Hyperbolic Part-Whole Image Segmentation

Mikhail Vlasenko
University of Amsterdam

Mina Ghadimi Atigh
University of Amsterdam

Pascal Mettes
University of Amsterdam

Abstract

Semantic segmentation typically focuses on pixel-level classification at the object level. Yet, objects naturally decompose into parts and subparts, mirroring human visual perception. In this work, we introduce a hyperbolic prototypical segmentation framework capable of simultaneously representing multiple granularity levels within a unified embedding space. Leveraging hyperbolic geometry's unique capacity to model hierarchies effectively, we propose to embed class prototypes within the Poincaré ball. We introduce a tree-aware prototype initialization strategy and a distortion-p loss that together yield improved hierarchical embeddings. Furthermore, we derive an optimized formulation of the hyperbolic distance function, enabling tractable inference for dense prediction tasks. A shared transformer encoder paired with separate hyperbolic heads allows efficient multi-level segmentation from a single model. Experiments on the recently introduced SubPartImageNet show that our approach (i) improves over the state-of-the-art, especially at the *subpart* and *part* levels, at a fraction of the number of parameters, (ii) enables zero-shot generalization, and (iii) allows for transfer from part- to object-level predictions without object-level supervision. All code is available at https://github.com/mikhail-vlasenko/hyp-segmentation.

1 INTRODUCTION

Image segmentation constitutes a central task in computer vision, with the goal to assign a semantic label to each pixel in an image (J. Long et al., 2015). Current literature in segmentation, from classical fully convolutional approaches (J. Long et al., 2015; Ronneberger et al., 2015; Badrinarayanan et al., 2017; L.-C. Chen et al., 2017, 2018) to modern attention-based frameworks (Xie et al., 2021; Liu et al., 2021; W. Wang et al., 2021), are commonly trained to perform semantic prediction at one level of granularity, namely the object-level. However, each pixel not only represents an object, but also a part or even a subpart. Consider, for example, an image of a *biped*, which has a *head* (part) and an *eye* (subpart), as shown in Figure 1. The human visual perception is well-known to organize scenes as part-whole organizations (Biederman, 1987).

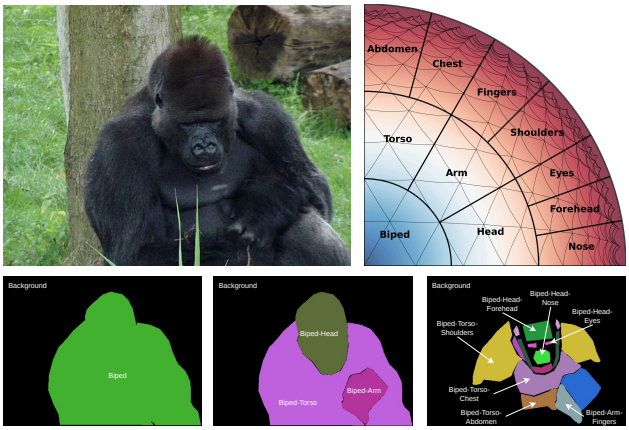


Figure 1: **The Hierarchy** of part-whole relations.

Image segmentation with part-whole hierarchies is, however, difficult to attain in practice, as current benchmarks largely focus on object-level understanding only. The recently introduced SubPartImageNet (SPIN) dataset makes it possible to reason at multiple levels of the visual hierarchy for common ImageNet classes (Myers-Dean et al., 2024). The dataset contains dense multi-level annotations (*object*, *part*, *subpart*), enabling rigorous study of the visual hierarchy for image segmentation. The state-of-the-art treats each level of granularity as independent, while

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

the multi-level annotations intuitively provide rich relations for predictive performance and generalization.

The goal of this work is to perform part-whole image segmentation using the inherent hierarchical organization from subparts to objects. We take inspiration from recent literature on hyperbolic learning, which was shown to be superior over Euclidean space for embedding hierarchies (Krioukov et al., 2010; Nickel & Kiela, 2017; Mettes et al., 2024), with rich applications in visual understanding (T. Long et al., 2020; Khrulkov et al., 2020). Specifically, several works have shown that hyperbolic embeddings benefit image segmentation (Atigh et al., 2022; Weber et al., 2024), improving segmentation accuracy and zero-shot generalization when dealing with semantic hierarchies. Part-whole segmentation introduces a complementary type of hierarchy: the visual hierarchy, which groups visually similar parts across different objects. Our framework supports both hierarchical views, and we find that the choice of hierarchy has a significant impact on generalization. This setting brings unique challenges and leads to the following contributions:

1. We introduce a multi-head hyperbolic prototype network, enabling part-whole image segmentation where each granularity can make use of a different preferred hierarchical organization.
2. We outline an improved construction of hyperbolic prototypes through tree-aware initialization and distortion-p fine-tuning.
3. We reformulate distance computations in hyperbolic space to scale prototype-based classification to the pixel-level.
4. Our approach obtains state-of-the-art mIoU performance on SPIN on *part* and *subpart* levels at a fraction of the parameters, while also enabling zero-shot and cross-granular generalization.

2 RELATED WORK

2.1 Image Segmentation

Semantic image segmentation constitutes a long-standing problem in computer vision, with deep learning solutions progressing from FCNs (J. Long et al., 2015) to encoder-decoder models like U-Net and SegNet (Ronneberger et al., 2015; Badrinarayanan et al., 2017) and dilated-convolution variants (L.-C. Chen et al., 2017, 2018) that expand receptive fields while preserving resolution. Building on these advances, transformer-based architectures improved efficiency and accuracy. For example, SegFormer (Xie et al., 2021) uses a hierarchical Mix Transformer encoder that produces multi-scale features without fixed positional encodings, and a lightweight all-MLP decoder that fuses them effectively. This design achieves strong performance with comparatively few parameters and provides an efficient encoder for our setting. Recent foundation models such as SAM and SAM2 (Kirillov et al., 2023; Ravi et al., 2024) further excel at mask generation, with prompt-based adaptations extending their applicability to diverse settings (Zhao & Shen, 2024; Rafaeli et al., 2024). The focus in current literature is to reason at the object-level for pixel classification. This work aims to classify each pixel not only at the object-level, but also at the part- and subpart-levels.

2.2 Part-Based Image Segmentation

While general segmentation focuses on entire objects, part-based segmentation aims for a more granular decomposition of wholes into components. This increased granularity is crucial for applications requiring detailed object understanding, such as fine-grained recognition, robotic manipulation, and surgical instrument segmentation (Wah et al., 2011; Krause et al., 2013; Levine et al., 2016; Florence et al., 2018; Yue et al., 2023). Progress was enabled by datasets such as PASCAL-Part and Part-ImageNet (X. Chen et al., 2014; He et al., 2022), which offer segmentation masks at two granularity levels. De Geus et al. (2021) take it further and construct part-level instance-aware annotations for two popular segmentation datasets. While these provide a great step towards hierarchical segmentation, the two semantic levels do not present a deep hierarchy. Recently, the SubPartImageNet (SPIN) dataset (Myers-Dean et al., 2024) uniquely introduced dense annotations at three granularities on natural images, providing masks for *objects*, *parts*, and *subparts*. This has enabled benchmarking of powerful vision-language models on multi-level segmentation, and has inspired concurrent works on part-level generation (Zhou et al., 2025) and consistent interactive segmentation (Myers-Dean, Liu, et al., 2025). HALLUMI (Myers-Dean, Price, et al., 2025) introduces an autoregressive language modeling approach that encodes parent-child relationships through special tokens, performing hierarchical segmentation in a single inference pass. While HALLUMI relies on an LLM to implicitly capture class relations, we take a complementary approach and embed the hierarchy directly into the projection space via prototypical learning in hyperbolic space.

2.3 Hyperbolic Deep Learning

To enable hyperbolic part-whole image segmentation, we take inspiration from recent advances in hyperbolic learning. Representing hierarchical relations in continuous spaces naturally motivates non-Euclidean geom-

etry. Hyperbolic spaces with negative curvature embed tree-like structures with low distortion (Krioukov et al., 2010; Nickel & Kiela, 2017; Sala et al., 2018; Mettes et al., 2024), with the Poincaré ball as a widely used model (van Spengler, Berkhout, & Mettes, 2023; Chami et al., 2019; Mathieu et al., 2019). The pioneering work of Nickel and Kiela (2017) demonstrated the effectiveness of learning graph node embeddings in the Poincaré ball, showing significant improvements over Euclidean methods for representing complex networks with latent hierarchies. Since then, hyperbolic methods have been successfully applied to natural language processing for learning word hierarchies from text corpora (Tifrea et al., 2018), modeling logical entailment (Ganea et al., 2018b), and generating sentence representations (Le et al., 2019). In other areas, they have proven effective for recommender systems (Vinh Tran et al., 2020) and video action recognition (T. Long et al., 2020). Several works have even shown how image segmentation can be done in hyperbolic space (Atigh et al., 2022; Weber et al., 2024). These works, however, focus on semantic hierarchies: the taxonomic relationship between objects. In this work, we additionally explore visual hierarchies, which group visually similar parts across different objects and bring new challenges. To that end, we introduce a multi-head hyperbolic segmentation framework, a new hierarchy embedding loss, and show how to scale prototype-based hyperbolic learning (Mettes et al., 2019; Kasarla et al., 2022; Ghadimi Atigh et al., 2021; Pal et al., 2024) to the pixel-level.

3 METHOD

In image segmentation, we are given an image $X \in \mathbb{R}^{w \times h \times 3}$, with w and h the width and height in number of pixels. For each pixel $x \in X$, our goal is to assign three labels $(y_o, y_p, y_s) \in Y_o \times Y_p \times Y_s$, denoting the object, part, and subpart levels. Let $\phi(X) : \mathbb{R}^{w \times h \times 3} \mapsto \mathbb{R}^{w \times h \times d}$ denote the backbone, which obtains a d-dimensional feature vector per pixel. We map each pixel $x = \phi(X)_{ij}$ to hyperbolic space through an exponential mapping:

$$z = \tanh\left(\sqrt{-c}\|\mathbf{v}\|\right) \frac{\mathbf{v}}{\sqrt{-c}\|\mathbf{v}\|}. \quad (1)$$

with c indicating the curvature of the Poincaré ball. We strive to align each pixel representation with the part-whole hierarchy. To make this possible, we first show how to embed the part-whole hierarchy in hyperbolic space. We then show how to scale hyperbolic prototype-based learning from class-level to pixel-level. Finally, we outline our overall framework for hyperbolic part-whole image segmentation.

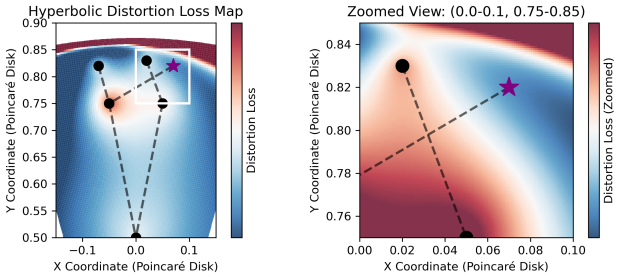


Figure 2: **The Issue of Uniform Initialization.** We Show the Distortion Loss Landscape for the Purple-Star Embedding. A Narrow High-Loss Ridge Separates the Current Position from a Lower-Loss Basin, Hindering Subtree Swaps, Resulting in Local Minima.

3.1 Embedding Part-Whole Hierarchies

Let V denote the union of all object, part, and subpart labels, with E denoting the hierarchical relations between the three. Our first goal is to embed directed acyclic graph $\mathcal{G} = (V, E)$ in hyperbolic space. We do so by constructing a learnable embedding for each node $v \in V$ in the d-dimensional Poincaré ball $\mathbb{B}^d$. The procedure of the embedding consists of three stages: (i) tree-aware initialization of the embeddings, (ii) pretraining with a contrastive objective, (iii) tuning with a higher-order distortion loss.

Tree-Aware Hierarchy Initialization. The convention in hyperbolic tree embedding methods is to initialize nodes as embeddings with a uniform distribution $\mathcal{U}(-0.001, 0.001)$ (Nickel & Kiela, 2017; Ganea et al., 2018a). However, we find that this approach is suboptimal, particularly in lower-dimensional spaces. In Figure 2, we visualize the issue by showing the distortion loss landscape for moving the embedding marked by the purple star. Placing the starred embedding at $(-0.03, 0.84)$ on the Poincaré disk would achieve a smaller loss, however, a transition there requires passing through a ridge around $(0.03, 0.84)$, which is an implausible trajectory for gradient descent. While increasing the embedding dimension partially mitigates this issue as the ridge becomes less extreme, it does not fundamentally address the problem of poor initial topology.

As a solution, we initialize by depth-first traversal of the hierarchy from the root, placing nodes in a circular arrangement in the Poincaré ball, similar in spirit to van Spengler and Mettes (2025). In higher dimensions, the first two coordinates follow this circle and the remaining dimensions use the uniform initialization.

Contrastive pre-training. Second, we employ the stochastic approach of Nickel and Kiela (2017) and iteratively optimize the embeddings with Riemannian SGD (Bécigneul & Ganea, 2018) on batches of edges:

$$\mathcal{L}_{\text{PE}} = -\frac{1}{|B|} \sum_{(u,v)\in B} \log \frac{\exp(-d_c(\mathbf{y}_u, \mathbf{y}_v))}{\sum_{v'} \exp(-d_c(\mathbf{y}_u, \mathbf{y}_{v'}))}, \quad (2)$$

where u, v are indices of source and target nodes for an edge; $\mathbf{y}_u, \mathbf{y}_v$ are Poincaré ball embeddings of nodes u and v; B is a batch of positive edges augmented with sampled negative examples; and d_c denotes the hyperbolic distance:

$$d_c(x, y) = \frac{2}{\sqrt{-c}} \tanh^{-1}\left(\sqrt{-c}\| - x \oplus_c y\|\right). \quad (3)$$

Distortion-p loss. Lastly, we fine-tune the embeddings using a distortion loss that aligns hyperbolic distances with target distances d_{uv}^{target} derived from shortest paths in the hierarchy graph $\mathcal{G}$. Prior works have shown that directly optimizing for distortion is effective (Sala et al., 2018), but only optimize for the absolute value of the distortion error. Instead, we propose the following distortion loss:

$$\mathcal{L}_{\text{dist}} = \frac{1}{|B|} \sum_{(u,v)\in B} \left| \frac{d_{\mathbb{B}}(\mathbf{z}_u, \mathbf{z}_v) - d_{uv}^{\text{target}}}{d_{uv}^{\text{target}}} \right|^p, \quad (4)$$

where $\mathbf{z}_u$ and $\mathbf{z}_v$ are the hyperbolic embeddings of nodes u and v in the Poincaré ball, and p is the proposed tunable power parameter controlling the emphasis on relative error. An absolute-valued term, i.e. with $p = 1$, leads to uniform penalization of relative distortion. However, in large hierarchies most node pairs are distant, biasing optimization toward global structure. Setting $p > 1$ emphasizes large relative errors and rebalances the objective toward local accuracy, improving segmentation performance.

3.2 Tractable Prototypical Learning

With the part-whole hierarchy embedded in hyperbolic space and each pixel also projected to the same d-dimensional embedding space, we align each pixel to the embedded hierarchy. To make this possible, we view the embedding of each label as a prototype, which serves as a target to optimize the segmentation network. However, applying this concept to a dense prediction task like semantic segmentation introduces a significant computational challenge posed by the standard formulation of the hyperbolic distance in the Poincaré ball model, as shown in Equation 3. In segmentation, this expression must be evaluated for every pixel representation against a set of class prototypes in hyperbolic space. This entails computing distances between $128 \times 128 \times$ `batch_size` pixel vectors

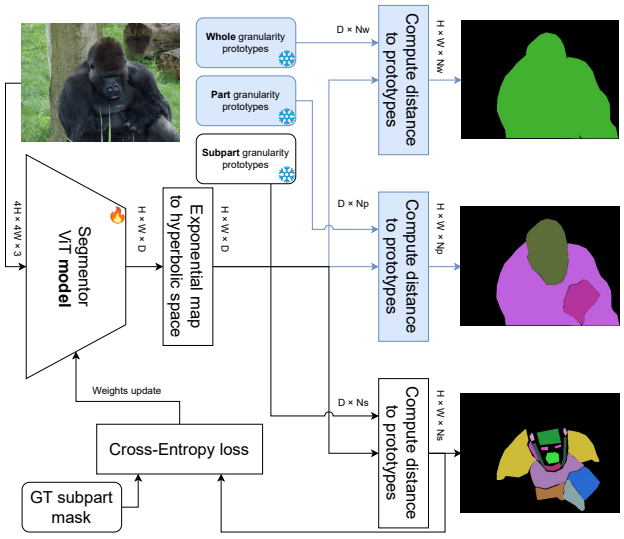


Figure 3: **Hyperbolic Prototypical Multi-Level Prediction.** The Framework Predicts All Three Levels Despite Supervision on Only One. Components Used Only at Inference are Shown in Blue. D is Prototype Dimensionality; N_w, N_p, N_s are Class Counts for *Whole*, *Part*, *Subpart*.

and over 200 class prototypes, resulting in prohibitive memory and compute demands using standard implementations.

Reformulation of Hyperbolic Distances. We propose to reformulate hyperbolic prototype-based segmentation inspired by Atigh et al. (2022). An important consideration is that the hyperbolic distance function only requires the *norm* of the Möbius difference $-x \oplus_c y$. Instead of explicitly computing the Möbius addition and only retaining the norm, we want to bypass needless intermediate computations.

To compactly express repeated terms, we define:

$$\alpha = 1 - 2c\langle x, y\rangle - c\|y\|^2 \quad \in \mathbb{R} \quad (5)$$
$$\beta = (1 + c\|x\|^2) \quad \in \mathbb{R} \quad (6)$$
$$\gamma = 1 - 2c\langle x, y\rangle + c^2\|x\|^2\|y\|^2 \quad \in \mathbb{R} \quad (7)$$

Then:

$$\|-x \oplus_c y\| = \sqrt{\sum_i \left(\frac{\alpha x_i + \beta y_i}{\gamma}\right)^2} = \frac{\sqrt{\sum_i (\alpha x_i + \beta y_i)^2}}{\gamma}$$

Expanding the sum, we can compute the resulting

Table 1: Our Compact Hyperbolic Shared Encoder Architecture Achieves a New SOTA on the SPIN Dataset, Surpassing the Multi-Billion-Parameter GLaMM-FT on the Most Challenging *part* and *subpart* Levels. Results Show Mean ± Standard Deviation Over 4 Seeds. † Reported by Myers-Dean et al. (2024). ‡ Reported by Myers-Dean, Price, et al. (2025).

Model	object mIoU	part mIoU	subpart mIoU	# parameters
HIPIE (X. Wang et al., 2023)†	0.2169	0.0823	0.0092	$800M$
PixelLLM (Xu et al., 2024)†	0.7996	0.3296	0.1013	$13B$
LISA (Lai et al., 2024)†	0.8545	0.3136	0.1155	$13B$
GLaMM (Rasheed et al., 2024)†	0.8631	0.4	0.11	$7B$
GLaMM-FT (Rasheed et al., 2024)†	**0.9108**	0.6076	0.2456	$3 \times 7B$
HALLUMI (Myers-Dean, Price, et al., 2025)‡	0.893	0.5815	0.1845	$7B$
This paper	0.895 ± 0.001	$\mathbf{0.677} \pm 0.005$	$\mathbf{0.2676} \pm 0.003$	$\mathbf{64M + 3 \times 3M}$

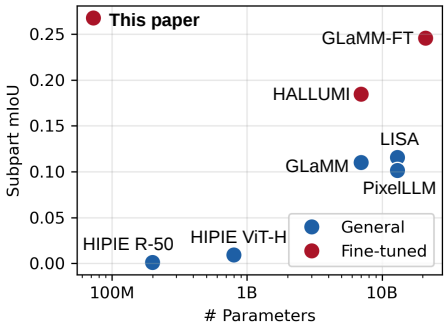


Figure 4: **Subpart mIoU vs. # Parameters.** Our method achieves the best subpart segmentation at a fraction of the model size.

scalar directly:

$$\sum_i (\alpha x_i + \beta y_i)^2 = \sum_i \left(\alpha^2 x_i^2 + 2\alpha\beta x_i y_i + \beta^2 y_i^2\right) = \alpha^2\|x\|^2 + 2\alpha\beta\langle x, y\rangle + \beta^2\|y\|^2 \quad (8)$$

This representation requires computing only 3 vector operations: $\|x\|^2, \|y\|^2, \langle x, y\rangle$. When x and y are large matrices, this approach avoids instantiating any high-dimensional intermediate tensors in hyperbolic distance calculation (Equation 3). Concretely, a naïve implementation requires a difference tensor of shape $(B, H, W, N_{\text{proto}}, D)$, resulting in ∼128 GB of memory for a single forward and backward pass of the distance function alone. Our reformulation reduces this to ∼2.5 GB, making hyperbolic prototypical learning feasible for dense, high-resolution tasks like semantic segmentation.

Treating Distances as Logits. We project decoder features for each pixel into the Poincaré ball via the exponential map from Equation 10. We then cast distances to prototypes as logits in a standard cross-entropy: for pixel (i, j) and class l,

$$\text{logits}_{ijl} = -\tau \cdot d_c(r_{ij}, p_l), \quad (9)$$

where d_c is the hyperbolic distance (Equation 3) with reformulated norm (Equation 8) under curvature c, r_{ij} the pixel embedding, p_l the class prototype, and τ a temperature controlling sharpness.

After training, this decouples encoding from classification: the network learns embeddings, while classification is a fixed comparison to precomputed prototypes. This separation provides remarkable flexibility, as the prototypes can be modified, swapped, or extended post-training without the need for retraining any part of the model.

3.3 Overall Framework

A key advantage of our prototypical framework is the explicit encoding of the class hierarchy within the geometry of the embedding space. This enables capabilities beyond standard segmentation, such as producing segmentation masks for higher hierarchical levels, entirely without the corresponding labels.

Single head. In the constrained setting, we construct the hierarchical prototypes for all classes (Section 3.1) and train the model using only leaf (*subpart*) masks. At inference, the same pixel embeddings are compared to prototypes at any granularity (*subpart*, *part*, *whole*) to produce all levels in a single forward pass, as shown in Figure 3. Because all prototypes lie in the same hyperbolic space, coarser levels require no additional training or parameters, yielding consistent, hierarchy-aware outputs and a strong zero-shot prior.

Shared Encoder with Multi-Level Supervision. To allow for greater flexibility, and leverage all available ground-truth data, our second approach allows for level-specific prototype arrangements. It employs

a shared backbone encoder with separate prediction heads for each hierarchical level. While the encoder learns a common feature representation, each head maintains its own set of hyperbolic prototypes, permitting each granularity to adopt its own optimal structure. During training, we calculate a loss for each head against its corresponding ground-truth masks. These individual losses are then summed to create an aggregated signal for the backward pass. By combining supervision from different granularities, the shared encoder learns a more robust and comprehensive representation that is informed by the full label hierarchy.

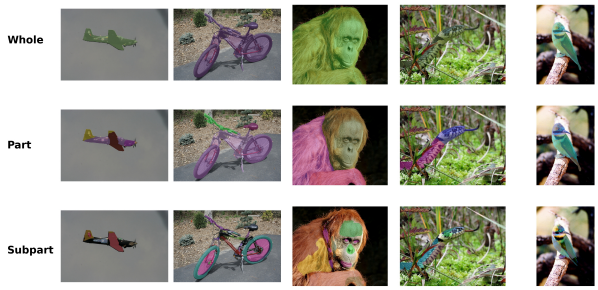


Figure 5: **Qualitative Results** From Our Hyperbolic Shared Encoder Model on the SPIN Test Set. Predicted Masks are Overlaid on the Inputs.

4 EXPERIMENTS

4.1 Setup

Dataset. Our experiments primarily focus on the SPIN (Myers-Dean et al., 2024) dataset, taking advantage of its vast 3-level class hierarchy and object-centric nature. SPIN annotates a subset of images from ImageNet (Deng et al., 2009), and spans 11/40/203 objects/parts/subparts with over 102k masks. It is split into 8,828 training images, 519 validation images, and 1,040 test images. For zero-shot evaluations, we additionally utilize PASCAL-Part (X. Chen et al., 2014) to validate the robustness and generalizability of our proposed method.

Implementation details. The experimental framework is based on SegFormer (Xie et al., 2021) and uses Geoopt (Kochurov et al., 2020) and HypLL (van Spengler, Wirth, & Mettes, 2023) libraries to handle most of the hyperbolic geometry operations. All experiments used global seeds 0, 1, 2, and 42, contributing to reproducibility and enabling comprehensive analysis of model performance and hyperparameter impact. The training was performed on one NVIDIA A100 Tensor Core GPU.

Evaluation metrics. To measure the segmentation performance, we adopt the mean Intersection over Union (mIoU) — a standard metric for semantic segmentation (Everingham et al., 2010; J. Long et al., 2015). The mIoU is calculated by averaging the per-class IoU, which is the ratio of the intersection to the union area between the predicted and ground truth masks of a given class.

4.2 Comparative analysis

In the first experiment, we compare our multi-level supervision framework to the state-of-the-art on the SPIN dataset. Table 1 summarizes the performance of our hyperbolic model in comparison with state-of-the-art foundation models. LISA and GLaMM are evaluated in a zero-shot manner, while GLaMM-FT is fine-tuned on the SPIN training set. HALLUMI (Myers-Dean, Price, et al., 2025) is the only other method that explicitly models the label hierarchy, doing so through autoregressive language modeling. Unlike these baselines, which treat each granularity level independently, our approach explicitly models the relational structure between levels in hyperbolic space — an advantage that not only improves segmentation accuracy but also enables the zero-shot generalization explored in Section 4.5. We find that our approach is competitive in object segmentation and obtains the best performance in part and subpart segmentation, outperforming both hierarchy-agnostic and hierarchy-aware baselines. Specifically, at the subpart level, we achieve an mIoU of 0.2676, representing a 9% improvement over GLaMM-FT's performance of 0.2456 and a 45% improvement over HALLUMI. We note that our results come at a fraction of the computational cost: GLaMM-FT has approximately 100 times more parameters and requires significantly more forward passes to create a complete mask for one input image. These results show that a hyperbolic prototype-based method is a natural fit for part-whole image segmentation.

Beyond mIoU, we also report the Spatial Consistency (SC) metric introduced by Myers-Dean et al. (2024). Our method achieves an SC of 0.9504 ± 0.0011 for subpart-to-part and 0.9600 ± 0.0016 for part-to-whole consistency, outperforming GLaMM-FT on subpart-to-part (0.8516) while remaining competitive on part-to-whole (0.9604). Qualitative examples of our model's multi-level output are shown in Figure 5.

4.3 Cross-Level Generalization

To evaluate the cross-level capabilities of the framework, we conduct an experiment where models with a

Table 2: Hyperbolic Geometry Enables Robust Multi-Level Generalization from Only *Subpart* Supervision.

Method	object mIoU	part mIoU	subpart mIoU
Baseline	0.0672	0.1208	0.2100
This paper	**0.3874**	**0.3607**	**0.2300**

Table 3: Segmentation Performance for Different Hierarchies and Prototype Dimensions. Part-first is Advantageous for Low Dimensionality.

Manifold	Hierarchy	Dim	Part mIoU
Euclidean	-	4	0.2517 ± 0.0581
Hyperbolic	Standard	4	0.6261 ± 0.0055
Hyperbolic	Part-first	4	$\mathbf{0.6439} \pm 0.0042$
Euclidean	-	64	0.6778 ± 0.0006
Hyperbolic	Standard	64	$\mathbf{0.6793} \pm 0.0050$
Hyperbolic	Part-first	64	0.6772 ± 0.0134

single prediction head are trained only on the *subpart* annotations and tested on their ability to predict *part* and *whole* masks, without any direct supervision on these coarser levels. In Table 2, we compare our hyperbolic method with its Euclidean counterpart, differing only in the manifold chosen for the prototype embeddings.

The hyperbolic model significantly outperforms the Euclidean one in transferring hierarchical knowledge upward, demonstrating a strong inductive bias towards hierarchy-aware representation learning. This crucially enables robust multi-level prediction from limited granularity annotations.

Finally, we find that thanks to the advantages of the hyperbolic space, it is possible to embed the prototypes in $d << \text{num\_classes}$ (such as 8 dimensions for 256 classes in this case), at negligible cost to segmentation performance. The downstream effect of the embedding dimension of the class prototypes is explored further in Tables 3 and 6.

4.4 Part-First Hierarchy

We additionally propose a second class hierarchy for SPIN: one where similar parts across different wholes are generalized to whole-agnostic pseudo classes, and the whole-specific part classes are their direct children. Subtrees of both hierarchies are presented in Figure 6 for visual comparison. Such hierarchy encourages the model to learn a more abstract, transferable, concept of a *part* (e.g. 'Torso') that is not strictly tied to being part of a specific *whole* (e.g. 'Bird' or 'Reptile'). Since the *part*-to-*subpart* relation in this hierarchy remains

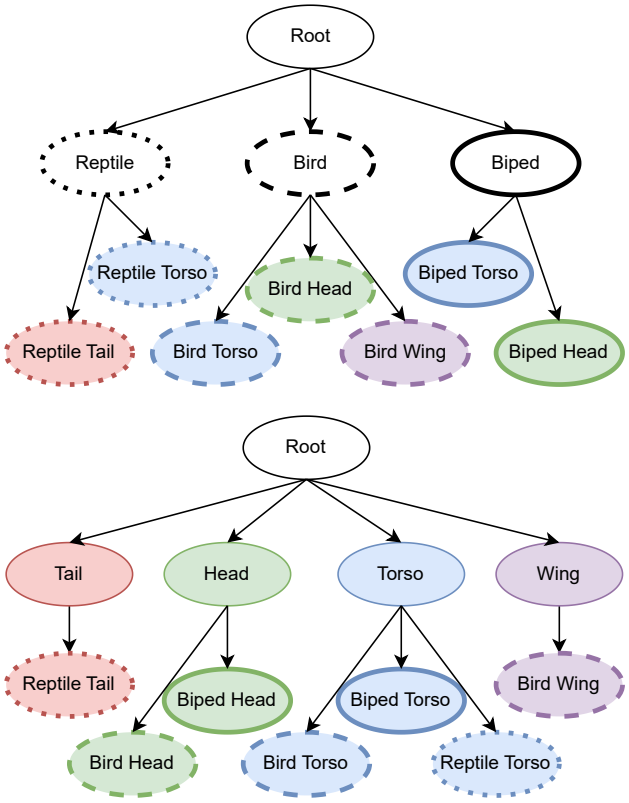


Figure 6: **Subtrees of Class Hierarchies.** Top: Standard Hierarchy. Bottom: Part-First Hierarchy. The Part-First Hierarchy Groups Classes of the *part* Granularity by Their Kind Instead of the *whole* That They Belong To.

unchanged, we benchmark this new hierarchy on the *part* level, rather than the *subpart* level.

As presented in Table 3, we find that the Part-first hierarchy is superior to our Standard hierarchy for classification at part granularity in low dimensions. In high dimensions, the difference is statistically insignificant. This is expected, as such setup positions 41 *part*-level class embeddings in a 64-dimensional space, removing any structural prior.

4.5 Zero-Shot Evaluation

We evaluate zero-shot by holding out one top-level *whole* (e.g., Biped). During training, we exclude all images containing that *whole*. At evaluation, we test only on images of the held-out *whole* and restrict predictions to its parts or the background. This zero-shot setup is possible because SPIN is object-centric: each image contains exactly one *whole* category, allowing us to cleanly remove that concept from training while still evaluating on it.

Table 4: Zero-shot *Part* mIoU Averaged Over Held-out *Wholes*. Our Hyperbolic Method and Part-first Hierarchy Vastly Outperform Alternatives.

Manifold	Hierarchy	Dim	Average part mIoU
Euclidean	-	4	0.0265 ± 0.0018
Euclidean	-	16	0.0304 ± 0.0019
Hyperbolic	Standard	4	0.1509 ± 0.0022
Hyperbolic	Part-first	4	0.3596 ± 0.0037
Hyperbolic	Part-first	8	0.3438 ± 0.0129
Hyperbolic	Part-first	32	$\mathbf{0.3985} \pm 0.0237$

Table 5: Zero-shot Evaluation on PASCAL-Part.

Manifold	Hierarchy	Dim	Car part mIoU
Euclidean	-	4	0.0070 ± 0.0064
Hyperbolic	Standard	4	0.0709 ± 0.0002
Hyperbolic	Standard	32	0.0710 ± 0.0003
Hyperbolic	Part-first	4	0.0725 ± 0.0010
Hyperbolic	Part-first	32	$\mathbf{0.0733} \pm 0.0028$

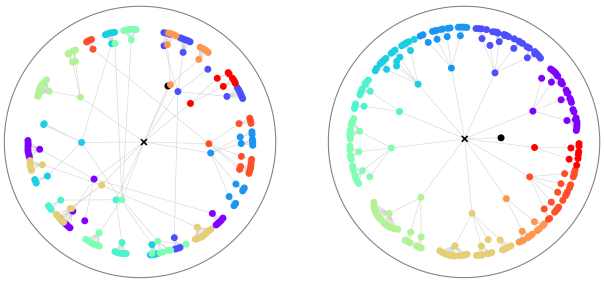
As presented in Table 4, the standard Euclidean model, lacking any explicit hierarchical structure, obtains near-zero mIoU. Hyperbolic prototypes substantially improve performance, and the Part-first hierarchy yields the strongest zero-shot transfer. Moreover, we observe that increasing the prototype embedding dimension further boosts the segmentation performance. Notably, the trends and values of mIoU are highly consistent between the tested held-out *wholes*.

The model's zero-shot performance converges quickly, achieving its best validation mIoU within the first 3 training epochs. This indicates that the model benefits only from the general alignment of its embedding space, rather than learning the fine-grained details of segmentation mask boundaries.

To guard against overfitting to SPIN and to check that trends replicate on a different dataset, we apply the same protocol to PASCAL-Part (X. Chen et al., 2014), holding out the *Car whole*. As shown in Table 5, hyperbolic again outperforms Euclidean, and Part-first hierarchy is also advantageous over the Standard one, though the difference is smaller than on SPIN. This demonstrates that our strong zero-shot capability is a property of the framework rather than an artifact of a single dataset.

4.6 Tree-Initialized Hyperbolic Prototypes

In this section, we compare tree-aware initialization against the standard random baseline to assess its effect on the learned geometry and downstream segmentation accuracy.



(a) Optimized From Random Initialization (b) Optimized From Tree Initialization (Ours)

Figure 7: **Tree-Aware Initialization** Yields a Much More Coherent Hierarchy.

The visualizations in Figure 7 provide a stark contrast between the two approaches. The optimizer struggles to correct the initial poor topology of the random initialization, resulting in a final embedding with higher distortion and less meaningful geometric relationships between classes. In contrast, tree-aware initialization leads to a clear, interpretable structure with roots near the center and children radiating outward.

This improved geometric arrangement directly benefits segmentation. As shown in Table 6, tree-aware initialization consistently outperforms random, with the gains most pronounced in lower dimensions that are sensitive to poor topology.

4.7 Loss Power

We now ablate the proposed distortion-p loss. Because large hierarchies contain predominantly distant pairs, the standard $p = 1$ biases optimization towards global structure at the expense of local separation. For a classification task, however, accurately representing the small distances between semantically similar classes is paramount, as these are the most likely to be confused.

Increasing p emphasizes large relative errors, improving local accuracy. As shown in Table 6, $p = 3$ yields the best mIoU, reduces distortion among nearby pairs, and decreases sibling confusion.

5 CONCLUSION

This work challenges the conventional paradigm in semantic segmentation, which typically treats classes as an independent and unstructured set, by developing a framework that embeds class hierarchies directly

Table 6: Prototype Construction Methods Significantly Improve Segmentation Performance. Tree-aware Initialization and Distortion-p Loss are Both Beneficial.

Initialization	p	Dim	Subpart mIoU
Random	1	64	0.1544 ± 0.0007
Random	3	4	0.1628 ± 0.0010
Random	3	8	0.1992 ± 0.0028
Random	3	64	0.2166 ± 0.0022
Random	5	64	0.2163 ± 0.0023
Tree-aware	1	64	0.1676 ± 0.0051
Tree-aware	3	4	0.0973 ± 0.0012
Tree-aware	3	8	$\mathbf{0.2300} \pm 0.0036$
Tree-aware	3	64	0.1843 ± 0.0037

into the geometry of a hyperbolic space. We demonstrate that explicitly modeling hierarchical relationships not only improves segmentation performance, but also opens the doors to multi-level and zero-shot generalization.

6 LIMITATIONS

While encoding the class hierarchy generally improves performance, we note two limitations. First, in some supervised settings, prototypes positioned in the maximal class separation configuration slightly outperform those strictly adhering to the hierarchy. Second, dataset annotation gaps introduce a systematic evaluation bias. Subpart masks do not tile the parent *part* region: some pixels that belong to a *part* have no *subpart* label and are treated as background during *subpart*-level training. When deriving *part*-level predictions from subpart-only supervision, the model classifies these pixels as background, reducing apparent *part*-level performance. This mismatch between label granularity and mask coverage accounts for some of the gap to directly supervised *part*-level models.

To address the dataset's annotation gaps, we propose training with joint supervision, leveraging *part*-level labels for pixels that lack a specific *subpart* annotation. The strong generalization results also suggest that this framework is an excellent candidate for few-shot or weakly-supervised learning, such as predicting fine-grained subparts from only coarse *part*-level labels. Exploring methods to learn class hierarchies directly from data or allowing prototypes to adapt dynamically during training could further enhance the model's flexibility and performance, moving us closer to a new generation of segmentation models that build a truly structured understanding of the visual world.

References

Atigh, M. G., Schoep, J., Acar, E., Van Noord, N., & Mettes, P. (2022). Hyperbolic image segmentation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4443–4452. https://doi.org/10.1109/CVPR52688.2022.00441 (cit. on pp. 2–4).

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481–2495 (cit. on pp. 1, 2).

Bécigneul, G., & Ganea, O.-E. (2018). Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760* (cit. on p. 4).

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological review*, *94*(2), 115 (cit. on p. 1).

Cannon, J. W., Floyd, W. J., Kenyon, R., Parry, W. R., et al. (1997). Hyperbolic geometry. *Flavors of geometry*, *31*(59-115), 2 (cit. on p. 13).

Chami, I., Wolf, A., Juan, D.-C., Sala, F., Ravi, S., & Ré, C. (2020). Low-dimensional hyperbolic knowledge graph embeddings. *arXiv preprint arXiv:2005.00545* (cit. on p. 14).

Chami, I., Ying, Z., Ré, C., & Leskovec, J. (2019). Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, *32* (cit. on p. 3).

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), 834–848 (cit. on pp. 1, 2).

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European conference on computer vision (ECCV)*, 801–818 (cit. on pp. 1, 2).

Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., & Yuille, A. (2014). Detect what you can: Detecting and representing objects using holistic models and body parts. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1971–1978 (cit. on pp. 2, 6, 8).

De Geus, D., Meletis, P., Lu, C., Wen, X., & Dubbelman, G. (2021). Part-aware panoptic segmentation. *Proceedings of the IEEE/CVF Confer-*

ence on Computer Vision and Pattern Recognition, 5485–5494 (cit. on p. 2).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248–255 (cit. on p. 6).

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, *88*, 303–338 (cit. on p. 6).

Florence, P. R., Manuelli, L., & Tedrake, R. (2018). Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756* (cit. on p. 2).

Ganea, O.-E., Bécigneul, G., & Hofmann, T. (2018a). Hyperbolic entailment cones for learning hierarchical embeddings. https://arxiv.org/abs/1804.01882 (cit. on p. 3).

Ganea, O.-E., Bécigneul, G., & Hofmann, T. (2018b, June 28). Hyperbolic neural networks. https://doi.org/10.48550/arXiv.1805.09112 (cit. on pp. 3, 13, 14).

Ghadimi Atigh, M., Keller-Ressel, M., & Mettes, P. (2021). Hyperbolic busemann learning with ideal prototypes. *Advances in neural information processing systems*, *34*, 103–115 (cit. on p. 3).

He, J., Yang, S., Yang, S., Kortylewski, A., Yuan, X., Chen, J.-N., Liu, S., Yang, C., Yu, Q., & Yuille, A. (2022, December 16). PartImageNet: A large, high-quality dataset of parts. https://doi.org/10.48550/arXiv.2112.00933 (cit. on p. 2).

Kasarla, T., Burghouts, G. J., & van Spengler, M. (2022). Maximum class separation as inductive bias in one matrix. *Advances in neural information processing systems*, *35*, 19553–19566 (cit. on p. 3).

Khrulkov, V., Mirvakhabova, L., Ustinova, E., Oseledets, I., & Lempitsky, V. (2020). Hyperbolic image embeddings. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6418–6428 (cit. on p. 2).

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. (2023). Segment anything. *Proceedings of the IEEE/CVF international conference on computer vision*, 4015–4026 (cit. on p. 2).

Kochurov, M., Karimov, R., & Kozlukov, S. (2020). Geoopt: Riemannian optimization in pytorch. *arXiv preprint arXiv:2005.02819* (cit. on p. 6).

Krause, J., Stark, M., Deng, J., & Fei-Fei, L. (2013). 3d object representations for fine-grained categorization. *Proceedings of the IEEE international conference on computer vision workshops*, 554–561 (cit. on p. 2).

Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., & Boguná, M. (2010). Hyperbolic geometry of complex networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, *82*(3), 036106 (cit. on pp. 2, 3).

Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., & Jia, J. (2024). Lisa: Reasoning segmentation via large language model. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9579–9589 (cit. on p. 5).

Le, M., Roller, S., Papaxanthos, L., Kiela, D., & Nickel, M. (2019). Inferring concept hierarchies from text corpora via hyperbolic embeddings. *arXiv preprint arXiv:1902.00913* (cit. on p. 3).

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, *17*(39), 1–40 (cit. on p. 2).

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022 (cit. on p. 1).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440 (cit. on pp. 1, 2, 6).

Long, T., Mettes, P., Shen, H. T., & Snoek, C. G. (2020). Searching for actions on the hyperbole. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1141–1150 (cit. on pp. 2, 3).

Mathieu, E., Le Lan, C., Maddison, C. J., Tomioka, R., & Teh, Y. W. (2019). Continuous hierarchical representations with poincaré variational auto-encoders. *Advances in neural information processing systems*, *32* (cit. on p. 3).

Mettes, P., Atigh, M. G., Keller-Ressel, M., Gu, J., & Yeung, S. (2024). Hyperbolic deep learning in computer vision: A survey. *International Journal of Computer Vision*, *132*, 3484–3508. https://doi.org/10.1007/s11263-024-02043-5 (cit. on pp. 2, 3, 13).

Mettes, P., Van der Pol, E., & Snoek, C. (2019). Hyperspherical prototype networks. *Advances in neural information processing systems*, *32* (cit. on p. 3).

Myers-Dean, J., Liu, K., Price, B., Fan, Y., Kuen, J., & Gurari, D. (2025). Consammé: Achieving consistent segmentations with sam. *Proceedings of the Computer Vision and Pattern Recognition Conference*, 759–768 (cit. on p. 2).

Myers-Dean, J., Price, B., Fan, Y., & Gurari, D. (2025). Hierarchical semantic segmentation with autoregressive language modeling. *Proceedings of the Computer Vision and Pattern Recognition Conference*, 4120–4130 (cit. on pp. 2, 5, 6).

Myers-Dean, J., Reynolds, J., Price, B., Fan, Y., & Gurari, D. (2024, August 8). SPIN: Hierarchical segmentation with subpart granularity in natural images. https://doi.org/10.48550/arXiv.2407.09686 (cit. on pp. 1, 2, 5, 6).

Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, *30* (cit. on pp. 2–4, 13).

Pal, A., Spengler, M. v., Melendugno, G. M. D. d., Flaborea, A., Galasso, F., & Mettes, P. (2024, October 9). Compositional entailment learning for hyperbolic vision-language models. https://doi.org/10.48550/arXiv.2410.06912 (cit. on p. 3).

Rafaeli, O., Svoray, T., Blushtein-Livnon, R., & Nahlieli, A. (2024). Prompt-based segmentation at multiple resolutions and lighting conditions using segment anything model 2. *arXiv preprint arXiv:2408.06970* (cit. on p. 2).

Rasheed, H., Maaz, M., Shaji, S., Shaker, A., Khan, S., Cholakkal, H., Anwer, R. M., Xing, E., Yang, M.-H., & Khan, F. S. (2024). Glamm: Pixel grounding large multimodal model. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13009–13018 (cit. on p. 5).

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). Sam 2: Segment anything in images and videos. https://arxiv.org/abs/2408.00714 (cit. on p. 2).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241 (cit. on pp. 1, 2).

Sala, F., De Sa, C., Gu, A., & Ré, C. (2018). Representation tradeoffs for hyperbolic embeddings. *International conference on machine learning*, 4460–4469 (cit. on pp. 3, 4).

Tifrea, A., Bécigneul, G., & Ganea, O.-E. (2018). Poincar\'e glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546* (cit. on p. 3).

van Spengler, M., Berkhout, E., & Mettes, P. (2023). Poincaré resnet. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5419–5428 (cit. on p. 3).

van Spengler, M., & Mettes, P. (2025). Low-distortion and gpu-compatible tree embeddings in hyperbolic space. *arXiv preprint arXiv:2502.17130* (cit. on p. 3).

van Spengler, M., Wirth, P., & Mettes, P. (2023). Hypll: The hyperbolic learning library. *Proceedings of the 31st ACM International Conference on Multimedia*, 9676–9679 (cit. on p. 6).

Vinh Tran, L., Tay, Y., Zhang, S., Cong, G., & Li, X. (2020). Hyperml: A boosting metric learning approach in hyperbolic space for recommender systems. *Proceedings of the 13th international conference on web search and data mining*, 609–617 (cit. on p. 3).

Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset (cit. on p. 2).

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., & Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *Proceedings of the IEEE/CVF international conference on computer vision*, 568–578 (cit. on p. 1).

Wang, X., Li, S., Kallidromitis, K., Kato, Y., Kozuka, K., & Darrell, T. (2023, December 21). Hierarchical open-vocabulary universal image segmentation. https://doi.org/10.48550/arXiv.2307.00764 (cit. on p. 5).

Weber, S., Zöngür, B., Araslanov, N., & Cremers, D. (2024, April 15). Flattening the parent bias: Hierarchical semantic segmentation in the poincaré ball. https://doi.org/10.48550/arXiv.2404.03778 (cit. on pp. 2, 3).

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021, October 28). SegFormer: Simple and efficient design for semantic segmentation with transformers. https://doi.org/10.48550/arXiv.2105.15203 (cit. on pp. 1, 2, 6).

Xu, J., Zhou, X., Yan, S., Gu, X., Arnab, A., Sun, C., Wang, X., & Schmid, C. (2024). Pixel-aligned language model. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13030–13039 (cit. on p. 5).

Yue, W., Zhang, J., Hu, K., Wu, Q., Ge, Z., Xia, Y., Luo, J., & Wang, Z. (2023). Surgicalpart-sam: Part-to-whole collaborative prompting for surgical instrument segmentation. *arXiv preprint arXiv:2312.14481* (cit. on p. 2).

Zhao, C., & Shen, L. (2024). Part-aware prompted segment anything model for adaptive segmentation. *arXiv preprint arXiv:2403.05433* (cit. on p. 2).

Zhou, M., Myers-Dean, J., & Gurari, D. (2025). Partstickers: Generating parts of objects for rapid prototyping. *Proceedings of the Computer Vision and Pattern Recognition Conference*, 6291–6301 (cit. on p. 2).

Checklist

1. For all models and algorithms presented, check if you include:
   (a) **Yes** — We describe the mathematical setting and model in Method (Sections 3.1 and 3.3) and preliminaries (Sections A.2 and A.2.1).
   (b) **Not Applicable** — No formal complexity analysis is required for our empirical method; we discuss tractability via our distance reformulation (Equation 8).
   (c) **Yes** — The full code is linked in a public GitHub repository.

2. For any theoretical claim, check if you include:
   (a) **Not Applicable**
   (b) **Not Applicable**
   (c) **Not Applicable**

3. For all figures and tables that present empirical results, check if you include:
   (a) **Yes**
   (b) **Yes**
   (c) **Yes**
   (d) **Yes**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) **Yes** — We cite all datasets and methods used (e.g., SPIN, PASCAL-Part, SegFormer, libraries).
   (b) **Not Applicable**
   (c) **Not Applicable**
   (d) **Not Applicable**
   (e) **Not Applicable**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) **Not Applicable**
   (b) **Not Applicable**
   (c) **Not Applicable**

A PRELIMINARIES

This appendix introduces the core concepts of hyperbolic geometry and its application in machine learning, which are useful for understanding our approach.

A.1 Hyperbolic Spaces

Hyperbolic spaces are a type of non-Euclidean geometry characterized by a constant negative curvature. Unlike Euclidean space, which has zero curvature, the volume of a hyperbolic space grows exponentially with its radius (Cannon et al., 1997). This exponential growth property makes hyperbolic spaces exceptionally well-suited for embedding hierarchical or tree-like data structures with significantly lower distortion than in Euclidean spaces of the same dimension (Nickel & Kiela, 2017; Mettes et al., 2024).

In machine learning, one of the commonly used models for hyperbolic geometry is the Poincaré ball model. We define the d-dimensional Poincaré ball with curvature $c < 0$ as the open unit ball in $\mathbb{R}^d$:

$$\mathbb{B}^d = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| < 1\}.$$

The geometry within this ball is distorted such that the distance between points grows rapidly as they approach the boundary of the ball $\|\mathbf{x}\| = 1$. Consequently, the boundary is considered to be "at infinity". This structure provides a natural way to represent hierarchies: general, high-level concepts can be placed near the origin (the center of the ball), while more specific, low-level concepts (or leaf nodes) are positioned closer to the boundary.

A.2 Exponential Map

The exponential map is used to project vectors from a Euclidean space into the hyperbolic manifold. The exponential map at the origin, $\exp_0^c(\mathbf{v})$, maps a vector $\mathbf{v}$ from the tangent space at the origin (which is a Euclidean space) to a point on the manifold (Ganea et al., 2018b):

$$\exp_0^c(\mathbf{v}) = \tanh\big(\sqrt{-c}\|\mathbf{v}\|\big)\frac{\mathbf{v}}{\sqrt{-c}\|\mathbf{v}\|}. \tag{10}$$

This operation is crucial for integrating hyperbolic geometry with standard neural network architectures, as it provides a way to map network outputs onto the Poincaré ball.

A.2.1 Möbius Addition

Möbius addition is a fundamental operation that generalizes the standard vector addition to hyperbolic space, maintaining consistency with the hyperbolic geometry. The operation, denoted as $\oplus_c$ in a manifold with curvature c is defined as:

$$x \oplus_c y = \frac{(1 - 2c\langle x, y\rangle - c|y|^2)x + (1 + c|x|^2)y}{1 - 2c\langle x, y\rangle + c^2|x|^2|y|^2}. \tag{11}$$

This operation is central to the geometry of the Poincaré ball. Unlike Euclidean addition, it is not commutative ($\mathbf{x} \oplus_c \mathbf{y} \neq \mathbf{y} \oplus_c \mathbf{x}$) nor associative ($\mathbf{x} \oplus_c (\mathbf{y} \oplus_c \mathbf{z}) \neq (\mathbf{x} \oplus_c \mathbf{y}) \oplus_c \mathbf{z}$). Nevertheless, the origin $\mathbf{0}$ acts as the identity element ($\mathbf{x} \oplus_c \mathbf{0} = \mathbf{x}$), and every point $\mathbf{x}$ has a unique inverse $-\mathbf{x}$ such that $\mathbf{x} \oplus_c (-\mathbf{x}) = \mathbf{0}$. Crucially, Möbius addition ensures that the result of combining two vectors always remains within the ball, preserving the manifold's structure.[1]

A.2.2 Hyperbolic Distances

The distance between two points $\mathbf{x}, \mathbf{y} \in \mathbb{B}^d$ is not the standard Euclidean distance but is instead defined by the geodesic path connecting them on the curved manifold. In the Poincaré ball model, this distance can be computed using Möbius addition:

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

[1] Euclidean addition can produce vectors outside of the hyperbolic manifold. For example, adding the two vectors $\mathbf{x} = (0.99, 0)$ and $\mathbf{y} = (0, 0.99)$ with the standard operation yields $\mathbf{x} + \mathbf{y} = (0.99, 0.99)$, which lies outside the Poincaré ball since $\|\mathbf{x} + \mathbf{y}\| = \sqrt{0.99^2 + 0.99^2} \approx 1.4 > 1$.

Here, the term $-\mathbf{x} \oplus_c \mathbf{y}$ can be interpreted as the "Möbius difference" or a translation of $\mathbf{y}$ by $-\mathbf{x}$. The distance function scales the Euclidean norm of this resulting vector to account for the curvature of the space. As points move away from the origin and approach the boundary of the ball, their norms approach 1. The $\tanh^{-1}$ function maps these values towards infinity, capturing the exponential expansion of the space and the rapidly increasing distances between points near the boundary. This property is precisely what allows the Poincaré ball to accurately represent the large distances between far leaves in a tree.

A.3 Hyperbolic MLR

Multinomial Logistic Regression (MLR) is a standard method for multi-class classification. Its generalization to hyperbolic space, Hyperbolic MLR, provides a way to define decision boundaries on the hyperbolic manifold (Ganea et al., 2018b).

A hyperbolic hyperplane is the generalization of a Euclidean hyperplane. It can be defined as the set of points equidistant from two given points or, more formally, as the projection of a linear subspace from a tangent space back onto the manifold. Hyperbolic MLR then defines classification logits based on the signed hyperbolic distance of a point to these hyperplanes. This provides a decision boundary that respects the structure of the hyperbolic space.

However, despite its geometric elegance, the Hyperbolic MLR framework has significant practical drawbacks that limit its scalability and stability. The core issue stems from its reliance on two learnable parameters for each class. This approach is inherently parameter-inefficient, especially for tasks with many categories. More critically, the reliance on on-manifold learnable parameters is the source of significant training instability. During training, gradient-based updates can push these parameters toward the boundary of the Poincaré ball. As a parameter's norm approaches 1, the denominators in the distance and gradient computations approach zero. This leads to severe floating-point inaccuracies, numerical overflow, and exploding gradients (Chami et al., 2020).

B SENSITIVITY ANALYSIS

We analyze the sensitivity of our method to the curvature c, the temperature τ, and the choice of backbone.

Curvature. We swept the curvature parameter c across values in $[0.01, 2]$. We found that values between 0.01 and 1 do not significantly affect downstream segmentation performance. Values above 2 led to numerical instabilities during training, consistent with known challenges of high-curvature Poincaré ball models.

Temperature. Figure 8 shows the effect of the temperature τ on segmentation performance in the single-head (subpart-supervised) setting. At the subpart level, performance is stable across all tested values ($\tau \in \{5, 10, 15, 20, 30\}$). At coarser levels, however, lower temperatures are preferable: whole-level mIoU drops sharply from 0.500 at τ=5 to 0.173 at τ=30. We use τ=15 for all experiments.

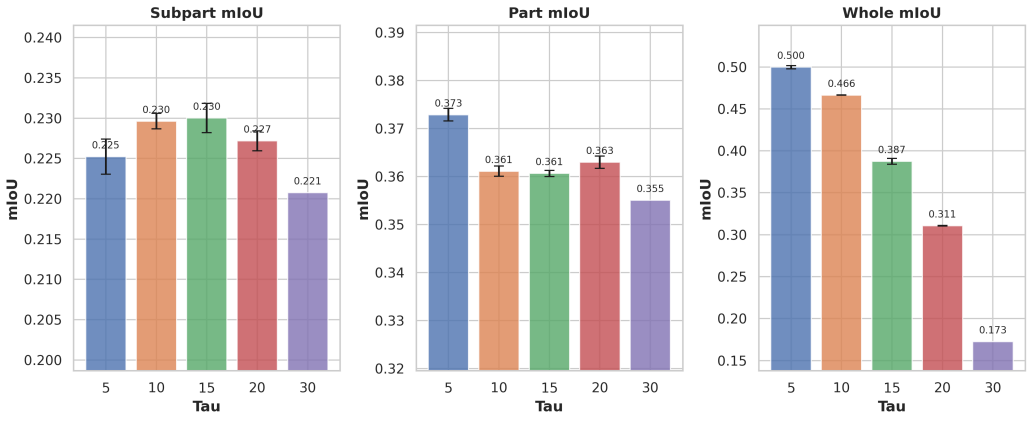


Figure 8: **Temperature Sensitivity** in the Single-Head Setting. Subpart mIoU is Stable Across τ Values, While Coarser Levels Benefit from Lower Temperatures.

Backbone. Our main experiments use the SegFormer-B4 backbone. To evaluate backbone dependence, we additionally trained a model with the smaller SegFormer-B0 backbone ($\sim$3.7M parameters). This model achieves 0.830 (object), 0.542 (part), and 0.182 (subpart) mIoU on the test set. Despite using only 6% of the parameters of our full model and 0.02% of the parameters of GLaMM-FT, it remains competitive, further highlighting the strength of our hyperbolic prototypical approach.