3% mIoU) of labeled pixels, reaching 70.8% and 74.5%, respectively. Additionally, our method is the first to surpass the supervised domain adaptation baseline (71.9%), even by a significant margin (+2.6%). A thorough analysis on the performance at increasing budgets is provided in Sec. 4.7.3. HALO achieves state-of-the-art also in the SYNTHIA→CS case (cf. Table 4.1b), where it improves by +2.4% and +4.2% using 2.2% and 5% of labels, reaching performances of 72.5% and 75.6%, respectively. Additionally, we train HALO with the SegFormer-B4 [146] segmenter to demonstrate the adaptability of our approach to different architectures. With SegFormer-B4,

Table 4.2. Ablation study conducted with the Hyperbolic DeepLab-v3+ as backbone with 5% budget. Performance of prediction entropy and hyperbolic radius scores in isolation (a and b) and combined (c).

Ablative version	**mIoU**
(a) Prediction Entropy only ($\mathcal{H}$)	63.2
(b) Hyperbolic Radius only ($\mathcal{R}$)	64.1
(c) **HALO ($\mathcal{R} \odot \mathcal{H}$)**	**74.5**

HALO improves by +3.3% in GTAV→CS and +2.5% in SYNTHIA→CS compared to HALO with DeepLab-v3+, using 5% of labels. Due to the absence of other ADA studies on CS→ACDC adaptation, we train RIPU [145] as a baseline for comparison with our method. HALO improves over RIPU by +2.9% mIoU with a 5% budget, reaffirming the effectiveness of our approach on a novel dataset, as shown in Table 4.1c.

4.7.2 Ablation studies

We begin by conducting an oracular study using ground-truth labels, followed by a series of ablation studies that explore various aspects of the model. These include the selection criteria, region- versus pixel-based acquisition scores, and the proposed Hyperbolic Feature Reweighting (HFR). Additionally, we investigate strategies for stable training in hyperbolic space, evaluate the model under the source-free protocol, and analyze the correlation between Riemannian variance and classification accuracy.

Oracle experiment with ground-truth boundaries To test our hypothesis that acquiring labels solely from class boundaries results in performance decline, we conduct an oracular experiment. We replace the pseudo-labels used in RIPU with ground-truth labels, effectively evaluating an AL acquisition strategy based on ground-truth boundary pixels. Although oracular, the experiment yields a performance drop of 1.4 mIoU (69.8 vs. to RIPU's 71.2), motivating the design of a novel acquisition strategy which samples also from non-boundary regions.

Selection criteria HALO demonstrates a substantial improvement of +10.4% compared to methods (a) and (b) in Table 4.2. More precisely, utilizing solely either

the entropy (a) or the hyperbolic radius (b) as acquisition scores yields comparable performance of 63.2% and 64.1%, respectively. When these two metrics are combined, the final performance is notably improved to 74.5%.

Table 4.3. Comparison of Region- vs. Pixel-based acquisition score.

Method	Region-based	Pixel-based	mIoU (%)
RIPU [145]	✓		71.2
HALO (ours)	✓		74.1
HALO (ours)		✓	**74.5**

Region- vs. Pixel-based acquisition score Unlike the region impurity score used in RIPU [145], which relies on region statistics, the hyperbolic radius in HALO is a continuous quantity that can be computed at both the pixel and region levels. We conducted experiments comparing the effectiveness of region- versus pixel-based acquisition scores, and the results show only a small difference in performance between the two approaches (74.1% vs. 74.5%), indicating that HALO does not require a region-based formulation. Table 4.3 further supports this finding. While the region impurity score in RIPU requires pixel regions to function, HALO's hyperbolic radius can be applied on both pixel and region bases. When we train HALO using the region-based approach for comparison, we observe a small performance difference of -0.4% on the GTAV→CS benchmark with 5% acquired labels. However, despite this slight reduction, HALO still achieves a significant improvement over the RIPU baseline, demonstrating the flexibility and robustness of HALO's pixel-based approach.

Hyperbolic Feature Reweighting (HFR) Hyperbolic Feature Reweighting (HFR) plays a crucial role in improving training stability and enhancing performance in hyperbolic models. Although the mIoU improvement is modest (+1.6%), the key benefit of HFR lies in its ability to stabilize training, as hyperbolic models often struggle to converge without it. Interestingly, HFR does not provide any benefit to Euclidean models and instead negatively impacts their performance.

Table 4.4 offers insights into the performance of both hyperbolic and Euclidean models, with and without HFR. For the HALO model, the performance remains unchanged in the source-only setting with or without HFR. However, in the

Table 4.4. HFR Performance Comparison: Evaluating the impact of Hyperbolic Feature Reweighting (HFR) on hyperbolic and Euclidean models in source-only and source+target protocols.

Encoder	Protocol	HFR	mIoU (%)
DeepLab-v3+	source-only	✗	36.3
DeepLab-v3+	source-only	✓	22.7
Hyper DeepLab-v3+	source-only	✗	39.0
Hyper DeepLab-v3+	source-only	✓	38.9
HALO	source+target	✗	72.9
HALO	**source+target**	✓	**74.5**

source+target ADA scenario, HFR results in a performance boost of 1.2%. Notably, HFR also stabilizes hyperbolic model training, as models trained without HFR require a warm-up schedule and still fail to converge in approximately 20% of runs. Therefore, HFR is essential not only for improving ADA performance but also for ensuring the stability of hyperbolic learning.

Table 4.5. HALO performance on the **source-free protocol** on GTAV→CS, compared with the previous state-of-the-art approach. Methods marked with ♯ are based on DeepLab-v3+ [23], whereas all the others use DeepLab-v2 [22].

Method	**Budget**	**mIoU**
RIPU [145]	2.2%	67.1
HALO (ours)	2.2%	**70.1**
HALO♯ (ours)	5%	**73.3**

Source-free domain adaptation In the source-free protocol, the model is pre-trained on the source dataset and domain-adapted using only the target dataset. In Table 4.5 we show the performance of HALO on the source-free GTAV → CS domain adaptation task. HALO surpasses the current best [145] by +3% using 2.2% of labels.

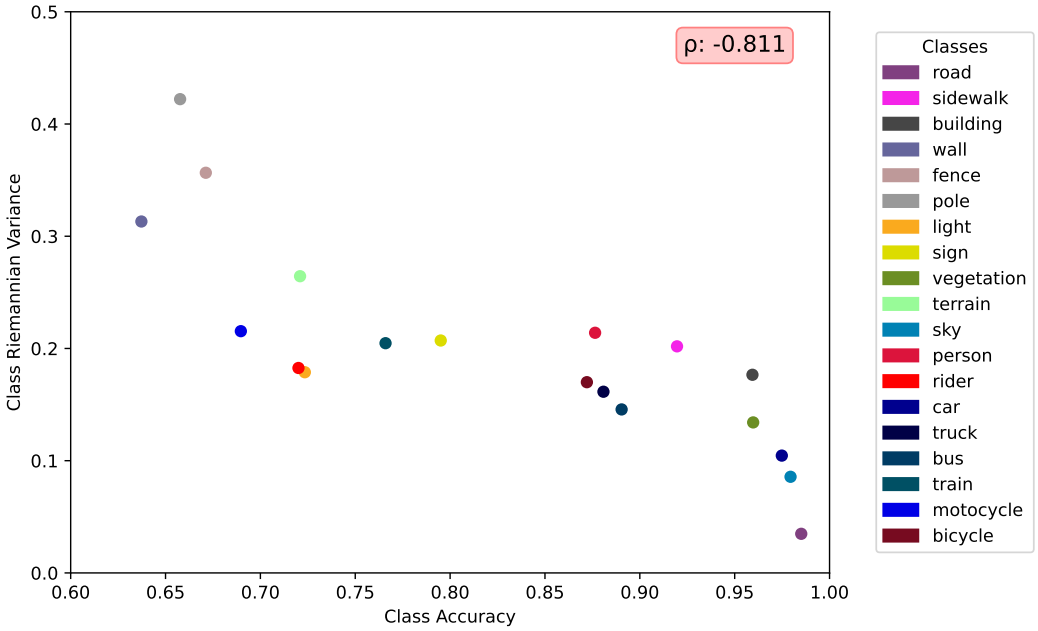

Figure 4.5. Plot of per-class accuracy against per-class Riemannian variance.

4.7.3 Additional analyses

Correlation analysis on Cityscapes→ACDC In addition to GTAV→CS, we report the correlations for CS→ACDC. The correlation of hyperbolic radius vs. percentage of target pixels is -0.868, while the correlation of prediction entropy vs. class accuracy is -0.892. These results are in line with the GTAV→CS case (-0.899 and -0.913), showing that the proposed method generalizes well even on a different domain adaptation benchmark.

Analysis on the Riemannian variance Fig. 4.5 complements our analysis by plotting the class accuracies vs. the Riemannian variance (see Eq. 2.11) of radii for each class. The latter generalizes the Euclidean variance, considering the increasing Poincaré ball density at larger radii. The correlation between accuracy and Riemannian variance is noteworthy ($\rho = -0.811$), indicating that challenging classes, like *pole*, exhibit lower accuracy and larger Riemannian variance, occupying a greater volume in the space.

Class imbalance with increasing budget We experiment with different labeling budgets, observing performance improvements as the number of labeled pixels

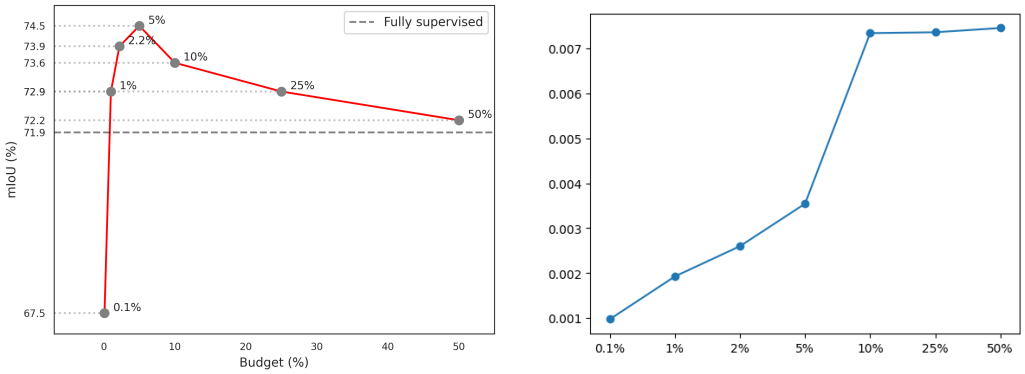

Figure 4.6. (left) Performance on GTAV → Cityscapes with different budgets. (right) Evolution of the variance (y axis) of selected pixels distributions with varying budget (x axis).

increases. However, beyond a threshold of 5%, adding more labeled pixels leads to diminishing returns (see Fig. 4.6 (left)). We believe this may be explained by data unbalance: taking all labels to domain adapt means that most of them belong to a few classes. Indeed, *road*, *building* and *vegetation* account for 77% of the total labels, potentially hindering successive training rounds due to data redundancy.
To verify the intuition on data imbalance, we have evaluated the variance of selected pixels distributions as the labelling budget increases. In Fig. 4.6 (right), we start with the acquisition of just 0.1% of labels from the target dataset. At this stage, the variance is at a minimum, as HALO manages to identify and select labels from each class in equal proportions. Then the variance increases slowly until the budget reaches 5%. This happens as the model manages to select pixels from each class, balancing the acquired data selection. The variance has a steep increase at budgets of 10% and higher. This occurs because the model has already selected most of the labels from the complex and scarce classes which it can identify thanks to the hyperbolic radius and the prediction entropy (cf. Sec. 4.6.2). So, for budgets of 10% or more, the data acquisition strategy is influenced by the target dataset imbalance. The imbalance trend in label selection matches the performance variation in Fig. 4.6 (left). Therefore, we conclude that HALO's selection aids performance, beyond the supervised domain adaptation, until the model manages to successfully identify complex and scarce classes, and until they are available in the target dataset.

Exploring strategies for stable training in hyperbolic space Here we conduct a more comprehensive evaluation of approaches aimed at stabilizing the training of

hyperbolic neural networks. We test [55]'s Feature Clipping method in our framework for comparison with our HFR. As shown in the Table 4.6, while Feature Clipping works and produces better results than the baseline RIPU, it still falls short of our HFR method (-1.2%). [55] utilize Feature Clipping to prevent vanishing gradients during backpropagation. Despite its simplicity, this technique restricts the model's representational capacity by clipping features, resulting in inferior performance compared to our HFR. We have not tested the curriculum learning of [41] and the initialization approach of [126], because their adaptation to the ADA task is not straightforward.

Table 4.6. Comparison of strategies for stable training in hyperbolic space

Method	mIoU (%)
Hyperbolic Feature Reweighting (ours)	**74.5**
Feature Clipping (Guo et al., 2022) [55]	73.3
Initialization (van Spengler et al., 2023) [126]	not compatible
Curriculum Learning (Franco et al., 2023) [41]	not compatible

The curriculum learning in [41] is specifically tailored for metric learning scenarios involving a hyperbolic loss, enabling training in hyperbolic space by utilizing cosine distance for improved initialization, gradually transitioning to the Poincaré loss. Our method does not involve comparing embeddings and leveraging the Poincaré loss. Similarly, the initialization approach in [126] is designed explicitly for fully hyperbolic ResNets, particularly hyperbolic convolutions. As we do not employ hyperbolic convolutional layers, their initialization approach is not immediately suitable for our model.

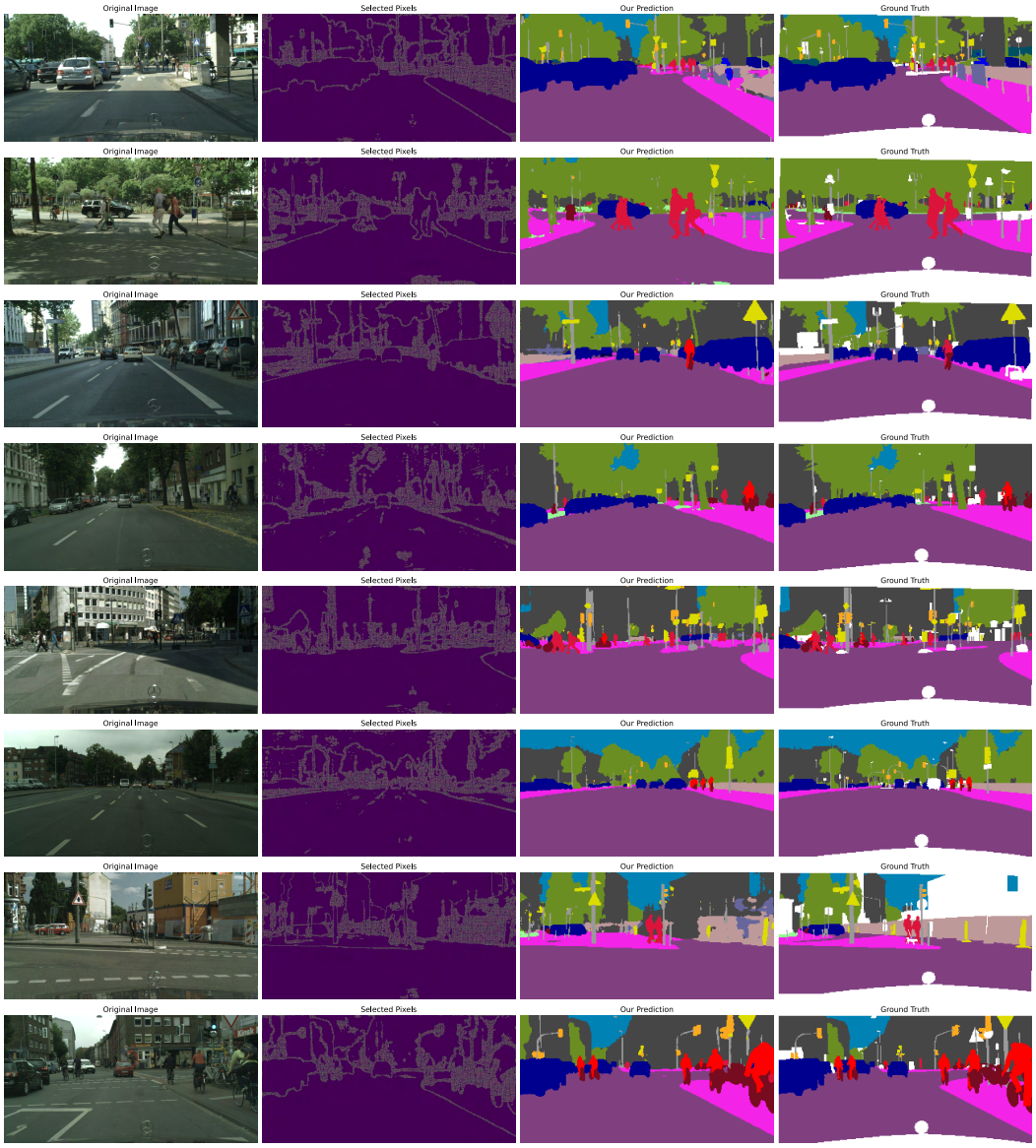

Figure 4.7. Qualitative Results Visualization for the GTAV → Cityscapes Task. The figure showcases different subfigures representing: the original image, HALO's pixel selection, HALO's prediction, and the ground-truth label. Zoom in for the details.

4.8 Qualitative results

In Fig. 4.7, we present visualizations of HALO's predicted segmentation maps and the selected pixels. In the first row, HALO prioritizes the selection of pixels that are not easily interpretable, as evident in the *fence* or *wall* on the right side of the image. Notably, HALO does not limit itself to selecting contours exclusively;

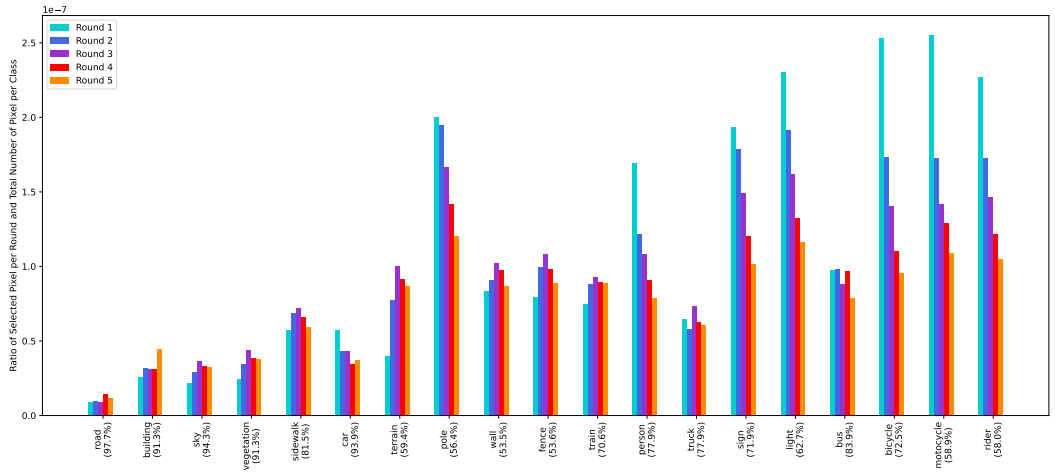

Figure 4.8. Ratio between the selected pixels for each class at each round and the total number of pixels per class. Each color shows the ratio in the specific round. On the x-axis are reported the classes with the relative mIoU (%) of HALO (cf. Table 1 of the main paper) ordered according to they decreasing hyperbolic radius.

it continues to acquire pixels within classes if they exhibit high acquisition score. This behavior is also observed in rows 2, 3, and 4 of Fig. 4.7. For classes with lower complexity, such as *road* and *car*, HALO acquires only the contours. However, for more intricate classes like *pole* and *signs*, it also selects pixels within the class.

In rows 5, 6, and 8, the images depict a crowded scene with numerous small objects from various classes. Remarkably, the selection process directly targets the more complex classes (such as *pole* and *signs*), providing an accurate classification of these. In row 7, we observe an example where the most common classes (*road*, *vegetation*, *building*, *sky*) dominate the majority of the image. HALO efficiently allocates the labeling budget by focusing on the more complex classes, rather than expending resources on these prevalent ones. Refer to Sec. 4.8.1 and Fig. 4.9 for a detailed overview of the selection prioritization during each active learning round.

4.8.1 Selection Rounds in the Data Acquisition Strategy

Here we analyze how the model prioritizes the selection of the pixels during the different rounds. In Fig. 4.8, we consider the ratio between the selected pixel at each round and the total number of pixels for the considered class. Note how the model selects in the early stages from the class with high intrinsic difficulty (e.g., rider, bicycle, pole). During the different rounds, the number of selected pixels decreases

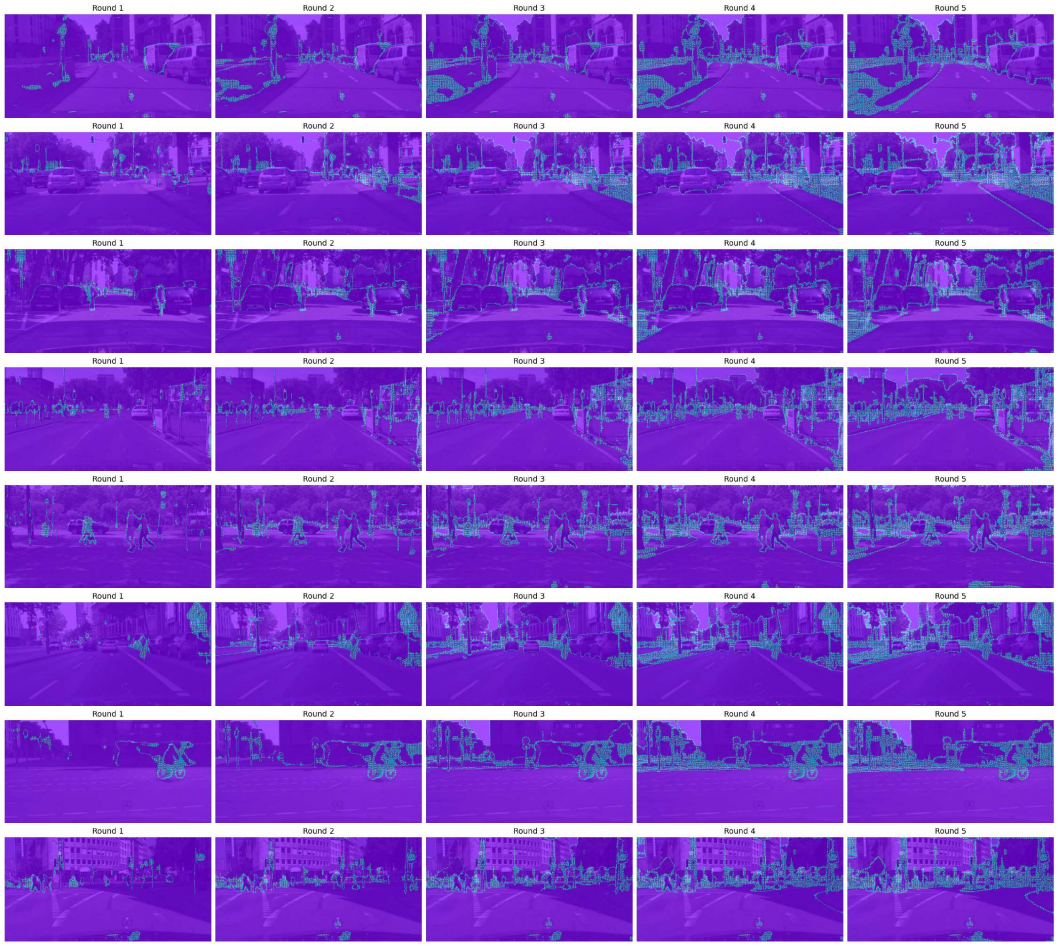

Figure 4.9. Qualitative analysis on the pixel selected by HALO at each round. Zoom in to see details.

because of the scarcity of pixels associated with these classes. On the other hand, less complex classes are less considered in the early stages and the model selects from them in the intermediate rounds if the class has an intermediate complexity (e.g., wall, fence, sidewalk) or in the last stages if it has low complexity (e.g., road or building).

The qualitative samples of pixel selections in Fig. 4.9 corroborate this observation. In rounds 1 and 2, the model gives precedence to selecting pixels from more complex classes (e.g., *poles*, *sign*, *person*, or *rider*). Subsequently, HALO shifts its focus to two distinct objectives: i) acquiring contours from classes with lower complexity (e.g., *road*, *car*, or *vegetation*), and ii) obtaining additional pixels from more complex classes (e.g., *pole* or *wall*). Notably, in rows 1, 2, 3, 5, and 6, HALO gives priority to selecting complete objects right from the initial round (as seen with the *sign*).

Another noteworthy instance is the acquisition of the *bicycle* in row 7. The hyperbolic radius score enables the acquisition of contours that extend beyond the boundaries of pseudo-label classes. In this case, we observe precise delineation of the internal portions of the wheels.

4.8.2 Qualitative comparison with the baseline model

The top row of Fig. 4.10 depicts label acquisition using the baseline RIPU method with budgets of 2.2% (left) and 5% (right). The bottom row illustrates visualizations with our proposed HALO using the same budgets.

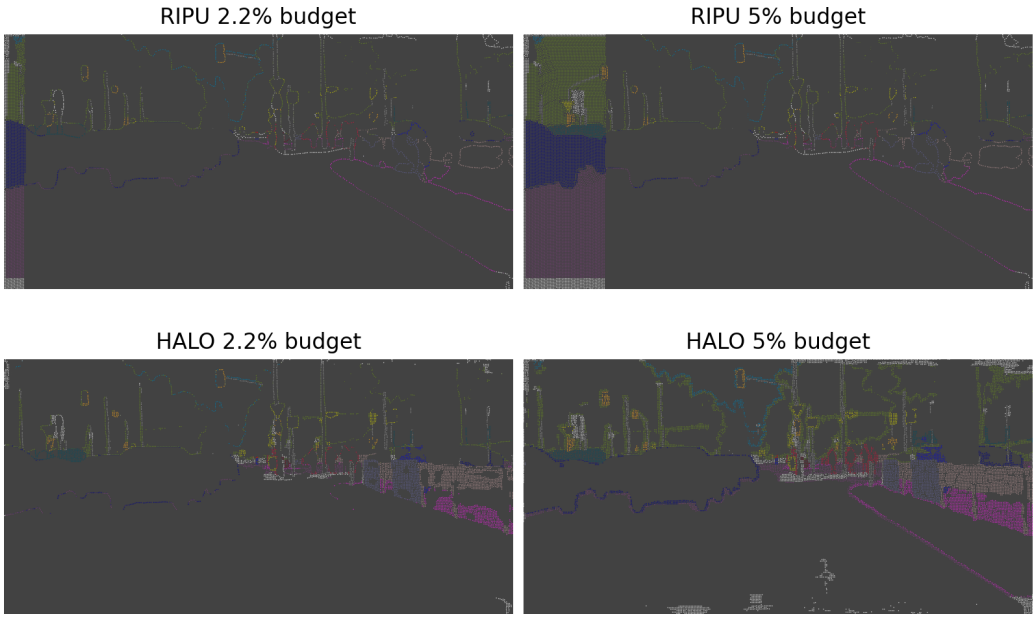

Figure 4.10. (top-row) Pixel selection with RIPU's baseline; (bottom-row) Pixel selection with out HALO; (left-column) Selection with budget 2.2%; (right-column) Selection with budget 5%. Zoom in for the details.

Noteworthy observations include:

- By design, RIPU only concentrates on selecting boundaries between semantic parts (ref. Fig. 4.10 top-left). However, since there are only a few (thin) boundary pixels, RIPU soon exhausts the pixel selection request. Next, when a larger budget is available, RIPU simply samples from the left side. The random selection still provides additional labels (ref. Fig. 4.10 top-right) and is a good baseline, cf. Table 3 of [145], although not as good as HALO's

acquisition strategy.

- By contrast, HALO showcases pixel selection from both boundaries and internal regions within semantic parts (ref. Fig. 4.10 bottom-left). Especially passing from 2.2% to 5% acquisition budget, HALO considers thick boundaries, so also parts of objects close to the boundaries, but also areas within objects, as it happens for wall, fence, pole, and sidewalk, cf. the right image part in the bottom-right of Fig. 4.10.

4.9 Implementation and Computational Considerations

4.9.1 Implementation details

For all experiments, the model is trained on 4 Tesla V100 GPUs using PyTorch [100] and PyTorch Lightning with an effective batch-size of 8 samples (2 per GPU). The DeepLab-v3+ architecture is initialized with an Imagenet pre-trained ResNet-101 as the backbone. *RiemannianSGD* optimizer with momentum of 0.9 and weight decay of 5×10^{-4} is used for all the trainings. The base learning rates for the encoder and decode head are 1×10^{-3} and 1×10^{-2} respectively, and they are decayed with a "polynomial" schedule with power 0.5. The models are pre-trained for 15K iterations and adapted for an additional 15K on the target set. As per [145], the source images are resized to 1280×720, while the target images are resized to 1280×640.

4.9.2 Comparison of parameters count

Table 4.7. Comparison of parameters count in HALO vs. RIPU [145].

Method	Segmenter	Dim.	HFR (params)	Total Params
RIPU	DeepLab-v3+	512	Not used	60.1M
HALO (ours)	Hyper-DeepLab-v3+	64	10k	60.1M

To provide additional insights into the hyperbolic architecture employed, we conduct a comparison of parameter counts between RIPU [145] and our method HALO (see Table 4.7). Both employ the DeepLab-v3+ architecture but with some distinctions. RIPU operates with a pixel embedding dimension of 512, resulting in a parameter count of 60.1M. In contrast, HALO operates with a reduced pixel

embedding dimension of 64, which the adoption of a hyperbolic learning enables. Moreover, the HyperMLR requires fewer parameters than the Euclidean Linear layer used for classification due to the reduced embedding dimension. This results in a slightly lower total parameter count than RIPU's (10k fewer params). Additionally, HALO introduces the HFR module, consisting of two linear layers separated by a BatchNorm layer and a ReLU. Thanks to the lower embedding dimensions, the input and output sizes of the HFR module are only 64-dimensional, adding less than 10k additional parameters. This roughly matches the number of parameters removed from the segmenter. These modifications result in the parameter count being nearly identical between the two methods (60.1M), aligning with other studies leveraging the DeepLab-v3+ architecture.

4.9.3 Computational Resources

We evaluated the computational resources required by our method, HALO, compared to the previous state-of-the-art, RIPU, using the setup described in Appendix 4.9.1. The comparison was conducted under the source+target protocol.

4.9.4 Environment and Computational Load Metrics

Using an identical environment — the same conda environment, 4 Tesla V100 GPUs, and a batch size of 2 per GPU — we compared different computational load metrics for HALO and RIPU using DeepLab-v3+. Table 4.8 summarizes the comparison:

Table 4.8. Comparison of computational load metrics for HALO and RIPU [145].

Method	FLOPS ↓	FPS ↑	Params ↓
RIPU	125.49 M	5.62	60.1 M
HALO (ours)	280.17 M	4.72	60.1 M

FLOPs (Floating Point Operations) are calculated only for the classification layers, which differ between the RIPU and HALO segmentation models. The rest of the model architectures require 1.7 TFLOPS. FPS (Frames Per Second) is measured at inference time. The parameter count (Params) refers to the entire model.

Both models have identical parameter counts and are trained using the same

batch size, resulting in negligible differences in memory consumption.

Training and inference times Training RIPU takes 12.5 hours, while training HALO takes 13.5 hours, which is just an 8% increase. This marginal difference is primarily due to the nature of operations in hyperbolic space, such as Möbius addition. Despite this, the increased computational cost of hyperbolic operations is offset by the reduced embedding size needed to achieve state-of-the-art performance in hyperbolic space, as detailed in Appendix 4.9.2.

At inference time, evaluated on the Cityscapes validation set, RIPU takes 1m:29s, while HALO takes 1m:46s. Again, the difference in computing time is minor.

Optimization considerations It is important to note that existing implementations of hyperbolic operations are still under active development and may not yet be fully optimized for mainstream deep learning frameworks and hardware. Specifically, the primary hyperbolic operations — exponential mapping (Eq. 2.4), Möbius addition (Eq. 2.7), and Poincaré distance (Eq. 2.8) — have not been optimized to the same extent as their Euclidean counterparts, which have benefited from over a decade of optimization.

This analysis demonstrates that the computational overhead introduced by hyperbolic operations is manageable and does not significantly impact the overall efficiency of our method.

4.10 Discussion

In this study, have introduced the first hyperbolic neural network technique for AL, which we have extensively validated as the novel state-of-the-art on semantic segmentation under domain shift. We have identified a novel interpretation of the hyperbolic radius as an estimator of data scarcity and epistemic uncertainty, and we have supported the finding with experimental evidence. The novel concept of hyperbolic radius and its successful use as an acquisition strategy in AL are a step forward in understanding hyperbolic neural networks.

Chapter 5

Hyperbolic Learning with Multimodal Large Language Models

5.1 Scaling Up Hyperbolic Learning

Building on HALO's innovative use of hyperbolic embeddings for active learning in semantic segmentation, we were inspired to push the boundaries of hyperbolic geometry in even more complex machine learning scenarios. As the field of AI rapidly advances towards large-scale multimodal models, we recognized an opportunity to extend hyperbolic representations to the realm of vision-language models (VLMs). This progression from HALO's focused application in semantic segmentation to the complex, high-dimensional space of multimodal learning represents a significant leap in both scale and complexity. By adapting hyperbolic embeddings to work within the BLIP-2 architecture, we aimed to harness the uncertainty-capturing capabilities of hyperbolic space in models with billions of parameters. This transition not only demonstrates the scalability of hyperbolic approaches across different AI domains but also highlights the challenges and potential benefits of incorporating non-Euclidean geometries in state-of-the-art multimodal systems. The journey from HALO to Hyperbolic Multimodal LLMs underscores our ongoing exploration of hyperbolic geometry's capacity to enhance various aspects of machine learning, from specialized computer vision tasks to broad, multimodal understanding.

5.2 Overview

To harness the recent advancements in language models across different modalities, modern vision-language models (VLMs) have evolved to combine visual processing with the reasoning capabilities of large language models (LLMs)[75, 81, 2, 154]. These integrated architectures enable reasoning tasks and encompass extensive world knowledge, facilitating capabilities such as zero-shot segmentation and improved generalization for images [73]. Models like BLIP-2 (Bootstrapping Language-Image Pre-training) combine language understanding with visual inputs through a two-stage pre-training approach [75]. In the first stage, BLIP-2 aligns visual features with textual descriptions. In the second stage, it fine-tunes this alignment for tasks such as image captioning and visual question answering. Despite these advancements, BLIP-2 still faces limitations, particularly in representing complex hierarchical structures and relationships in the data. These limitations are crucial during image and text alignment and highlight the need for an uncertainty measure to ensure consistency between image and language representations.

To better understand the embedding space of an LLM, we compare its Euclidean and hyperbolic counterparts. Hyperbolic space provides a more efficient and compact representation of hierarchical structures due to its exponential scaling properties [35, 90]. In contrast, Euclidean space grows linearly, making it challenging to represent hierarchical data with low distortion. Leveraging the hyperbolic space allows models to achieve an effective understanding of hierarchical data, leading to improved performance in tasks such as active learning [40], action recognition [41], and segmentation [5]. Additionally, hyperbolic embeddings contain an uncertainty measure represented by the distance from the center of the Poincaré ball, a specific instance of a hyperbolic manifold. However, it has not yet been mathematically proven how this property is achieved. Some works [21, 5, 130] show that a lower radius is associated with higher uncertainty, and conversely at a high radius there are less uncertain samples. In contrast, other recent work [41] links radius to complexity or specifically epistemic uncertainty. In this case the situation is reversed; at low radius there are more generic and common classes, while at high radius more complex and specific classes. Although some work is still needed in this direction, utilizing this distance-based measure of uncertainty models can better assess and manage the confidence levels of their predictions, leading to a more reliable and interpretable grounding of vision and language.

Although some studies [35, 63] have explored the use of hyperbolic space in medium-sized vision-language models like CLIP [106], hyperbolic embeddings remain largely unexplored in large-scale modern vision-language models (VLMs) such as BLIP-2, which features 2.7 billion parameters and maps vision-based features into a shared multimodal space with text before passing them to the LLM. We propose combining BLIP-2 with hyperbolic embeddings to enhance its representational power, improving the model's ability to capture intricate relationships and perform complex reasoning, while also enabling the use of hierarchical data structures in multimodal learning. In this work, we demonstrate that traditional hyperbolic approaches, such as using Poincaré distance as a similarity measure, fail when applied within BLIP-2. To overcome these limitations, we propose a stable hyperbolic training method that maintains performance with the Euclidean metrics on retrieval tasks while producing image embeddings with varying hyperbolic radii. Our key contributions are as follows:

- We present, to our knowledge, the first large-scale hyperbolic VLM and show that hyperbolic embeddings can capture uncertainty information via their radii, while achieving performance comparable to the Euclidean baseline of BLIP-2;
- We provide detailed quantitative and visual analyses of the different embedding spaces, offering insights into the operation of current VLMs and how to efficiently represent them in hyperbolic space.

5.3 Related Works

5.3.1 Hyperbolic Representation Learning

Hyperbolic representation learning in deep neural networks gained momentum with [45], introducing hyperbolic counterparts for classical fully-connected layers, multinomial logistic regression, and RNNs. This approach later expanded to hyperbolic convolutional neural networks [117], hyperbolic graph neural networks [84, 18], and hyperbolic attention networks [51]. Hyperbolic geometry's ability to encode hierarchies and tree-like structures [94, 132, 18, 68, 130, 5] as well as the notion of uncertainty via embedding radius [37, 21, 39, 40] have been key motivations. Recent advancements include self-supervised learning in hyperbolic space [149, 130, 37, 41,

40], where hierarchy or uncertainty measure emerge without a direct supervision.

Transposing models from Euclidean to hyperbolic is challenging due to the specific hyperparameter configurations required, particularly the curvature of the manifold, which is data-dependent [68, 37]. Hyperbolic neural networks have been explored for encoding full images [125, 7], single pixels [5, 41, 21], and text [132, 36]. Combining different modalities is even more complex. [35] introduced MERU, a CLIP-like model to align text and image modalities. However, these models are still not comparable to large-scale and multi-purpose models like BLIP-2 [75], which leverage extensive pre-training and fine-tuning for tasks such as VQA and captioning.

5.3.2 Image-Text Alignment

Mapping images and text into the same latent space has been a useful technique within a range of multimodal tasks, such as cross-modal retrieval. It is most directly useful for cross-modal retrieval tasks. CLIP [106] is a popular model in this space, consisting of image and text encoders trained on large-scale data to embed images and text into a shared latent space, using contrastive learning. Recent VLMs have taken a different approach, training multimodal adapter layers between a vision encoder and LLM [33, 75, 81, 153, 160]. This aligns visual representations with the LLM input space. As part of training the multimodal adapter, BLIP-2 and InstructBLIP [33, 75] additionally use an image-text contrastive loss similar to CLIP.

In our work, we explore a hyperbolic version of BLIP-2, presenting our findings and highlighting the challenges in leveraging contrastive learning for hyperbolic representations.

5.4 Background

5.4.1 Hyperbolic Geometry

For an introduction to hyperbolic geometry and hyperbolic neural networks, please refer to Chapter 2.

5.4.2 BLIP-2

In BLIP-2, Li et al. [75] leverage pre-trained vision and language models to reach a strong multimodal understanding with minimal additional training. More specifically, it begins by extracting a visual embedding from an image I using a pre-trained and frozen image encoder M_{image}. The embedding for image I is thus computed as:

$$v = M_{\text{image}}(I) \tag{5.1}$$

Next, a relatively small adapter model, called a Q-Former (or Querying-Transformer) M_{QF}, is trained along with a projection layer P to translate the embedding into the input space of a pre-trained and frozen LLM M_{text} (e.g., OPT [158]). The Q-Former is a transformer encoder that takes in a sequence of N fixed "query" vectors $\mathbf{Q} = [q^j]_{j=1}^N$ that interact with the image embedding v via cross-attention. This produces output query embeddings $\mathbf{E}$:

$$\mathbf{E} = \left[e^j\right]_{j=1}^N = M_{\text{QF}}(v, \mathbf{Q}) \tag{5.2}$$

In parallel, a text encoder M_{text} represents the caption T relative to the image.

$$u = M_{\text{text}}(T) \tag{5.3}$$

For clarity, we will consider the text encoder M_{text} as part of the Q-former.

In BLIP-2, the text is aligned via cosine distance with the most similar query embedding in $\mathbf{E}$.

$$L_{\text{contr}}(\mathbf{E}, u) = \min_{j=1,\dots,N} d_{cosine}(e^j, u) = \min_{j=1,\dots,N} 1 - \frac{e^j \cdot u}{\|e^j\|\|u\|} \tag{5.4}$$

Finally, each embedding e^j is projected by the linear layer P to match the dimensionality of the LLM input token embeddings. To generate text, the LLM prompt begins with the sequence of embeddings $\mathbf{E}$, followed by the rest of the tokenized prompt.

5.5 Stabilizing Hyperbolic BLIP-2

In this section, we discuss challenges encountered in adopting standard approaches to hyperbolic contrastive learning and the solutions we implemented.

Additionally, we introduce two novel strategies: Random Query Selection (RQS) and Random Text Pruning (RTP). These strategies aim to effectively utilize hyperbolic embeddings while enhancing model stability and performance in multimodal tasks.

5.5.1 Hyperbolic Image-Text Contrastive

We applied contrastive learning in hyperbolic space by first mapping the Euclidean feature vectors u and e^j relative to the text and image inputs into the Poincaré ball using exponential mapping (see Eq. 2.4):

$$u_h = \exp_0^c(u), \quad e_h^j = \exp_0^c(e^j) \quad \text{for } j = 1, ..., N \tag{5.5}$$

This transformation positions the embeddings u_h, $\left[e_h^j\right]_{j=1}^N$ within the hyperbolic space, preserving their relational structure.

Next, we substitute the distance used in 5.4 with the hyperbolic metric, specifically the Poincaré distance (see Eq. 2.8):

$$L_{\text{hyper}}(\mathbf{E_h}, u_h) = \min_{j=1,...,N} d_{Poin}(e_h^j, u_h) \tag{5.6}$$

This distance measure, based on hyperbolic geometry, calculates the length of the geodesic between two embeddings. Notably, embeddings closer to the edge of the Poincaré ball result in longer geodesics and thus greater distances.

Following the approach in [46], we utilized the Poincaré distance to assess the similarity between embeddings. However, as noted in [41], the Poincaré distance is effective for positive-only samples [49, 26] but introduces ambiguity when applied to both positive and negative samples [25]. This ambiguity arises because repelling negative samples along geodesics can push embeddings toward the edge of the ball, thereby biasing the uncertainty.

Given these considerations and our experimental results, we opted to use the standard cosine distance with hyperbolic image and text embeddings:

$$L_{\text{hyper-cosine}}(\mathbf{E_h}, u_h) = \min_{j=1,...,N} d_{cosine}(e_h^j, u_h) \tag{5.7}$$

This choice aims to preserve the performance of the Euclidean baseline while exploring whether the model can still learn uncertainty effectively within the hyperbolic space.

5.5.2 Random Query Selection

The baseline model utilizes the Q-Former architecture [75], which outputs 32 learned query embeddings $\mathbf{E_h}$. These embeddings are designed to capture diverse semantic information from the image and align it with text. However, when computing the contrastive learning loss (e.g., Eq. 5.4, 5.6, 5.7), the model selects the most similar query embedding exclusively. This approach limits the utilization of the rich information stored in the remaining query tokens. As a result, the model tends to merge the different semantic information into a single query embedding, consistently selecting the same embedding for all inputs (c.f., Fig. 5.4).

To address this limitation, we introduce Random Query Selection (RQS) to diversify the semantics captured by the different query embeddings. Specifically, during training, the model selects the most similar query embedding to the reference text embedding with a probability p. With a probability $1-p$, the model selects a random query embedding instead:

$$L_{\text{hyper-RQS}}(\mathbf{E_h}, u_h) = \begin{cases} \min_j d_{cosine}(e_h^j, u_h), & \text{with prob} = p \\ d_{cosine}(e_h^{\tilde{j}}, u_h), & \text{with prob} = 1-p \end{cases} \tag{5.8}$$

where $\tilde{j}$ is uniformly sampled in $\{1, ..., N\}$. This stochastic selection process encourages the model to distribute the semantic information across the 32 query embeddings, thereby enhancing the robustness and diversity of the learned representations.

5.5.3 Random Text Pruning

In our model, the hyperbolic embeddings u_h associated with text inputs tend to migrate towards the edge of the Poincaré ball. This results in uniform radii, rendering them ineffective as a proxy for uncertainty (c.f. Fig. 5.3 right in purple). To address this, we induce meaningful variations in text radii by introducing noise into the text inputs. This is expected to reflect different uncertainty levels through varying noise.

To achieve this, we implement Random Text Pruning (RTP), where a randomly selected segment of the input text T is pruned. Specifically, a window of 0 to 7 consecutive tokens is removed, introducing varying levels of noise. Through experimentation, we determined that a window length of 7 yields the best performance.

By applying RTP, we expect the embeddings of noisier texts to exhibit lower

radii, aligning with the interpretation of hyperbolic representation where increased uncertainty corresponds to a reduced radius. This technique encourages the model to produce hyperbolic embeddings with radii that reflect the underlying uncertainty of the text inputs, thereby enhancing the expressiveness and utility of the hyperbolic space in representing text semantics.

5.6 Experiments and Results

This section outlines the training pipeline for our models, analyzes the learned hyperbolic embeddings, details the evaluation metrics, and presents the results on downstream tasks.

5.6.1 Training Pipeline

The training pipeline follows the methodology described in BLIP-2, integrating vision-language pre-training with contrastive learning in hyperbolic space. The main components of the training pipeline are outlined below.

Dataset The trainings of our models were conducted on the COCO dataset [80]. **COCO (Common Objects in Context)** consists of over 200,000 images with a wide range of annotations for object detection, segmentation, and captioning. It captures a variety of everyday scenes with common objects in their natural contexts.

Model Architecture: Our model architecture includes a frozen pre-trained state-of-the-art vision transformer, ViT-g/14 from CLIP [106], a frozen language model OPT [158], and the Q-former. The outputs of the vision and language encoders are projected into the Poincaré ball for contrastive learning in the Q-former.

Model Training: The model undergoes a two-stage pre-training process: 250k steps in the first stage and 80k steps in the second, following the same setup as BLIP-2. The batch size is 800 for the first stage and 512 for the second. Each image is resized to 224x224 with standard BLIP-2 augmentations. Pre-training is conducted on a single machine with 8 V100 (32GB) GPUs. Stage 1 takes 12 hours, while Stage 2 takes 10 hours. Both stages use the same set of hyperparameters:

Figure 5.1. (top-row) Images with lowest hyperbolic radius; (bottom-row) Images with highest hyperbolic radius. Radius is indicated above each image.

AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and a weight decay of 0.05. The learning rate starts at 1e-6 with a 2000-step warmup to 1e-4, then decays following a cosine schedule to a minimum of 1e-5. After pre-training, the model is fine-tuned on the image captioning task.

5.6.2 Analysis

This section systematically examines the principal findings derived from our experiments and visualizations. Our objective was to establish the radius as an indicator of uncertainty. We provide a detailed analysis of the results obtained from employing conventional methodologies for hyperbolic representations, as mentioned in 5.5.1. Furthermore, we demonstrate the challenges encountered during this process and the corresponding countermeasures implemented to mitigate these issues.

Hyperbolic Radius as a Measure of Uncertainty In Figure 5.1, we visualize how the hyperbolic radius provides insights into its potential as a measure of uncertainty or complexity in the embeddings. Our experiments revealed that the radius can exhibit variability, hinting at the degree of uncertainty or the complexity of the embedded information. In fact, at a lower radius (see Figure 5.1 top), we observe "people skiing" where the majority of the image is dominated by background classes like snow, sky, or vegetation, while at a higher radius (see Figure 5.1 bottom),

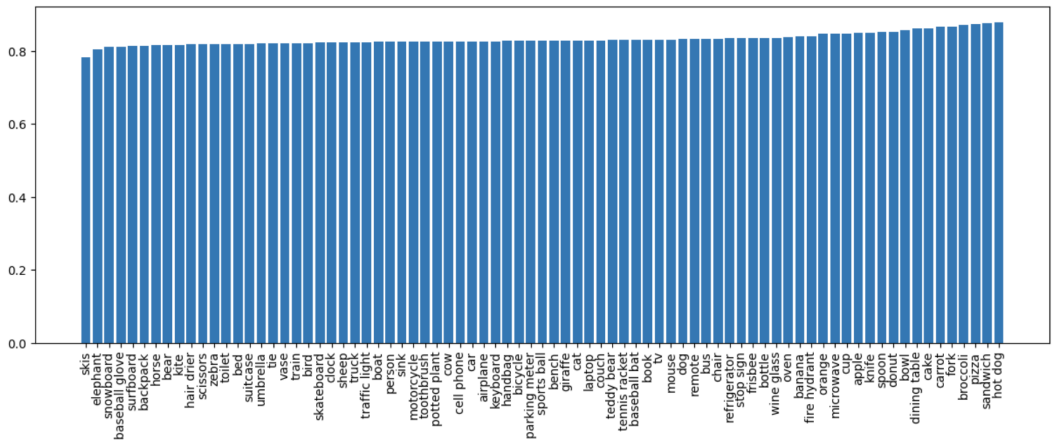

Figure 5.2. Per class radius distribution, using the annotations in the COCO dataset.

we have more complex scenarios with tables full of different objects.

In stark contrast to previous works such as[41], we observe that the average radius distribution of the different classes [1] (c.f. Figure 5.2) is mainly uniform for intermediate radii, while it looks significant only at the extremes of the radii distribution.

Challenges in Contrastive Learning with Hyperbolic Space Following the observations from [41], we found that applying contrastive learning in hyperbolic space can be problematic when utilizing both positive and negative samples, as in [25]. The Poincaré distance, while effective for measuring geodesic lengths in hyperbolic space, introduces ambiguity when repulsing with negative samples, potentially biasing the embeddings towards the edge of the ball. This issue does not arise with positive-only samples [49], suggesting a more stable training dynamic in those scenarios.

Another critical challenge identified was the difficulty of fine-tuning the Q-former, pre-trained in the Euclidean manifold and then frozen, within hyperbolic space. The misalignment between the pre-trained Euclidean image encoder and the hyperbolic representation space caused significant performance issues, indicating the necessity for consistency in embedding space across training stages.

[1] The classes are the ones provided in the COCO dataset that are used just for the analysis and comparison in the literature.

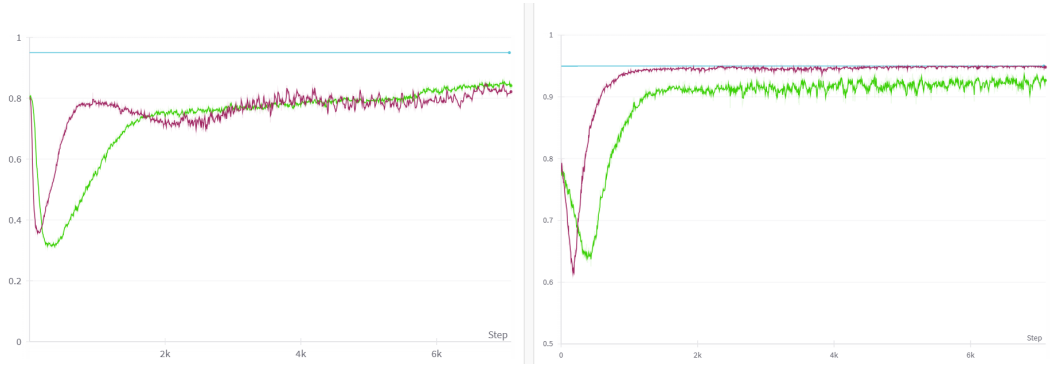

Figure 5.3. Plot of average hyperbolic radius for the image embeddings (left) and text embeddings (right). The radius is stable at the maximum level using hidden dimension 256 (cyan), while it varies using a lower dimension, i.e., 16 (purple). The text embedding does not converge at the edge of the Poincaré ball, only using Random Text Pruning (green), resulting in a more variable and meaningful radius.

Hyperbolic Mapping and Similarity Measures Our findings in Table 5.1, indicate that hyperbolic mapping remains effective only when the dot product is used to compute similarity. When the Poincaré distance is employed, the performance degrades, rendering it ineffective for the contrastive loss. However, similarities derived from the Poincaré distance can be useful for negative sample selection in Image-Text Matching tasks, providing a nuanced approach to leveraging hyperbolic geometry.

Impact of Embedding Dimension on Radius and Performance In Figure 5.3 (cyan plot), we observe that the hyperbolic radius tends to cap at its maximum level (the edge of the Poincaré ball) when using high-dimensional embeddings (i.e., 256). Consequently, the radius does not affect the training, and we do not have a meaningful radius as an uncertainty proxy. We see a significant variation in the radius by reducing the embedding dimension (purple plot) to 16. Performance metrics show a notable decline when the embedding dimension was reduced to lower values (e.g., 16), while higher dimensions (e.g., 128) maintained comparable performance to Euclidean models. These results highlight the sensitivity of hyperbolic embeddings to dimensionality, affecting both the interpretability of the radius and the overall model performance.

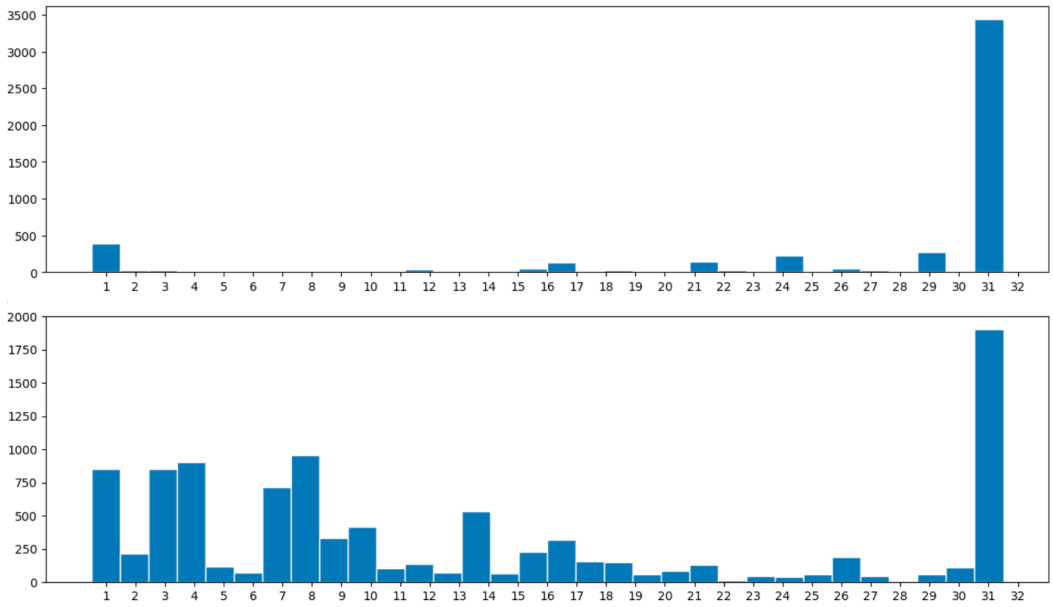

Figure 5.4. Query selection distribution across the test dataset. The top graph depicts the selection performed by the BLIP-2 model, whereas the bottom graph shows the selection performed by our model.

Variability of Image and Text Embeddings' Radii Figure 5.3 shows a distinct behavior between image (left) and text embeddings (right). Image embeddings exhibit variable radii, whereas text embeddings consistently stay at the edge of the Poincaré ball. Feature clipping [54] is used in the current literature to help in training stabilization and to alleviate this effect. However, we do not observe any significant influence on the performance of the hyperbolic model or the hyperbolic radius. For this reason, we introduce noise through Random Text Pruning (c.f. Sec. 5.5.3), which causes slight variations in the text embeddings' radii (green plot), indicating a potential but limited ability to capture text uncertainty through hyperbolic geometry.

Random Query Selection The baseline model employs the Q-former architecture, which generates 32 learned query embeddings. These embeddings are intended to encapsulate diverse semantic information from the image and align it with text semantics. However, when calculating the contrastive learning loss with the text embedding, the model exclusively selects the most similar query embedding.

In Figure 5.4 (top), we present the distribution of the selected query embeddings for the baseline BLIP-2 model. Our observations indicate that query 31 is almost

invariably chosen as the embedding that best represents the image. By introducing random query selection, we achieve a more distributed representation of information (see Figure 5.4 bottom). In stage 2, when the LLM receives all the query embeddings as input, it benefits from more informative embeddings that capture various aspects of the input image, thereby enhancing performance.

5.6.3 Evaluation Metrics

The performance of our model is evaluated using standard metrics for the downstream task of zero-shot image-text retrieval and image captioning.

Image Retrieval @1 measures the percentage of times the correct image is retrieved as the top result when given a text query.

Text Retrieval @1 measures the percentage of times the correct text is retrieved as the top result when given an image query.

BLEU (Bilingual Evaluation Understudy) [99] measures the precision of n-grams in the generated text that match n-grams in the reference text, and it is widely used for its strong correlation with human judgment.

METEOR (Metric for Evaluation of Translation with Explicit ORdering) [6] calculates a score based on the harmonic mean of unigram precision and recall, incorporating stemming and synonymy matching to correlate well with human judgment at the sentence level.

CIDEr (Consensus-based Image Description Evaluation) [134] evaluates image captions by measuring the similarity of a generated sentence to a set of ground truth sentences, emphasizing consensus among multiple references.

ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation) [77] uses the Longest Common Subsequence (LCS) between the generated text and the reference text to evaluate sentence-level structure similarity.

SPICE (Semantic Propositional Image Caption Evaluation) [3] compares the semantic propositional content of the generated text with the reference text, focusing on the meaning conveyed by the captions for a nuanced assessment of semantic accuracy.

5.6.4 Experimental Results

To evaluate the contribution of various components in our model, we conduct a series of experiments. Specifically, we compare the performance of the Euclidean and hyperbolic baselines with models that integrate Random Query Selection (RQS) and Random Text Pruning (RTP), as described in Sec. 5.5. Additionally, we provide the results of the hyperbolic baseline model using the Poincaré distance in contrastive learning. For the COCO zero-shot retrieval evaluation, we use the stage-1 pre-trained models, while for the image captioning evaluation, we employ the stage-2 models after fine-tuning them for the captioning downstream task. All models utilize a ViT-g vision encoder and, in the image captioning task, the OPT 2.7B language model. The results of these studies are summarized in Table 5.1. Below, we describe each setup and discuss its impact on the performance metrics.

Model	COCO Fine-tuned (5K test set)						COCO Fine-tune Karpathy test					
	TR@1	TR@5	TR@10	IR@1	IR@5	IR@10	B@1	B@4	M	R	C	S
Euclidean baseline	69.6	90.0	94.8	53.5	79.6	86.9	68.6	30.1	29.2	54.0	110.7	21.8
Euclidean RQS	72.1	**92.1**	**96.2**	**55.6**	80.9	87.8	77.3	36.8	29.2	57.9	127.5	23.1
Euclidean RQS+RTP	**72.2**	**92.1**	96.1	**55.6**	**81.2**	**88.4**	78.8	37.7	29.9	58.8	129.8	**23.4**
Hyperbolic baseline	67.0	88.0	91.5	51.2	77.5	85.0	65.0	28.5	28.0	52.5	105.0	20.5
Hyperbolic (Poincaré)	8.6	12.1	12.9	1.3	2.3	2.7	14.0	0.1	6.1	12.8	0.1	0.9
Hyperbolic RQS	71.3	91.4	95.8	54.2	80.7	87.1	77.5	37.1	28.7	58.1	128.8	23.2
Hyperbolic RTP	70.8	91.2	95.2	53.8	80.3	86.9	77.1	36.6	28.3	57.9	127.8	22.7
Hyperbolic RQS+RTP	71.9	91.6	96.0	54.5	80.2	87.7	**79.3**	**37.9**	**31.0**	**59.7**	**130.2**	23.3

Table 5.1. Performance comparison of various models on COCO fine-tuned 5K test set for zero-shot image-text retrieval (TR) and image retrieval (IR), and on the COCO Karpathy test set for image captioning evaluated with BLEU (B), METEOR (M), ROUGE-L (R), CIDEr (C), and SPICE (S). Models include Euclidean and hyperbolic baselines, with and without Random Query Selection (RQS) and Random Text Pruning (RTP). All models use the ViT-g vision encoder and OPT 2.7B language model.

Euclidean Baseline The Euclidean baseline model utilizes the BLIP-2 configuration without any hyperbolic modifications. This model serves as a reference point, providing performance benchmarks for both retrieval and captioning tasks. It

achieves solid performance in both zero-shot retrieval, with 69.6 for TR@1 and 53.5 for @ IR@1, and image captioning, with scores of 68.6 for BLEU-1, 30.1 for BLEU-4, 29.2 for METEOR, 54.0 for ROUGE-L, 110.7 for CIDEr, and 21.8 for SPICE.

Euclidean RQS This model incorporates Random Query Selection (RQS) into the Euclidean baseline. The results demonstrate improvements in most metrics, with a notable increase to 72.1 (+2.5) for TR@1, 55.6 (+2.1) for IR@1, 77.3 (+8.7) for BLUE-1, 36.8 (+6.7) for BLUE-4, and 127.5 (+16.8) for CIDEr, highlighting the benefit of this approach.

Euclidean RQS+RTP This model combines Euclidean Random Query selection (RQS) and Text Pruning (RTP) within the BLIP-2 baseline. The inclusion of RTP in combination with RQS further enhances performance, with scores such as 78.8 (+1.5) for BLEU-1, 37.7 (+0.9) for BLEU-4, 58.8 (+0.9) for ROUGE-L, and 129.8 (+2.3) for CIDEr, underscoring the effectiveness of RQS and RTP in the Euclidean space.

Hyperbolic Baseline The hyperbolic baseline model serves as the counterpart to the Euclidean baseline but operates within hyperbolic space. This setup allows us to directly compare the effectiveness of hyperbolic versus Euclidean representations in both retrieval and captioning tasks. The performance are slightly lower than the Euclidean counterpart. This is most probably due to the Euclidean pre-trained weights used to initialize the ViT encoder.

Hyperbolic baseline w/ Poincaré distance This version of the hyperbolic baseline model uses the Poincaré distance in the computation of the contrastive loss. This model shows a significant drop in performance across all metrics, with scores such as 30.0 for TR@1, 14.0 for BLEU-1, and 0.1 for CIDEr, indicating that using the Poincaré distance with a model pre-trained in Euclidean space creates too much instability.

Hyperbolic RQS In this ablation, Random Query selection (RQS) is applied at training time using the hyperbolic model. The results show improvements compared to the hyperbolic baseline, with scores of 71.3 for TR@1, 54.2 for IR@1, and 128.8

for CIDEr, demonstrating the benefits of RQS in hyperbolic space and shortening the gap with the Euclidean RQS model.

Hyperbolic RTP This model incorporates Random Text Pruning (RTP) at training time. This setup focuses on pruning text from the captions to improve model robustness. The results indicate enhancements in various metrics compared to the hyperbolic baseline, such as 70.8 for TR@1, 95.2 for TR@10, 53.8 for IR@1, and 127.8 for CIDEr, but are still slightly lower than the Hyperbolic RQS model.

Hyperbolic RQS+RTP This configuration combines both Random Query selection (RQS) and Random Text Pruning (RTP) within the hyperbolic space. It achieves slighter lower performance on zero-shot retrieval compared to the Euclidean RQS+RTP counterpart, but significant improvements on the image captioning task, with scores of 79.3 (+0.5) for BLEU-1, 37.9 (+0.2) for BLEU-4, 31.0 (+1.1) for METEOR, 59.7 (+0.9) for ROUGE-L, 130.2 (+2.7) for CIDEr, and 23.3 (-0.1) for SPICE. These results highlight the combined effectiveness of RQS and RTP in enhancing the performance of the hyperbolic model.

5.7 Discussion

This study investigated the integration of hyperbolic embeddings in the large-scale BLIP-2 vision-language model, marking the first application of hyperbolic embeddings at this scale. While hyperbolic space theoretically offers advantages for capturing hierarchical relationships and uncertainty, we show that conventional approaches like Poincaré distance face different challenges.

Our hyperbolic training method achieved performance comparable or superior to Euclidean models while providing image embeddings with varying hyperbolic radii. Conventional hyperbolic approaches struggle both in performance and in capturing meaningful uncertainty. Our study demonstrates that it is possible to maintain or improve performance while incorporating a measure of uncertainty in the embeddings.

Moreover, the newly introduced Random Query Selection and Random Text Pruning techniques are independent of the embedding space used, despite being

inspired by challenges faced when adapting BLIP-2 to hyperbolic space. Consequently, these techniques are broadly applicable and beneficial for Large Multimodal Models in general.

With our research, we want to highlight the limitations and the findings to advance the understanding of hyperbolic embeddings in VLMs. Future efforts should aim to refine hyperbolic methods to overcome scalability challenges and fully realize their potential in enhancing vision-language models.

Chapter 6

Conclusion

This thesis has explored the application of hyperbolic neural networks across various domains in machine learning, demonstrating their unique advantages in capturing hierarchical relationships, estimating uncertainty, and enhancing performance. Our research has spanned three main areas: self-supervised representation learning for skeleton-based action recognition, active domain adaptation for semantic segmentation, and multimodal vision-language models.

We have introduced HYSP [41], a novel self-supervised approach for learning skeleton-based action representations. HYSP has leveraged leverages the hyperbolic radius as a measure of uncertainty to self-paced learning process by dynamically scaling the gradients. This method has not only outperformed state-of-the-art approaches but has also eliminated the need for computationally expensive negative mining procedures.

Building on these insights, we have developed HALO [40], a pixel-level active learning approach for semantic segmentation under domain shift. In HALO, the hyperbolic radius is interpreted as an indicator of data scarcity and, combined with prediction entropy, highly correlates with epistemic uncertainty, which is employed for data acquisition. This novel approach has achieved state-of-the-art results and has been the first to surpass supervised domain adaptation performance using only a small fraction of labels.

Finally, we have explored the integration of hyperbolic embeddings into large-scale vision-language models [88], adapting the BLIP-2 architecture. Despite challenges in scaling hyperbolic representations to billions of parameters, we have

developed a novel training strategy that maintained performance comparable to Euclidean models while providing meaningful uncertainty estimates.

Throughout these studies, the hyperbolic radius has emerged as a valuable tool for quantifying uncertainty and data scarcity, providing insights not readily available in traditional Euclidean models. However, challenges remain in scaling hyperbolic neural networks to very large settings and ensuring training stability, highlighting important areas for future research. Ultimately, this thesis contributes to a broader understanding of hyperbolic neural networks and their potential to advance deep learning, illustrating the significant role of non-Euclidean geometries in artificial intelligence and offering promising avenues for more efficient, interpretable, and powerful learning algorithms.

Chapter 7

Future Work

This thesis has extensively investigated the potential of hyperbolic neural networks across diverse machine learning domains, paving the way for several promising directions of future research:

1. **Calibration of Hyperbolic Neural Networks**: Developing robust calibration techniques specifically tailored for hyperbolic space and investigating the relationship between hyperbolic radius and different types of uncertainty is crucial for enhancing the reliability and interpretability of HNNs, particularly in high-stakes applications.

2. **Fully-Hyperbolic Neural Network Architectures**: The development of fully-hyperbolic neural networks, where every layer operates in hyperbolic space, presents an exciting avenue for future work. This includes designing stable and efficient hyperbolic variants of common neural network layers, investigating initialization strategies and optimization techniques, and comparing performance and efficiency to hybrid and Euclidean models across various tasks.

3. **Unlearning in Hyperbolic Space**: The concept of unlearning – selectively removing information from trained models – presents intriguing possibilities in hyperbolic space. Future work could focus on developing hyperbolic-specific unlearning algorithms, investigating how the hierarchical nature of hyperbolic embeddings impacts the unlearning process, comparing the effectiveness and efficiency to Euclidean approaches, and exploring applications in federated learning and differential privacy.

4. **Scaling Hyperbolic Models**: As we have seen in our work on hyperbolic multimodal learning [88], scaling hyperbolic models to billions of parameters presents unique challenges. Future research should focus on developing more efficient training strategies for large-scale hyperbolic neural networks, and investigating the impact of model size on hyperbolic embeddings and uncertainty estimation. Advancements in this area could unlock the benefits of hyperbolic geometry for state-of-the-art AI systems across various domains.

Bibliography

[1] Alessandro Achille and Stefano Soatto. "Emergence of Invariance and Disentanglement in Deep Representations". In: *2018 Information Theory and Applications Workshop (ITA)*. 2018, pp. 1–9. DOI: 10.1109/ITA.2018.8503149.

[2] Jean-Baptiste Alayrac et al. "Flamingo: a visual language model for few-shot learning". In: *Advances in neural information processing systems* 35 (2022), pp. 23716–23736.

[3] Peter Anderson et al. "Spice: Semantic propositional image caption evaluation". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*. Springer. 2016, pp. 382–398.

[4] Jordan T Ash et al. "Deep batch active learning by diverse, uncertain gradient lower bounds". In: *arXiv preprint arXiv:1906.03671* (2019).

[5] Mina Ghadimi Atigh et al. "Hyperbolic image segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 4453–4462.

[6] Satanjeev Banerjee and Alon Lavie. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ed. by Jade Goldstein et al. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72. URL: https://aclanthology.org/W05-0909.

[7] Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. "Fully Hyperbolic Convolutional Neural Networks for Computer Vision". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=ekz1hN5QNh.

[8] Shai Ben-David et al. “A theory of learning from different domains”. In: *Machine Learning* 79 (2010), pp. 151–175.

[9] Yoshua Bengio et al. “Curriculum Learning”. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2009, pp. 41–48. ISBN: 9781605585161. DOI: 10.1145/1553374.1553380. URL: https://doi.org/10.1145/1553374.1553380.

[10] Silvère Bonnabel. “Stochastic Gradient Descent on Riemannian Manifolds”. In: *IEEE Transactions on Automatic Control* 58.9 (2013), pp. 2217–2229. DOI: 10.1109/TAC.2013.2254619.

[11] David Brüggemann et al. “Refign: Align and refine for adaptation of semantic segmentation to adverse conditions”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 3174–3184.

[12] James W Cannon et al. “Hyperbolic geometry”. In: *Flavors of geometry* 31.59-115 (1997), p. 2.

[13] Mathilde Caron et al. “Deep Clustering for Unsupervised Learning of Visual Features”. In: *European Conference on Computer Vision*. 2018.

[14] Mathilde Caron et al. “Unsupervised Learning of Visual Features by Contrasting Cluster Assignments”. In: *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[15] Eduardo D. C. Carvalho et al. “Scalable Uncertainty for Computer Vision With Functional Variational Inference”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12000–12010. DOI: 10.1109/CVPR42600.2020.01202.

[16] Ines Chami et al. “From Trees to Continuous Embeddings and Back: Hyperbolic Hierarchical Clustering”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[17] Ines Chami et al. “From trees to continuous embeddings and back: Hyperbolic hierarchical clustering”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15065–15076.

[18] Ines Chami et al. “Hyperbolic graph convolutional neural networks”. In: *Advances in neural information processing systems* 32 (2019), pp. 4868–4879.

[19] Bike Chen et al. "Hyperbolic Uncertainty Aware Semantic Segmentation". In: *IEEE Transactions on Intelligent Transportation Systems* 25.2 (2024), pp. 1275–1290. DOI: 10.1109/TITS.2023.3312290.

[20] Bike Chen et al. "Hyperbolic uncertainty aware semantic segmentation". In: *arXiv preprint arXiv:2203.08881* (2022).

[21] Bike Chen et al. "Hyperbolic uncertainty aware semantic segmentation". In: *IEEE Transactions on Intelligent Transportation Systems* (2023).

[22] Liang-Chieh Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (Apr. 2018), pp. 834–848.

[23] Liang-Chieh Chen et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *Computer Vision – ECCV 2018*. Cham, 2018, pp. 833–851.

[24] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 1597–1607. URL: https://proceedings.mlr.press/v119/chen20j.html.

[25] Ting Chen et al. "A simple framework for contrastive learning of visual representations". In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org, 2020.

[26] Xinlei Chen and Kaiming He. "Exploring Simple Siamese Representation Learning". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 15745–15753. URL: https://api.semanticscholar.org/CorpusID:227118869.

[27] Xinlei Chen and Kaiming He. "Exploring Simple Siamese Representation Learning". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).

[28] Xinlei Chen et al. "Improved Baselines with Momentum Contrastive Learning". In: *ArXiv* abs/2003.04297 (2020).

[29] Yuxin Chen et al. "Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13359–13368.

[30] Zhan Chen et al. “Multi-Scale Spatial Temporal Graph Convolutional Network for Skeleton-Based Action Recognition”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.2 (May 2021), pp. 1113–1122.

[31] Liu Chunhui et al. “PKU-MMD: A Large Scale Benchmark for Continuous Multi-Modal Human Action Understanding”. In: *arXiv preprint arXiv:1703.07475* (2017).

[32] Marius Cordts et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *CoRR* abs/1604.01685 (2016). arXiv: 1604.01685.

[33] Wenliang Dai et al. *InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning*. 2023. arXiv: 2305.06500 [cs.CV].

[34] Stefan Depeweg et al. “Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and Risk-sensitive Learning”. In: *International Conference on Machine Learning*. 2017.

[35] Karan Desai et al. “Hyperbolic image-text representations”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 7694–7731.

[36] Bhuwan Dhingra et al. “Embedding Text in Hyperbolic Spaces”. In: *TextGraphs@NAACL-HLT*. 2018.

[37] Aleksandr Ermolov et al. “Hyperbolic Vision Transformers: Combining Improvements in Metric Learning”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[38] R. A. Fisher. “Inverse Probability”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 26.4 (1930), pp. 528–535. DOI: 10.1017/S0305004100016297.

[39] A. Flaborea et al. “Are we certain it’s anomalous?” In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2023, pp. 2897–2907. DOI: 10.1109/CVPRW59228.2023.00291.

[40] Luca Franco et al. “Hyperbolic Active Learning for Semantic Segmentation under Domain Shift”. In: *Forty-first International Conference on Machine Learning*. 2024. URL: https://openreview.net/forum?id=hKdJPMQvew.

[41] Luca Franco et al. “Hyperbolic Self-paced Learning for Self-supervised Skeleton-based Action Representations”. In: *The Eleventh International Conference on Learning Representations*. 2023.

[42] Geoffrey French, Michal Mackiewicz, and Mark Fisher. "Self-ensembling for visual domain adaptation". In: *arXiv preprint arXiv:1706.05208* (2017).

[43] Bo Fu et al. "Transferable query selection for active domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 7272–7281.

[44] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. "Deep bayesian active learning with image data". In: *International conference on machine learning*. PMLR. 2017, pp. 1183–1192.

[45] Octavian Ganea, Gary Becigneul, and Thomas Hofmann. "Hyperbolic Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/dbab2adc8f9d078009ee3fa810bea142-Paper.pdf.

[46] Songwei Ge et al. "Hyperbolic Contrastive Learning for Visual Representations beyond Objects". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

[47] S. Gidaris, P. Singh, and N. Komodakis. "Unsupervised representation learning by predicting image rotations". In: *ICLR*. 2018.

[48] S Alireza Golestaneh and Kris M Kitani. "Importance of Self-Consistency in Active Learning for Semantic Segmentation". In: *BMVC* (2020).

[49] Jean-Bastien Grill et al. "Bootstrap your own latent a new approach to self-supervised learning". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. 2020.

[50] Jean-Bastien Grill et al. "Bootstrap Your Own Latent: A new approach to self-supervised learning". In: *Neural Information Processing Systems*. Montréal, Canada, 2020.

[51] Caglar Gulcehre et al. "Hyperbolic attention networks". In: *arXiv preprint arXiv:1805.09786* (2018).

[52] Tianyu Guo et al. "Contrastive Learning from Extremely Augmented Skeleton Sequences for Self-supervised Action Recognition". In: *AAAI*. 2022.

[53] Y. Guo et al. "Clipped Hyperbolic Classifiers Are Super-Hyperbolic Classifiers". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2022, pp. 1–10.

[54] Yunhui Guo et al. “Clipped Hyperbolic Classifiers Are Super-Hyperbolic Classifiers”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 1–10. URL: https://api.semanticscholar.org/CorpusID:244731271.

[55] Yunhui Guo et al. “Clipped Hyperbolic Classifiers Are Super-Hyperbolic Classifiers”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11–20.

[56] Yunhui Guo et al. “Clipped Hyperbolic Classifiers Are Super-Hyperbolic Classifiers”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022.

[57] Yunhui Guo et al. “Free Hyperbolic Neural Networks with Limited Radii”. In: *ArXiv* abs/2107.11472 (2021).

[58] Kaiming He et al. “Momentum Contrast for Unsupervised Visual Representation Learning”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[59] Judy Hoffman et al. “Cycada: Cycle-consistent adversarial domain adaptation”. In: *International conference on machine learning*. Pmlr. 2018, pp. 1989–1998.

[60] Stephen C. Hora. “Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management”. In: *Reliability Engineering & System Safety* 54 (1996), pp. 217–223. URL: https://api.semanticscholar.org/CorpusID:111162869.

[61] Lukas Hoyer et al. “MIC: Masked image consistency for context-enhanced domain adaptation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 11721–11732.

[62] Eyke Hüllermeier and Willem Waegeman. “Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods”. In: *Machine Learning* 110 (Mar. 2021). DOI: 10.1007/s10994-021-05946-3.

[63] Sarah Ibrahimi et al. “Intriguing Properties of Hyperbolic Embeddings in Vision-Language Models”. In: *Transactions on Machine Learning Research* (2024).

[64] Lu Jiang et al. “Self-Paced Learning with Diversity”. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014.

[65] Pin Jiang et al. "Bidirectional Adversarial Training for Semi-Supervised Domain Adaptation". In: *IJCAI*. 2020, pp. 934–940.

[66] Alex Kendall and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Neural Information Processing Systems*. 2017.

[67] Valentin Khrulkov et al. "Hyperbolic Image Embeddings". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[68] Valentin Khrulkov et al. "Hyperbolic image embeddings". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6418–6428.

[69] Hanul Kim et al. "Efficient Action Recognition via Dynamic Knowledge Propagation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 13719–13728.

[70] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. *BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning*. 2019. arXiv: 1906.08158 [cs.LG].

[71] Max Kochurov, Rasul Karimov, and Serge Kozlukov. "Geoopt: Riemannian Optimization in PyTorch". In: *CoRR* abs/2005.02819 (2020). arXiv: 2005.02819. URL: https://arxiv.org/abs/2005.02819.

[72] M. Kumar, Benjamin Packer, and Daphne Koller. "Self-Paced Learning for Latent Variable Models". In: *Advances in Neural Information Processing Systems*. Ed. by J. Lafferty et al. Vol. 23. Curran Associates, Inc., 2010. URL: https://proceedings.neurips.cc/paper/2010/file/e57c6b956a6521b28495f2886ca0977a-Paper.pdf.

[73] Xin Lai et al. "Lisa: Reasoning segmentation via large language model". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 9579–9589.

[74] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Neural Information Processing Systems*. 2016. URL: https://api.semanticscholar.org/CorpusID:6294674.

[75] Junnan Li et al. "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models". In: *International conference on machine learning*. PMLR. 2023, pp. 19730–19742.

[76] Linguo Li et al. “3D Human Action Representation Learning via Cross-View Consistency Pursuit”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[77] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[78] Lilang Lin et al. “MS2L: Multi-Task Self-Supervised Learning for Skeleton Based Action Recognition”. In: *Proceedings of the 28th ACM International Conference on Multimedia*. MM ’20. Seattle, WA, USA: Association for Computing Machinery, 2020, pp. 2490–2498. ISBN: 9781450379885. DOI: 10.1145/3394171.3413548. URL: https://doi.org/10.1145/3394171.3413548.

[79] Lilang Lin et al. “MS2L: Multi-Task Self-Supervised Learning for Skeleton Based Action Recognition”. In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 2490–2498.

[80] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context”. In: *CoRR* abs/1405.0312 (2014). arXiv: 1405.0312. URL: http://arxiv.org/abs/1405.0312.

[81] Haotian Liu et al. “Visual instruction tuning”. In: *Advances in neural information processing systems* 36 (2024).

[82] Jun Liu et al. “NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.10 (2020), pp. 2684–2701.

[83] Qi Liu, Maximilian Nickel, and Douwe Kiela. “Hyperbolic Graph Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/103303dd56a731e377d01f6a37badae3-Paper.pdf.

[84] Qi Liu, Maximilian Nickel, and Douwe Kiela. “Hyperbolic graph neural networks”. In: *Advances in neural information processing systems* 32 (2019).

[85] Yuang Liu, Wei Zhang, and Jun Wang. “Source-free domain adaptation for semantic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1215–1224.

[86] Zhizhe Liu et al. "Margin Preserving Self-paced Contrastive Learning Towards Domain Adaptation for Medical Image Segmentation". In: *CoRR* abs/2103.08454 (2021). arXiv: 2103.08454. URL: https://arxiv.org/abs/2103.08454.

[87] Ziwei Liu et al. "Open compound domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12406–12415.

[88] Paolo Mandica et al. "Hyperbolic Learning with Multimodal Large Language Models". In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2024.

[89] Ke Mei et al. "Instance adaptive self-training for unsupervised domain adaptation". In: *Computer Vision–ECCV 2020: 16th European Conference Proceedings, Part XXVI 16*. 2020, pp. 415–430.

[90] Pascal Mettes et al. "Hyperbolic deep learning in computer vision: A survey". In: *International Journal of Computer Vision* (2024), pp. 1–25.

[91] Rhiannon Michelmore et al. "Uncertainty Quantification with Statistical Guarantees in End-to-End Autonomous Driving Control". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 7344–7350. DOI: 10.1109/ICRA40945.2020.9196844.

[92] Sudhanshu Mittal et al. *Best Practices in Active Learning for Semantic Segmentation*. 2023. arXiv: 2302.04075 [cs.CV].

[93] Olivier Moliner, Sangxia Huang, and Kalle Åström. "Bootstrapped Representation Learning for Skeleton-Based Action Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2022, pp. 4154–4164.

[94] Maximilian Nickel and Douwe Kiela. *Poincaré Embeddings for Learning Hierarchical Representations*. 2017. arXiv: 1705.08039 [cs.AI].

[95] Munan Ning et al. "Multi-anchor active domain adaptation for semantic segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9112–9122.

[96] Mehdi Noroozi and Paolo Favaro. "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles". In: *ECCV*. 2016.

[97] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. *Representation Learning with Contrastive Predictive Coding*. 2018. DOI: 10.48550/ARXIV.1807.03748. URL: https://arxiv.org/abs/1807.03748.

[98] Giancarlo Paoletti et al. "Unsupervised Human Action Recognition with Skeletal Graph Laplacian and Self-Supervised Viewpoints Invariance". In: *The 32nd British Machine Vision Conference (BMVC)*. 2021.

[99] Kishore Papineni et al. "BLEU: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://doi.org/10.3115/1073083.1073135.

[100] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 8024–8035.

[101] Sujoy Paul et al. *Domain Adaptive Semantic Segmentation Using Weak Labels*. 2020. arXiv: 2007.15176 [cs.CV].

[102] Jizong Peng et al. "Self-Paced Contrastive Learning for Semi-supervised Medical Image Segmentation with Meta-labels". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 16686–16699. URL: https://proceedings.neurips.cc/paper/2021/file/8b5c8441a8ff8e151b191c53c1842a38-Paper.pdf.

[103] Wei Peng et al. "Hyperbolic Deep Neural Networks: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.12 (2022), pp. 10023–10044. DOI: 10.1109/TPAMI.2021.3136921.

[104] Wei Peng et al. "Mix Dimension in Poincaré Geometry for 3D Skeleton-based Action Recognition". In: *Proceedings of the 28th ACM International Conference on Multimedia* (2020).

[105] Viraj Prabhu et al. "Active domain adaptation via clustering uncertainty-weighted embeddings". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8505–8514.

[106] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[107] Haocong Rao et al. "Augmented Skeleton Based Contrastive Action Learning with Momentum LSTM for Unsupervised Action Recognition". In: *Information Sciences* 569 (2021), pp. 90–109. DOI: https://doi.org/10.1016/j.ins.2021.04.023.

[108] Stephan R. Richter et al. "Playing for Data: Ground Truth from Computer Games". In: *European Conference on Computer Vision (ECCV)*. Vol. 9906. 2016, pp. 102–118.

[109] German Ros et al. "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[110] Kuniaki Saito et al. "Semi-Supervised Domain Adaptation via Minimax Entropy". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[111] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. "ACDC: The Adverse Conditions Dataset with Correspondences for Semantic Driving Scene Understanding". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021.

[112] Frederic Sala et al. "Representation tradeoffs for hyperbolic embeddings". In: *International conference on machine learning*. PMLR. 2018, pp. 4460–4469.

[113] Enver Sangineto et al. "Self Paced Deep Learning for Weakly Supervised Object Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.3 (2019), pp. 712–725. DOI: 10.1109/TPAMI.2018.2804907.

[114] Rik Sarkar. "Low distortion delaunay embedding of trees in hyperbolic plane". In: *Graph Drawing: 19th International Symposium, GD 2011, Revised Selected Papers 19*. 2012, pp. 355–366.

[115] Ozan Sener and Silvio Savarese. "Active learning for convolutional neural networks: A core-set approach". In: *arXiv preprint arXiv:1708.00489* (2017).

[116] Amir Shahroudy et al. "NTU RGB+D: A large scale dataset for 3D human activity analysis". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1010–1019.

[117] Ryohei Shimizu, YUSUKE Mukuta, and Tatsuya Harada. "Hyperbolic Neural Networks++". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=Ec85b0tUwbA.

[118] Gyungin Shin, Weidi Xie, and Samuel Albanie. "All you need are a few pixels: semantic segmentation with pixelpick". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1687–1697.

[119] Inkyu Shin et al. "Labor: Labeling only if required for domain adaptive semantic segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8588–8598.

[120] Connor Shorten and Taghi M. Khoshgoftaar. "A survey on Image Data Augmentation for Deep Learning". In: *Journal of Big Data* 6 (2019), p. 60. ISSN: 2196-1115. DOI: https://doi.org/10.1186/s40537-019-0197-0.

[121] Ankit Singh. "CLDA: Contrastive Learning for Semi-Supervised Domain Adaptation". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 5089–5101.

[122] Ankit Singh et al. "Semi-Supervised Action Recognition With Temporal Contrastive Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 10389–10399.

[123] Anurag Singh et al. "Improving Semi-Supervised Domain Adaptation Using Effective Target Selection and Semantics". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2021, pp. 2703–2712.

[124] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. "Variational adversarial active learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5972–5981.

[125] Max van Spengler, Erwin Berkhout, and Pascal Mettes. "Poincaré ResNet". In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), pp. 5396–5405. URL: https://api.semanticscholar.org/CorpusID:257757355.

[126] Max van Spengler, Erwin Berkhout, and Pascal Mettes. *Poincaré ResNet*. 2023. arXiv: 2303.14027 [cs.CV].

[127] Jong-Chyi Su et al. "Active adversarial domain adaptation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 739–748.

[128] Kun Su, Xiulong Liu, and Eli Shlizerman. "Predict & cluster: Unsupervised skeleton based action recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9631–9640.

[129] Yukun Su, Guosheng Lin, and Qingyao Wu. "Self-Supervised 3D Skeleton Action Representation Learning With Motion Consistency and Continuity". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 13328–13338.

[130] Dídac Surís, Ruoshi Liu, and Carl Vondrick. "Learning the Predictability of the Future". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12607–12617.

[131] Fida Mohammad Thoker, Hazel Doughty, and Cees G. M. Snoek. "Skeleton-Contrastive 3D Action Representation Learning". In: *Proceedings of the 29th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, 2021.

[132] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. "Poincaré glove: Hyperbolic word embeddings". In: *arXiv preprint arXiv:1810.06546* (2018).

[133] M. Valdenegro-Toro and D. Mori. "A Deeper Look into Aleatoric and Epistemic Uncertainty Disentanglement". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2022, pp. 1508–1516. DOI: 10.1109/CVPRW56347.2022.00157.

[134] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. "Cider: Consensus-based image description evaluation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4566–4575.

[135] Tuan-Hung Vu et al. "ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[136] Dan Wang and Yi Shang. "A new active labeling method for deep learning". In: *2014 International joint conference on neural networks (IJCNN)*. IEEE. 2014, pp. 112–119.

[137] Keze Wang et al. "Cost-effective active learning for deep image classification". In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.12 (2016), pp. 2591–2600.

[138] Minsi Wang, Bingbing Ni, and Xiaokang Yang. "Learning Multi-View Interactional Skeleton Graph for Action Recognition." In: *IEEE transactions on pattern analysis and machine intelligence* PP (2020).

[139] Ping Wang et al. "Self-paced and self-consistent co-training for semi-supervised image segmentation". In: *Medical Image Analysis* 73 (2021), p. 102146. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2021.102146.

[140] Tsung-Han Wu et al. "D 2 ADA: Dynamic Density-Aware Active Domain Adaptation for Semantic Segmentation". In: *Computer Vision ECCV 2022 Proceedings, Part XXIX*. Springer. 2022, pp. 449–467.

[141] Tsung-Han Wu et al. "Redal: Region-based and diversity-aware active learning for point cloud semantic segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15510–15519.

[142] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. "When Do Curricula Work?" In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=tW4QEInpni.

[143] Yijun Xiao and William Wang. "Quantifying Uncertainties in Natural Language Processing Tasks". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (July 2019), pp. 7322–7329. DOI: 10.1609/aaai.v33i01.33017322.

[144] Binhui Xie et al. "Active learning for domain adaptation: An energy-based approach". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 2022, pp. 8708–8716.

[145] Binhui Xie et al. "Towards Fewer Annotations: Active Learning via Region Impurity and Prediction Uncertainty for Domain Adaptive Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 8068–8078.

[146] Enze Xie et al. "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers". In: *Neural Information Processing Systems (NeurIPS)*. 2021.

[147] Jingwei Xu et al. "Deep Kinematics Analysis for Monocular 3D Human Pose Estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[148] Jiexi Yan et al. "Unsupervised Hyperbolic Metric Learning". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 12460–12469. DOI: 10.1109/CVPR46437.2021.01228.

[149] Jiexi Yan et al. "Unsupervised hyperbolic metric learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12465–12474.

[150] Sijie Yan, Yuanjun Xiong, and Dahua Lin. "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, 2018.

[151] Siyuan Yang et al. "Skeleton Cloud Colorization for Unsupervised 3D Action Representation Learning". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), pp. 13403–13413.

[152] Yanchao Yang and Stefano Soatto. "FDA: Fourier Domain Adaptation for Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[153] Qinghao Ye et al. "mplug-owl: Modularization empowers large language models with multimodality". In: *arXiv preprint arXiv:2304.14178* (2023).

[154] Shukang Yin et al. "A survey on multimodal large language models". In: *arXiv preprint arXiv:2306.13549* (2023).

[155] Bing Yu, Haoteng Yin, and Zhanxing Zhu. "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 3634–3640. DOI: 10.24963/ijcai.2018/505. URL: https://doi.org/10.24963/ijcai.2018/505.

[156] Dingwen Zhang et al. "Bridging Saliency Detection to Weakly Supervised Object Detection Based on Self-Paced Curriculum Learning". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI'16. New York, New York, USA: AAAI Press, 2016, pp. 3538–3544. ISBN: 9781577357704.

[157] Richard Zhang, Phillip Isola, and Alexei A Efros. "Colorful Image Colorization". In: *ECCV*. 2016.

[158] Susan Zhang et al. "Opt: Open pre-trained transformer language models". In: *arXiv preprint arXiv:2205.01068* (2022).

[159] Nenggan Zheng et al. "Unsupervised representation learning with long-term dynamics for skeleton based action recognition". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[160] Deyao Zhu et al. "MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models". In: *arXiv preprint arXiv:2304.10592* (2023).