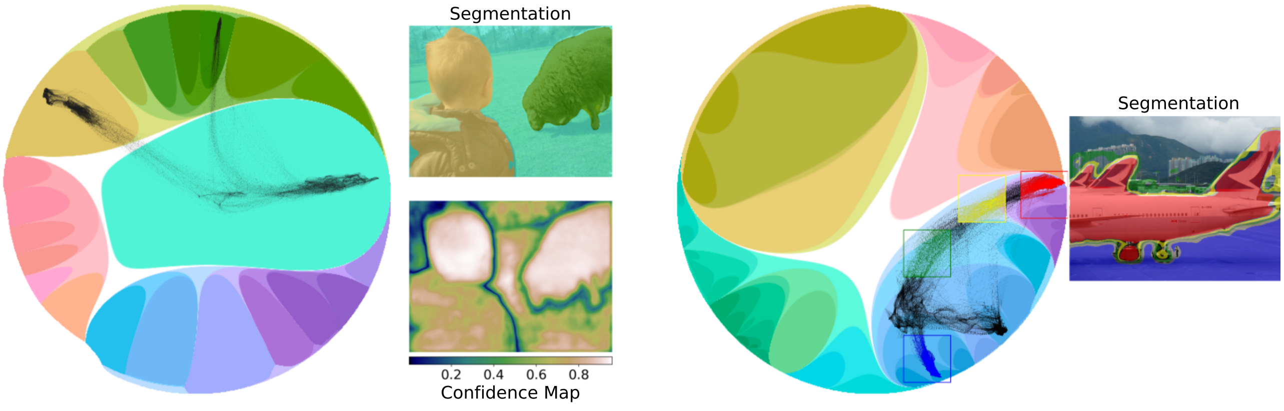
Hyperbolic Image Segmentation

Mina GhadimiAtigh[1]*, Julian Schoep[1]*, Erman Acar[2], Nanne van Noord[1], Pascal Mettes[1]
[1]University of Amsterdam, [2]Leiden University, [2]Vrije Universiteit Amsterdam



(a) Prediction uncertainty for free

(b) Boundary information for free

Figure 1. **Two examples of insights that come for free with Hyperbolic Image Segmentation.** For both examples, each black dot denotes a pixel embedding in hyperbolic space. Left (Pascal VOC): next to per-pixel classification, the distance to the origin in hyperbolic space provides a free measure of uncertainty. Right (COCO-Stuff-10k): the hyperbolic positioning of pixels even allows us to pinpoint interiors and edges of objects, as indicated by the colored boxes and their corresponding pixels in the segmentation map. Other benefits of hyperbolic embeddings for segmentation include zero-label generalization and better performance in low-dimensional embedding spaces.

Abstract

For image segmentation, the current standard is to perform pixel-level optimization and inference in Euclidean output embedding spaces through linear hyperplanes. In this work, we show that hyperbolic manifolds provide a valuable alternative for image segmentation and propose a tractable formulation of hierarchical pixel-level classification in hyperbolic space. Hyperbolic Image Segmentation opens up new possibilities and practical benefits for segmentation, such as uncertainty estimation and boundary information for free, zero-label generalization, and increased performance in low-dimensional output embeddings.

1. Introduction

A ubiquitous goal in visual representation learning is to obtain discriminative and generalizable embeddings. Such visual embeddings are learned in a deep and highly non-linear fashion. On top, a linear layer separates categories through Euclidean hyperplanes. The choice for a zero curvature Euclidean embedding space, although a *de facto* standard, requires careful re-consideration as it has direct consequences for how well a task can be optimized given the latent structure that is inherently present in both the data and the category space [19, 22, 29].

This work takes inspiration from recent literature advocating hyperbolic manifolds as embedding spaces for machine learning and computer vision tasks. Foundational work showed that hyperbolic manifolds are able to embed hierarchies and tree-like structures with minimal distortion [29]. Follow up work has demonstrated the benefits of hyperboles for various tasks with latent hierarchical structures, from text embedding [42, 55] to graph inference [8, 12, 22]. Notably, Khrulkov *et al.* [19] showed that hyperbolic embeddings also have profound connections to visual data, due to latent hierarchical structures present in vision datasets. This connection has brought along early hyperbolic success in computer vision for few-shot and zero-shot learning [15, 19, 23], unsupervised learning [32, 46], and video recognition [25, 40].

Common amongst current hyperbolic computer vision works is that the task at hand is global, *i.e.* an entire image or video is represented by a single vector in the hy-

* Equal contribution

perbolic embedding space [3, 19, 23, 25]. Here, our goal is to take hyperbolic deep learning to the pixel level. This generalization is however not trivial. The change of manifold brings different formulations for basic operations such as addition and multiplication, each with different spatial complexity. Specifically, the additional spatial complexity that comes with the Möbius addition as part of the hyperbolic multinomial logistic regression makes it intractable to simultaneously optimize or infer all pixels of even a single image. Here, we propose an equivalent re-formulation of multinomial logistic regression in the Poincaré ball that bypasses the explicit computation of the Möbius addition, allowing for simultaneous segmentation optimization on batches of images in hyperbolic space. We furthermore outline how to incorporate hierarchical knowledge amongst labels in the hyperbolic embedding space, as previously advocated in image and video recognition [23, 25]. The proposed approach is general and can be plugged on top of any segmentation architecture. The code is available at https://github.com/MinaGhadimiAtigh/HyperbolicImageSegmentation.

We perform a number of analyses to showcase the effect and new possibilities that come with Hyperbolic Image Segmentation. We present the following: *(i)* Hyperbolic embeddings provide natural measures for uncertainty estimation and for semantic boundary estimation in image segmentation, see Figure 1. Different from Bayesian uncertainty estimation, our approach requires no additional parameters or multiple forward passes, *i.e.* this information comes for free. *(ii):* Hyperbolic embeddings with hierarchical knowledge provide better zero-label generalization than Euclidean counterparts, *i.e.* hyperboles improve reasoning over unseen categories. *(iii):* Hyperbolic embeddings are preferred for fewer embedding dimensions. Low-dimensional effectiveness is a cornerstone in hyperbolic deep learning [29]. We find that these benefits extend to image segmentation, with potential for explainability and on-device segmentation [3]. We believe these findings bring new insights and opportunities to image segmentation.

2. Related work

2.1. Image segmentation

Widely used segmentation approaches follow the encoder-decoder paradigm, where an encoder learns lower-dimensional representations and the decoders serves to reconstruct high-resolution segmentation maps [5, 9, 10, 24, 31, 34]. Early adaptations of decoders used parametrized upsampling operations through deconvolutions [24, 31] or multiple blocks of a bi-linear upsampling followed by more convolutional layers [5]. More recent works seek to reinforce the upsampling with context information by merging feature maps at various scales, *i.e.* feature pyramids [52], or by combining the decoding with global context features through fully connected layers [49]. For example, the widely adapted Deeplab architecture [9] uses atrous convolutions with various levels of dilation within the decoder to effectively obtain context information at various scales. Other recent approaches focus on improving the utilization of multi-scale information, *e.g.* using multi-scale attention [41], squeeze-and-attention [53], and Transformers [48]. Commonly in semantic image segmentation, the final classification is performed through multinomial logistic regression in Euclidean space. As a promising alternative, we advocate for using the hyperbolic space to perform pixel-level classification on top of any existing architecture.

2.2. Hyperbolic deep learning

The hyperbolic space has gained traction in deep learning literature for representing tree-like structures and taxonomies [18, 20, 29, 30, 36, 38, 47], text [2, 42, 55], and graphs [4, 8, 12, 22, 26, 50]. Hyperbolic alternatives have been proposed for various network layers, from intermediate layers [17, 39] to classification layers [3, 11, 17, 39]. Recently, hyperboles have also been applied in computer vision for hierarchical action search [25], few-shot learning [19], hierarchical image classification [13], and zero-shot image recognition [23]. In this work, we build upon these foundations and make the step towards semantic image segmentation, which requires a reformulation of the hyperbolic multinomial logistic regression to become tractable.

Previous works have shown the potential of a hierarchical view on image segmentation. For instance, [51] incorporate an open vocabulary perspective based on WordNet [27] hypernym/hyponym relations. By learning a joint-embedding of image features and word concepts, combined with a dedicated scoring function to enforce the asymmetric relation between hypernyms and hyponyms, their model is able to predict hierarchical concepts. This approach is akin to that of [21] who use hierarchy-level specific convolutional blocks. These blocks, individually tasked with discriminating only between child classes, are dynamically activated such that only a subset of the entire graph is activated at any given time depending on which concepts are present in the image. This is trained with a loss function consisting of a sum of binary cross-entropy losses at each of the child-concept prediction maps. Here, we seek to incorporate hierarchical information on the hyperbolic manifolds, which can be applied on top of any segmentation architecture without needing to change the architecture itself.

Recent work by [44] investigated the use of the hyperbolic space for instance segmentation in images, but only do so after the fact, *i.e.* on top of predicted instance segmentations. In contrast, our approach enables tractable hyperbolic classification as part of the pixel-level segmentation itself.

3. Image segmentation on the hyperbole

3.1. Background: The Poincaré ball model

Hyperbolic geometry encompases several conformal models [7]. Based on its widespread use in deep learning and computer vision, we operate on the Poincaré ball. The Poincaré ball is defined as $(\mathbb{D}^n_c, g^{\mathbb{D}_c})$, with manifold $\mathbb{D}^n_c = \{x \in \mathbb{R}^n : c||x|| < 1\}$ and Riemannian metric:

$$g^{\mathbb{D}_c}_x = (\lambda^c_x)^2 g^E = \frac{2}{1 - c||x||^2}\mathbb{I}^n, \tag{1}$$

where $g^E = \mathbb{I}^n$ denotes the Euclidean metric tensor and c is a hyperparameter governing the curvature and radius of the ball. Segmentation networks operate in Euclidean space and to be able to operate on the Poincaré ball, a mapping from the Euclidean tangent space to the hyperbolic space is required. The projection of a Euclidean vector x onto the Poincaré ball is given by the exponential map with anchor v:

$$\exp^c_v(x) = v \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda^c_v||x||}{2}\right)\frac{x}{\sqrt{c}||x||}\right), \tag{2}$$

with $\oplus_c$ the Möbius addition:

$$v \oplus_c w = \frac{(1 + 2c\langle v, w\rangle + c||w||^2)v + (1 - c||v||^2)w}{1 + 2c\langle v, w\rangle + c^2||v||^2||w||^2}. \tag{3}$$

In practice, $\mathbf{v}$ is commonly set to the origin, simplifying the exponential map to

$$\exp_0(x) = \tanh(\sqrt{c}||x||)(x/(\sqrt{c}||x||)). \tag{4}$$

3.2. Tractable pixel-level hyperbolic classification

For the problem of image segmentation, we are given an input image $X \in \mathbb{R}^{w\times h\times 3}$, with w and h the width and height of the image respectively. For each pixel $x \in X$, we need to assign a label $y \in Y$, where Y denotes a set of C class labels. Let $f(X) : \mathbb{R}^{w\times h\times 3} \mapsto \mathbb{R}^{w\times h\times n}$ denote an arbitrary function that transforms each pixel to a n-dimensional representation, *e.g.* an image-to-image network. Common amongst current approaches is to feed all pixels in parallel to a linear layer followed by a softmax, resulting in a C-dimensional probability distribution over all C classes per pixel, optimized with cross-entropy.

This paper advocates the use of the hyperbolic space to perform the per-pixel classification for image segmentation. We start from the geometric interpretation of the hyperbolic multinomial logistic regression given by Ganea *et al.* [18], which defines the gyroplane, *i.e.* the hyperplane in the Poincaré ball, as:

$$H^c = \{z_{ij} \in \mathbb{D}^n_c, \langle -p \oplus_c z_{ij}, w\rangle = 0\}, \tag{5}$$

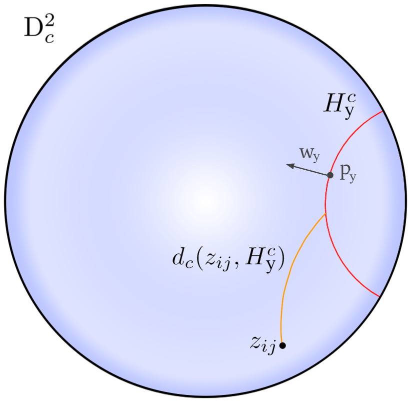


Figure 2. Visualization of the hyperbolic gyroplane (p_y, w_y) and distance to output z_{ij} on a two-dimensional manifold. In the context of this work, z_{ij} denotes the output representation at pixel location (i, j) and H^c_y denotes the hyperplane for class y.

where $z_{ij} = \exp_0(f(X)_{ij})$ denotes the exponential map of the network output at pixel location (i, j) and with $p \in \mathbb{D}^n_c$ the offset and $w \in \mathcal{T}_p\mathbb{D}^n_c$ the orientation of the gyroplane. The hyperbolic distance of z_{ij} to the gyroplane of class y is given as:

$$d_c(z_{ij}, H^c_y) = \frac{1}{\sqrt{c}}\sinh^{-1}\left(\frac{2\sqrt{c}\langle -p_y \oplus_c z_{ij}, w_y\rangle}{(1-c||-p_y \oplus_c z_{ij}||^2)||w_y||}\right). \tag{6}$$

Figure 2 illustrates a gyroplane on the hyperbole defined by its offset and orientation, along with the geodesic from pixel output z_{ij} to the gyroplane. Based on this distance, the logit of class y for pixel output z_{ij} using the metric of Equation 1 is given as:

$$\zeta_y(z_{ij}) = \frac{\lambda^c_{p_y}||w_y||}{\sqrt{c}}\sinh^{-1}\left(\frac{2\sqrt{c}\langle -p_y \oplus_c z_{ij}, w_y\rangle}{(1-c||-p_y \oplus_c z_{ij}||^2)||w_y||}\right). \tag{7}$$

Consequently, the likelihood is given as:

$$p(\hat{y} = y|z_{ij}) \propto \exp(\zeta_y(z_{ij})), \tag{8}$$

which can be optimized with the cross-entropy loss and gradient descent.

The geometric interpretation of Ganea *et al.* [18] provides a framework for classifying output vectors in hyperbolic space. In contrast to standard classification, image segmentation requires per-pixel classification in parallel. This setup is however intractable for the current implementation of hyperbolic multinomial logistic regression. The bottleneck is formed by the explicit computation of the Möbius addition. In a standard example segmentation setting ($W = H = 513$, $K = 100$ classes, $n = 256$, and batch size 5), this would induce a memory footprint of roughly 132 GB in 32-bit float precision, compared to roughly 0.5 GB in Euclidean space. Here, we propose an equivalent computation of the margin likelihood by factoring out the explicit computation of the Möbius addition, resulting in a memory footprint of 1.1 GB. The key to our approach is the observation that we do not need the actual result of the addition, only its inner product in the numerator of Equation 7 $\langle -p_y \oplus_c z_{ij}, w_y\rangle$ and its squared norm in the denominator $|| - p_y \oplus_c z_{ij}||^2$.

To that end, we first rewrite the Möbius addition as:

$$\begin{aligned}
\hat{p}_y \oplus_c z_{ij} &= \alpha\hat{p}_y + \beta z_{ij}, \\
\alpha &= \frac{1 + 2c\langle\hat{p}_y, z_{ij}\rangle + c||z_{ij}||^2}{1 + 2c\langle\hat{p}_y, z_{ij}\rangle + c^2||\hat{p}_y||^2||z_{ij}||^2}, \\
\beta &= \frac{1 - c||\hat{p}_y||^2}{1 + 2c\langle\hat{p}_y, z_{ij}\rangle + c^2||\hat{p}_y||^2||z_{ij}||^2}.
\end{aligned} \tag{9}$$

with $\hat{p}_y = -p_y$ for clarity. The formulation above allows us to precompute α and β for reuse. Then, we rewrite the inner product with w_y as:

$$\begin{aligned}
\langle\hat{p}_y \oplus_c z_{ij}, w_y\rangle &= \langle\alpha\hat{p}_y + \beta z_{ij}, w_y\rangle, \\
&= \alpha\langle\hat{p}_y, w\rangle + \beta\langle z_{ij}, w\rangle.
\end{aligned} \tag{10}$$

Where an explicit computation of the Möbius addition requires evaluating a tensor in $\mathbb{R}^{W\times H\times C\times n}$ for a single image, this is reduced to adding two tensors in $\mathbb{R}^{W\times H\times C}$. The squared norm of the Möbius addition can be efficiently computed as follows:

$$\begin{aligned}
||\hat{p}_y \oplus_c z_{ij}||^2 &= \sum_{m=1}^{n}(\alpha\hat{p}_y^m + \beta z_{ij}^m)^2, \\
&= \sum_{m=1}^{n}(\alpha\hat{p}_y^m)^2 + \alpha\hat{p}_y^m\beta z_{ij}^m + (\beta z_{ij}^m)^2, \\
&= \alpha^2\sum_{m=1}^{n}(\hat{p}_y^m)^2 + 2\alpha\beta\sum_{m=1}^{n}\hat{p}_y^m z_{ij}^m + \beta^2\sum_{m=1}^{n}(z_{ij}^m)^2, \\
&= \alpha^2||\hat{p}_y||^2 + 2\alpha\beta\langle\hat{p}_y, z_{ij}\rangle + \beta^2||z_{ij}||^2,
\end{aligned} \tag{11}$$

which is a summation of three tensors in $\mathbb{R}^{W\times H\times C}$. Moreover, all terms have already been computed when precomputing α and β. By the reformulation of the inner product and squared norm when computing the class logits, we make hyperbolic classification feasible at the pixel level.

3.3. Hierarchical hyperbolic class embedding

It has been repeatedly shown that the hyperbolic space is able to embed hierarchical structures with minimal distortion [33, 36, 38]. To that end, we investigate the potential of incorporating hierarchical relations between classes for image segmentation on hyperbolic manifolds. Let Y denote the set of all classes, which form the leaf nodes of hierarchy $\mathcal{N}$. For class $y \in Y$, let $\mathcal{A}_y$ denote the ancestors of y.

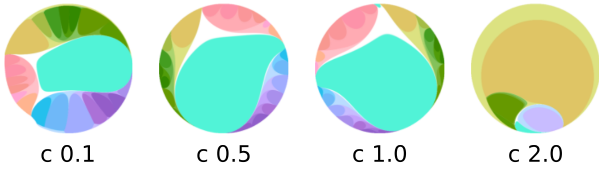


Figure 3. **Visualizing class embeddings** in hyperbolic space for the 20 classes of Pascal VOC. The colors outline the hierarchical structure of the classes. The higher the curvature, the more the gyroplanes are positioned towards the edge of the Poincaré disk. In the analyses, we investigate the quantitative effect of hyperbolic curvature for segmentation performance.

The probability of class y for output z_{ij} is then given by a hierarchical softmax:

$$\begin{aligned}
p(\hat{y} = y|z_{ij}) &= \prod_{h\in\mathcal{H}_y} p(h|\mathcal{A}_h, z_{ij}) \\
&= \prod_{h\in\mathcal{H}_y} \frac{\exp(\zeta_h(z_{ij}))}{\sum_{s\in S_h}\exp(\zeta_s(z_{ij}))},
\end{aligned} \tag{12}$$

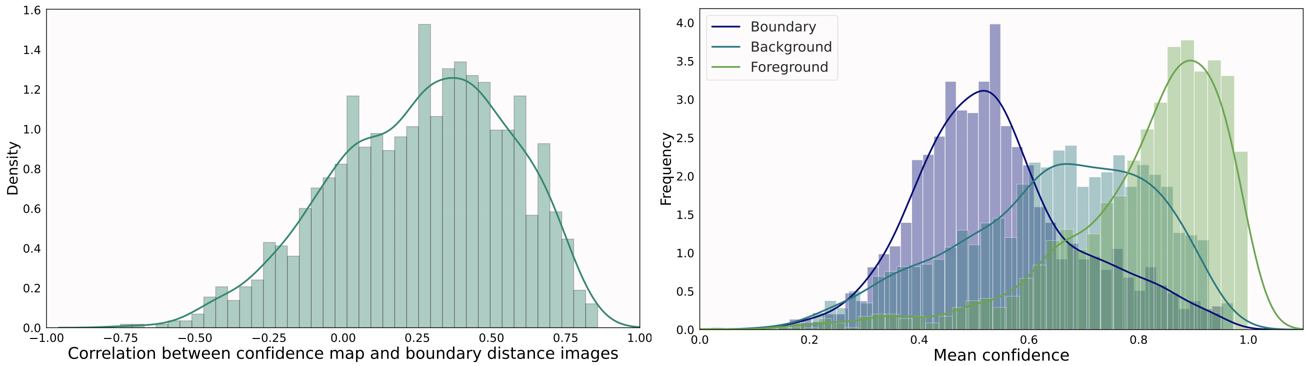
with $\mathcal{H}_y = \{y\} \cup \mathcal{A}_y$ and with S_h the siblings of h. The above formulation calculates the joint probability from root to leaf node, where the probability at each node is given as the softmax normalized by the siblings in the same subtree. Given this probability function, training can be performed with cross-entropy and the most likely class is selected during inference based on Equation 12. In Figure 3, we visualize how incorporating such knowledge results in a hierarchically consistent embedding of class gyroplanes.

4. Analyses

4.1. Setup

Datasets. We evaluate Hyperbolic Image Segmentation on three datasets, COCO-Stuff-10K [6], Pascal VOC [14], and ADE20K [54]. **COCO-Stuff-10K** contains 10,000 images from 171 classes consisting of 80 countable *thing* classes such as *umbrella* or *car*, and 91 uncountable *stuff* classes such as *sky* or *water*. The dataset is split into 9,000 images in the training set and 1,000 images in the test set. **Pascal VOC** contains 12,031 images from 21 classes consisting of 20 object classes like *person* and *sheep* and a *background* class. The dataset is split into 10,582 images in the train set and 1,449 images in the test set. **ADE20K** contains 22,210 images from 150 classes, such as *car* and *water*. The dataset is split into 20,210 in the train set and 2000 images in the test set. For all datasets, we have made the full hierarchies and they are shown in the supplementary materials.

Implementation details. For all experiments, we use DeeplabV3+ with a ResNet101 backbone [10]. We initialize the learning rate to be 0.001, 0.001, and 0.01 for COCO-stuff-10k, ADE20K, and Pascal VOC. We train the model



(a) Hyperbolic uncertainty correlates with boundary distance.

(b) Hyperbolic uncertainty is higher for boundaries than object interiors.

Figure 4. **Is hyperbolic uncertainty semantically meaningful?** We perform two quantitative experiments on Pascal VOC with 2 embedding dimensions to uncover whether hyperbolic uncertainty provides meaningful insights. Left: we find that the per-pixel hyperbolic uncertainty (here shown as its inverse, namely confidence) strongly correlates with semantic boundaries in the segmentation. Right: hyperbolic confidence is highest for foreground pixels denoting object interiors, followed by background pixels and finally semantic boundaries.

for 70, 140, and 40 epochs for COCO-stuff-10K, ADE20K, and Pascal VOC with a batch size of 5. To optimize Euclidean parameters, we use SGD with a momentum of 0.9 and polynomial learning rate decay with a power of 0.9 akin to [10]. To optimize Hyperbolic parameters, we use RSGD, similar to [18].

Evaluation metrics. We perform the evaluation on both standard and hierarchical metrics. For the standard metrics, we use pixel accuracy (PA), class accuracy (CA), and mean Intersection Over Union (mIOU). Pixel accuracy denotes the percentage of pixels in the image with the correct label. Class accuracy first calculates the accuracy per class and then averages over all classes. IOU denotes the spatial overlap of ground truth and predicted segmentation. mIOU denotes the mean IOU over all classes. To evaluate hierarchical consistency and robustness, we also report sibling and cousin variants of each metric, following [25]. In the sibling variant of the metrics, a prediction is also counted as correct if it shares a parent with the target class. In the cousin variants, the predicted labels need to share a grandparent with the target class to count as correct.

4.2. Uncertainty and boundary information for free

The ability to interpret predictions is vital in many segmentation scenarios, from medical imaging to autonomous driving, to invoke trust and enable decision making with a human in the loop [1]. For the first analysis, we investigate the role of hyperbolic embeddings for interpretation in segmentation. Specifically, we show how the distance to the origin of each pixel in the hyperbolic embedding space provides a natural measure of uncertainty prediction. We draw comparisons to Bayesian uncertainty and investigate whether hyperbolic uncertainty is semantically meaningful.

Hyperbolic vs Bayesian uncertainty. To obtain per-pixel uncertainty in Hyperbolic Image Segmentation, we simply measure the ℓ_2 norm to the origin in the Poincaré ball, regardless of their positioning to the class-specific gyroplanes. In conventional segmentation architectures, such uncertainty measures are more commonly obtained through Bayesian optimization, either by making the network Bayesian from the start [43] or through Monte-Carlo dropout during inference [28].

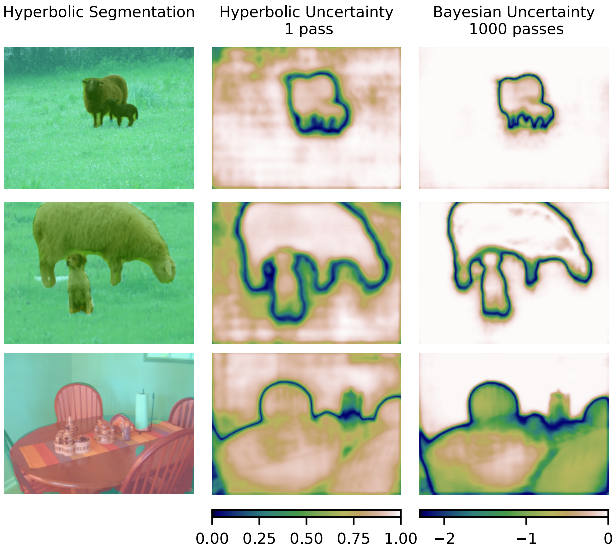


Figure 5. **Hyperbolic vs Euclidean uncertainty** for examples from Pascal VOC. Both measures of uncertainty are highly aligned and focus on semantic boundaries. However, the Bayesian uncertainty for Euclidean embeddings require 1,000 passes, whereas we obtain uncertainty for free with hyperbolic embeddings.

In Figure 5, we show the uncertainty maps for examples from Pascal VOC for hyperbolic uncertainty with 2 embedding dimensions and curvature 0.1. We draw a qualitative comparison to its Bayesian counterpart in Euclidean space by way of dropout during inference [16]. Both variants employ the same backbone. To create the Bayesian uncertainty map, we add Mont-Carlo dropout after Resnet blocks with a drop ratio of 0.5 and pass each image 1,000 times through the network, similar to [28]. Figure 5 shows three hyperbolic and Bayesian uncertainty example maps. Both uncertainty maps are highly interpretable, focusing on semantic boundaries and occluded areas of the image. A key difference however is the amount of network passes required to obtain the maps: while Bayesian uncertainty requires many passes due to the MC dropout, we obtain the uncertainty maps for free, resulting in a 1,000-fold inference speed-up.

Is hyperbolic uncertainty semantically meaningful? The qualitative results suggest that the hyperbolic uncertainty measure is semantically meaningful, as it relates to the semantic boundaries between objects. To test this hypothesis, we have outlined a quantitative experiment: for each pixel in the ground truth segmentation map, we compute the Euclidean distance to the nearest pixel with another class label. Intuitively, this distance correlates with prediction confidence; the closer to the boundary, the smaller the hyperbolic norm. We perform a correlation analysis between confidence and boundary distance for all pixels in an image. We then aggregate the correlations over all images.

In Figure 4a, we show a histogram of the correlations over all images in Pascal VOC with the same embedding dimensionality and curvature as above. The histogram shows that the confidence (inverse of uncertainty) from our hyperbolic approach clearly correlates with the distance to the nearest boundary. This result highlights that hyperbolic uncertainty provides a direct clue about which regions in the image contain boundaries between images, which can, in turn, be used to determine whether to ignore such regions or to pinpoint where to optimize further as boundary areas commonly contain many errors [37]. We provide the same experiment for 256 embedding dimensions in the supplementary materials, which follows the same distribution.

To further highlight the relation between hyperbolic uncertainty and semantic boundaries, we have performed a second quantitative experiment, where we classify each pixel into one of three classes: boundary pixel if it is within 10 distances from the nearest other class, background pixel, or foreground pixel (*i.e.* one of the other objects). In Figure 4b, we plot the mean confidence per pixel on Pascal VOC over all three classes, showing that hyperbolic confidence is highest for foreground pixels and lowest for boundary pixels, with background pixels in between. All information about boundaries and pixel classes comes for free with hyperboles as the embedding space in segmentation.

		COCO-Stuff-10k		
Manifold	Hierarchical	Class Acc	Pixel Acc	mIOU
$\mathbb{R}$		0.44	0.33	0.23
$\mathbb{R}$	✓	3.29	48.65	18.53
$\mathbb{D}$	✓	**3.46**	**51.70**	**21.15**

		Pascal VOC		
Manifold	Hierarchical	Class Acc	Pixel Acc	mIOU
$\mathbb{R}$		4.88	10.84	2.59
$\mathbb{R}$	✓	7.80	31.04	16.15
$\mathbb{D}$	✓	**12.15**	**47.92**	**34.87**

Table 1. **Zero-label generalization** on Coco-Stuff-10k and Pascal VOC. On both datasets, combining hierarchical knowledge with hyperbolic embeddings provides a more suitable foundation for generalizing to unseen classes than its Euclidean counterpart.

4.3. Zero-label generalization

In the second analysis, we demonstrate the potential of hyperbolic embeddings to generalize to unseen classes for image segmentation. We perform zero-label experiments on COCO-Stuff-10k and Pascal VOC and follow the zero-label semantic segmentation setup from Xian *et al.* [45]. For COCO-Stuff-10k we use a set of 15 unseen classes for inference, corresponding to all classes in the dataset that do not occur in the 2014 ImageNet Large Scale Visual Recognition Challenge [35], on which the backbone was pre-trained. This assures that the model has never seen any of the classes during training. For Pascal VOC, we follow the 15/5 seen/unseen split of [45]. We draw a comparison to two baselines: the standard DeepLabV3+, which operates in Euclidean space and does not employ hierarchical relations, and a variant of DeepLabV3+ that employs a Euclidean hierarchical softmax.

More formally, given a set of unseen classes C_U and a set of seen classes C_S, we remove all $k \in C_U$ from the dataset by replacing them with an ignore label. This effectively means that these pixels are not used during optimization and the model is therefore not optimized on these classes. As such, in images containing concepts from C_U, the pixels containing the concepts from C_S are still used in training. Different from the more widely known zero-shot image classification task, images containing unseen concepts are not removed from the training set. Removing these images would result in a significantly reduced training set, which is impractical for the purposes of the evaluation. After training on C_S, we perform inference by choosing only between unseen concepts for each pixel. We note that we do not adapt our approach to the zero-label setting, we employ the same network and loss as for supervised segmentation, the only difference lies in the used classes for training and inference.

The results on COCO-Stuff-10k and Pascal VOC are shown in Table 1 for 256 output dimensions and respective curvatures 1 and 2. In the supplementary materials, we also show the results using the sibling and cousin variants of the three metrics. For both datasets, we first observe that using a standard Euclidean architecture without hierarchical knowledge results in near-random zero-label performance. When using hierarchical knowledge and Euclidean embeddings, it

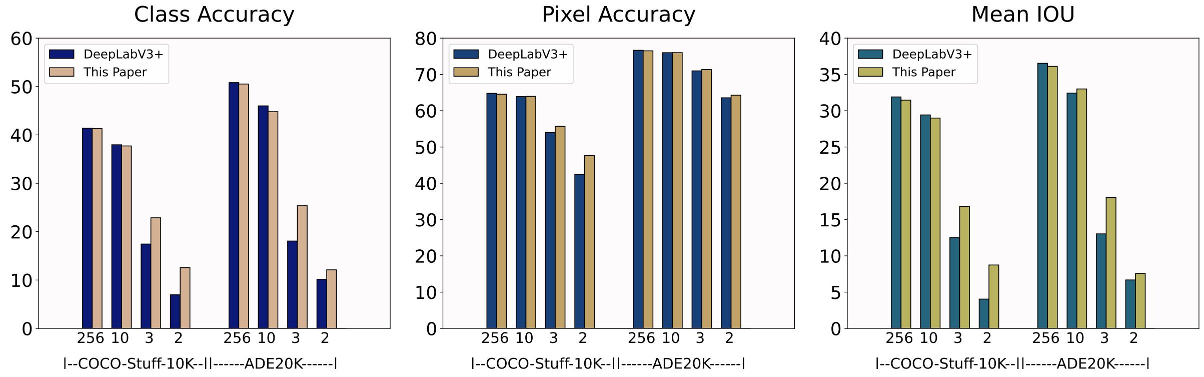


Figure 6. **Low-dimensional effectiveness of hyperbolic embeddings** for image segmentation on COCO-Stuff-10k and ADE20k. Across all three metrics, our approach obtains competitive performance in high-dimensional embedding spaces to the Euclidean counterpart. When restricting the embedding space to a few dimensions, hyperbolic embeddings are preferred for segmentation.

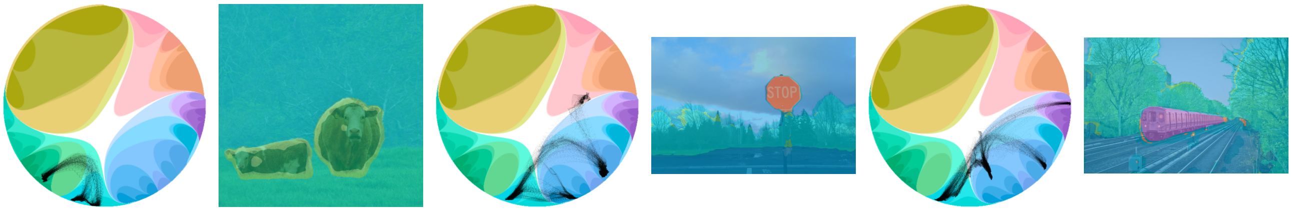


Figure 7. **Qualitative examples of Hyperbolic Image Segmentation** with two embedding dimensions on COCO-Stuff-10k. For each example, we show the projection of all pixels in the hyperbolic embedding (left) and the segmentation result (right). From left to right: the lime color denotes *cow* (partial failure case), the red color denotes *stop sign*, and the purple color denotes *train*.

becomes possible to recognize unseen classes. To generalize towards unseen classes, however, it is best to combine class hierarchies with hyperbolic embeddings. On COCO-Stuff-10k, the mIOU increases from 18.53 to 20.76. On Pascal VOC, the difference is even bigger; from 16.15 to 34.87. This experiment shows the strong affinity between hierarchical knowledge and hyperbolic embeddings for image segmentation and the potential for generalizing to unseen classes. We conclude that the hyperbolic space provides a more suitable foundation for generalizing to unseen classes in the context of segmentation. Qualitative zero-label results are provided in the supplementary materials.

4.4. Low-dimensional embedding effectiveness

In the third analysis, we demonstrate the effectiveness of hyperbolic embeddings in a low-dimensional setting. Hyperboles have shown to be beneficial with few embedding dimensions on various data types. In Figure 6, we compare the default Euclidean embeddings to hyperbolic embeddings for DeepLabV3+ on COCO-Stuff-10k and ADE20K, with a dimensionality ranging from 256 to 2. The standard setting of classical segmentation for DeepLabV3+ is to operate on a dimensionality of 256. Low dimensional embeddings are however preferred for explainability and on-device segmentation [3], due to their reduced complexity and smaller memory footprint.

Our results show a consistent pattern across both datasets and the metrics, where hyperbolic embeddings obtain comparable performances for high (256) or medium (10) dimension settings. In low-dimensional settings (2 and 3), our approach outperforms DeepLabV3+. As expected, the performance of both models drops when using lower dimensional embeddings, but as is especially apparent on the COCO-Stuff-10k dataset, the Euclidean default is affected most. By using a structured embedding space we are able to obtain better performance in low-dimensions, for as low as 2 dimensions. When using 3 dimensions, hyperbolic embeddings improve the mIOU by 4.32 percent point on COCO-Stuff-10k and by 4.99 on ADE20k. The benefits of this low dimensional embedding for explainability are demonstrated with the hyperbolic disk visualisations in this paper, which are based on models trained in 2 dimensions. We conclude that the low-dimensional effectiveness of hyperbolic embeddings extends to the task of image segmentation. In Figure 7 we provide qualitative examples in 2-dimensional hyperbolic embedding spaces. Further explanation on the colors is provided in the supplementary materials.

4.5. Further ablations

To complete the analyses, we ablate two design choices in our approach, namely the hyperbolic curvature and the use of hierarchical relations in the hyperbolic embedding space. Both ablations are performed on COCO-Stuff-10k.

Curvature. Since hyperbolic spaces are curved, there

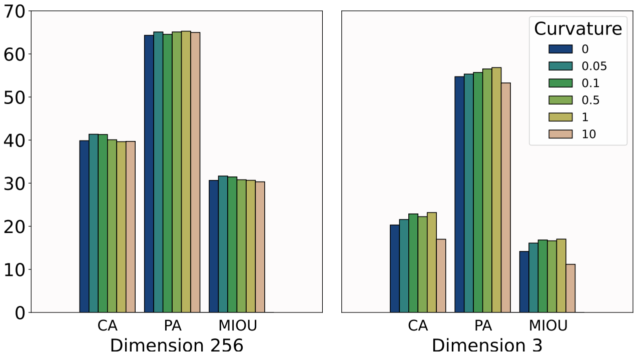


Figure 8. **Comparison of curvature** for high (256) and low (3) dimensional hyperbolic embeddings. Performance reported as Classification Accuracy (CA), Pixel Accuracy (PA), and Mean IOU (MIOU). For high dimensions the model is robust to changes in the curvature value, in a low dimensional setting similar robustness can be observed for low curvature values, however the performance drops when using a high curvature value (10).

is an additional hyperparameter compared to the Euclidean space (*i.e.* $c = 0$) that governs the curvature and radius of the Poincaré ball. In Figure 8 we show the effect of different curvatures for image segmentation on both 256- and 3-dimensional embeddings. For 256-dimensional embeddings, we can observe that the effect of the curvature value is negligible, with only minor changes in performance even for large curvature differences (e.g., 0.05 to 10). A similar observation can be made with 3 dimensions, except that for this lower dimensionality we see a drop in performance when the curvature is set to 10. We suspect that, because the embedding space shrinks with increasing curvature, a low dimensionality combined with a high curvature reduces the size of the embedding space too far. In practice, we use validation to determine the curvature in a range of 0.1 to 2.

Hierarchical versus flat hyperbolic softmax. Throughout the analyses, we have combined hyperbolic embeddings for image segmentation with hierarchical relations amongst the target classes, due to the well-established match between hierarchies and hyperbolic space. In this ablation study, we show the effect of incorporating such hierarchical knowledge in the context of segmentation. We draw a comparison to the conventional flat setting with one-hot encodings over all classes (*i.e.* omitting hierarchies). The results shown in Table 2 clearly highlight the benefits of hierarchical softmax, outperforming the flat softmax in almost all cases - on both the hierarchical and the standard metrics. Increasing the dimensionality reduces the difference between the hierarchical and flat softmax, with the flat softmax even slightly outperforming the hierarchical softmax on the standard metric in 256 dimensions. Nevertheless, across all dimensionalities the hierarchical softmax is preferred for the hierarchical metrics, demonstrating the benefit of incorporating hierarchical knowledge for segmentation.

		Mean IOU		
Dimension	Softmax	$\sim$	S	C
2	Flat	4.31	12.47	19.48
	Hierarchical	8.74	22.67	33.05
3	Flat	11.11	26.19	34.41
	Hierarchical	16.82	34.85	45.89
10	Flat	28.89	46.85	55.85
	Hierarchical	28.99	47.35	56.74
256	Flat	31.77	48.59	57.27
	Hierarchical	31.46	48.73	58.34

Table 2. **Effect of embedding hierarchical knowledge** in Hyperbolic Image Segmentation on COCO-Stuff-10k. In few dimensions, employing a hierarchical softmax is preferred over a flat softmax based on one-hot vectors. As dimensionality increases, this preference diminishes for standard metrics, while hierarchical softmax remains preferred for the hierarchical metrics.

5. Conclusions

This work investigates semantic image segmentation from a hyperbolic perspective. Hyperbolic embeddings have recently shown to be effective for various machine learning tasks and data types, from trees and graphs to images and videos. Current hyperbolic approaches do however not scale to the pixel level, since the corresponding operations are memory-wise intractable. We introduce Hyperbolic Image Segmentation, the first approach for image segmentation in hyperbolic embedding spaces. We outline an equivalent and tractable formulation of hyperbolic multinomial logistic regression to enable this step. Through several analyses, we demonstrate that operating in hyperbolic embedding spaces brings new possibilities to image segmentation, including uncertainty and boundary information for free, improved zero-label generalization, and better performance in low-dimensional embedding spaces.

Limitations and negative impact. Throughout the experiments, we have used DeepLabv3+ as backbone due to the well-known and performant nature of the architecture. Our analyses do not yet uncover the effect of hyperbolic embeddings in more shallow or deeper architectures, or their effect beyond natural images such as the medical domain. While we do not focus on specific applications, segmentation in general does have potentially negative societal applications that the reader needs to be aware of, such as segmentation in surveillance and military settings.

Acknowledgments. Erman Acar is generously funded by the Hybrid Intelligence Project which is financed by the Dutch Ministry of Education, Culture and Science with project number 024.004.022.

References

[1] Zeynep Akata, Dan Balliet, Maarten De Rijke, Frank Dignum, Virginia Dignum, Guszti Eiben, Antske Fokkens, Davide Grossi, Koen Hindriks, and Holger Hoos. A research agenda for hybrid intelligence: Augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. *Computer*, 53(8):18–28, 2020. 5

[2] Rami Aly, Shantanu Acharya, Alexander Ossa, Arne Köhn, Chris Biemann, and Alexander Panchenko. Every child should have parents: a taxonomy refinement algorithm based on hyperbolic term embeddings. *ACL*, 2019. 2

[3] Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic busemann learning with ideal prototypes. *NeurIPS*, 2021. 2, 7

[4] Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. Constant curvature graph convolutional networks. In *ICML*, 2021. 2

[5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 2017. 2

[6] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, 2018. 4

[7] James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 1997. 3

[8] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *NeurIPS*, 2019. 1, 2

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017. 2

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2, 4, 5

[11] Hyunghoon Cho, Benjamin DeMeo, Jian Peng, and Bonnie Berger. Large-margin classification in hyperbolic space. In *AISTATS*, 2019. 2

[12] Jindou Dai, Yuwei Wu, Zhi Gao, and Yunde Jia. A hyperbolic-to-hyperbolic graph convolutional network. In *CVPR*, 2021. 1, 2

[13] Ankit Dhall, Anastasia Makarova, Octavian Ganea, Dario Pavllo, Michael Greeff, and Andreas Krause. Hierarchical image classification using entailment cone embeddings. In *CVPRw*, 2020. 2

[14] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 4

[15] Pengfei Fang, Mehrtash Harandi, and Lars Petersson. Kernel methods in hyperbolic spaces. In *ICCV*, 2021. 1

[16] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016. 5

[17] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *ICML*, 2018. 2

[18] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *NeurIPS*, 2018. 2, 3, 5

[19] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *CVPR*, 2020. 1, 2

[20] Marc Law, Renjie Liao, Jake Snell, and Richard Zemel. Lorentzian distance learning for hyperbolic representations. In *ICML*, 2019. 2

[21] Xiaodan Liang, Hongfei Zhou, and Eric P. Xing. Dynamic-structured semantic propagation network. In *CVPR*, 2018. 2

[22] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *NeurIPS*, 2019. 1, 2

[23] Shaoteng Liu, Jingjing Chen, Liangming Pan, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang. Hyperbolic visual embedding learning for zero-shot recognition. In *CVPR*, 2020. 1, 2

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[25] Teng Long, Pascal Mettes, Heng Tao Shen, and Cees GM Snoek. Searching for actions on the hyperbole. In *CVPR*, 2020. 1, 2, 5

[26] Aaron Lou, Isay Katsman, Qingxuan Jiang, Serge Belongie, Ser-Nam Lim, and Christopher De Sa. Differentiating through the fréchet mean. In *ICML*, 2020. 2

[27] George A. Miller. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, 38:39–41, 1995. 2

[28] Jishnu Mukhoti and Yarin Gal. Evaluating bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv:1811.12709*, 2018. 5, 6

[29] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NeurIPS*, 2017. 1, 2

[30] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *ICML*, 2018. 2

[31] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *CVPR*, 2015. 2

[32] Jiwoong Park, Junho Cho, Hyung Jin Chang, and Jin Young Choi. Unsupervised hyperbolic representation learning via message passing auto-encoders. In *CVPR*, 2021. 1

[33] Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *arXiv preprint arXiv:2101.04562*, 2021. 4

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 6

[36] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *ICML*, 2018. 2, 4

[37] Laurens Samson, Nanne van Noord, Olaf Booij, Michael Hofmann, Efstratios Gavves, and Mohsen Ghafoorian. I bet you are wrong: Gambling adversarial networks for structured semantic segmentation. In *ICCVw*, 2019. 6

[38] Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pages 355–366. Springer, 2011. 2, 4

[39] Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. *ICLR*, 2021. 2

[40] Dídac Surís, Ruoshi Liu, and Carl Vondrick. Learning the predictability of the future. In *CVPR*, 2021. 1

[41] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *CoRR*, 2020. 2

[42] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincaré glove: Hyperbolic word embeddings. In *ICLR*, 2019. 1, 2

[43] Dustin Tran, Michael W. Dusenberry, Mark van der Wilk, and Danijar Hafner. Bayesian layers: A module for neural network uncertainty. In *NeurIPS*, 2019. 5

[44] Zhenzhen Weng, Mehmet Giray Ogut, Shai Limonchik, and Serena Yeung. Unsupervised discovery of the long-tail in instance segmentation using hierarchical self-supervision. *CoRR*, 2021. 2

[45] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *CVPR*, 2019. 6

[46] Jiexi Yan, Lei Luo, Cheng Deng, and Heng Huang. Unsupervised hyperbolic metric learning. In *CVPR*, 2021. 1

[47] Tao Yu and Chris De Sa. Numerically accurate hyperbolic embeddings using tiling-based models. In *NeurIPS*, 2019. 2

[48] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020. 2

[49] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018. 2

[50] Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. Lorentzian graph convolutional networks. In *WWW*, 2021. 2

[51] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open vocabulary scene parsing. In *ICCV*, Oct 2017. 2

[52] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2

[53] Zilong Zhong, Zhong Qiu Lin, Rene Bidart, Xiaodan Hu, Ibrahim Ben Daya, Zhifeng Li, Wei-Shi Zheng, Jonathan Li, and Alexander Wong. Squeeze-and-attention networks for semantic segmentation. In *CVPR*, 2020. 2

[54] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 4

[55] Yudong Zhu, Di Zhou, Jinghui Xiao, Xin Jiang, Xiao Chen, and Qun Liu. Hypertext: Endowing fasttext with hyperbolic geometry. In *EMNLP*, 2020. 1, 2