Capturing implicit hierarchical structure in 3D biomedical images with self-supervised hyperbolic representations

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider the task of representation learning for unsupervised segmentation of 3D voxel-grid biomedical images. We show that models that capture implicit hierarchical relationships between subvolumes are better suited for this task. To that end, we consider encoder-decoder architectures with a hyperbolic latent space, to explicitly capture hierarchical relationships present in subvolumes of the data. We propose utilizing a 3D hyperbolic variational autoencoder with a novel gyroplane convolutional layer to map from the embedding space back to 3D images. To capture these relationships, we introduce an essential self-supervised loss—in addition to the standard VAE loss—which infers approximate hierarchies and encourages implicitly related subvolumes to be mapped closer in the embedding space. We present experiments on synthetic datasets along with a dataset from the medical domain to validate our hypothesis.

1 Introduction

Advances in biomedical imaging techniques such as cryogenic electron tomography (cryo-ET) and magnetic resonance imaging (MRI) have resulted in an ever-increasing amount of 3D biomedical image data. In these data domains, a growing body of work shows that, when provided with labels, machine learning models achieve good performance on many tasks [Çiçek et al., 2016, Milletari et al., 2017, Dou et al., 2017, Falk et al., 2019]. However, these labels, especially for segmentation, are very costly as they often have to be provided by experts in the appropriate field. Consequently, supervised learning and even semi-supervised learning remain limited in this setting as (1) tasks and domains are very diverse, making it intractable for experts to provide labelled data for all problems; and (2) experts can only label objects they already know, restricting the potential for scientific discovery using machine learning methods. In this work, we tackle the task of unsupervised segmentation in 3D biomedical image data.

Our key insight is that 3D biomedical images have inherent hierarchical structure. For example, in the cryo-ET domain, an image of a cell has a hierarchy that at the highest level comprises the entire cell; at a finer level comprises organelles such as the mitochondria and nucleus; and at an even finer level comprises sub-structures such as the nucleolus of a nucleus or proteins within organelles. Such types of hierarchies are present in many types of biomedical images (e.g., nested anatomical structures within MRI images). We hypothesize that in the unsupervised setting, models that are able to encode this internal hierarchical structure will provide better data representations for downstream tasks. To that end, we propose learning representations based on embedding subvolumes of 3D images in hyperbolic space.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

In contrast to traditional Euclidean embeddings, hyperbolic embeddings better preserve hierarchical relationships present in the data. Hyperbolic representations have been proposed as a continuous way to represent hierarchical data, due to their ability to embed trees with arbitrarily low error [Sarkar, 2011]. A recent line of work utilizes hyperbolic representations to model hierarchical data across domains ranging from natural language word taxonomies [Nickel and Kiela, 2017, 2018] and graphs [Nickel and Kiela, 2017, Mathieu et al., 2019, Ovinnikov, 2019, Chami et al., 2019], to image classification [Mathieu et al., 2019]. In these settings, the objects, and in most cases their relationships, are explicitly encoded in the data. However, 3D biomedical images consist of subvolumes that represent parts of an *implicit* hierarchical structure. In our case, for any single 3D voxel-grid, we wish to embed and infer the implicit relationships between all of its subvolumes without any supervision.

To embed our 3D images in hyperbolic space, we use a 3D hyperbolic variational autoencoder (VAE). For the decoder, we propose a gyroplane convolutional layer which maps from the latent space back to 3D images while respecting hyperbolic geometry. In addition to the VAE loss, we propose an essential self-supervised loss to capture the hierarchical structure present in the data. More specifically, we consider reconstruction of implicit hierarchies as a pretext task. Concretely, we add a triplet loss which encourages a child subvolume to be mapped close to its parent subvolume in hyperbolic space. To capture hierarchical relationships of varying granularity, we train on subvolumes sampled at multiple scales. Finally, for a specified scale, we cluster the subvolumes in latent space and produce a segmentation map.

We evaluate our model on datasets with different domains: synthetic datasets and a medical dataset. We construct synthetic datasets where we generate structures at various scales and show that our model segments objects at multiple levels of hierarchy better than all prior unsupervised segmentation methods. We demonstrate performance gains ranging from 7% for the smallest objects to 32% for the largest objects. On the real-world medical dataset (BraTS Brain Tumor Segmentation Challenge) [Menze et al., 2014, Bakas et al., 2017, 2018], we show that our method outperforms prior works by 19%, and even achieves comparable performance to semi-supervised methods although we do not use any labels.

2 Related Work

Segmentation on 3D voxel data Many diverse biomedical images, ranging from MRI and CT scans to fluorescence microscopy, come in the form of 3D voxel-grids. Since 3D voxel-grids are dense, computer vision tasks such as supervised segmentation are commonly performed using deep learning architectures with 3D convolutional layers [Chen et al., 2016, Dou et al., 2017, Hesamian et al., 2019, Zheng et al., 2019]. However, due to the challenges of obtaining voxel-level annotations in 3D, there has been significant effort in finding semi-supervised approaches, including using labels only from several fully annotated 2D slices of an input volume [Çiçek et al., 2016], using a smaller set of segmentations with joint segmentation and registration [Xu and Niethammer, 2019], and using one segmented input in conjunction with other unlabelled data [Zhao et al., 2019].

Unsupervised approaches for 3D segmentation are useful not only for further reducing the manual annotation effort required, which is especially expensive for segmentation, but also for scientific discovery tasks where we would like to identify previously unknown structures for which annotations are impossible to produce. Moriya et al. [2018] extends to 3D data an iterative approach of feature learning followed by clustering [Yang et al., 2016]. Nalepa et al. [2020] uses a 3D convolutional autoencoder architecture and performs clustering of the latent representations. Another approach, [Dalca et al., 2018], uses a network pre-trained on manual segmentations from a separate dataset to perform unsupervised segmentation of 3D biomedical images. However, this limits applicability to areas where we already have a dataset with manual annotations and makes it unsuitable for unbiased scientific discovery. Gur et al. [2019] and Kitrungrotsakul et al. [2019] develop unsupervised methods for 3D segmentation of vessel structures, but these are specialized and do not generalize to the segmentation of other structures.

Another line of work performs unsupervised 2D segmentation, such as Ji et al. [2019] which proposes a mutual information objective for clustering, and Caron et al. [2018], which uses the clustered output of an encoder as pseudo-labels. While these methods can be applied to 2D slices of a 3D volume to perform 3D segmentation, they generally suffer limitations due to insufficient modeling of the 3D

spatial information. None of the aforementioned approaches explicitly learn hierarchical structure of the data, which is the main focus of our work.

Hyperbolic representations A recent line of work employs hyperbolic space to represent hierarchical structures, with the intuition that tree structures can be naturally embedded into continuous hyperbolic space [Nickel and Kiela, 2017]. These works utilize hyperbolic representations for a variety of tasks, including MNIST classification [Mathieu et al., 2019, Nagano et al., 2019, Ovinnikov, 2019], natural language processsing tasks such as embedding word taxonomies and entailment [Nickel and Kiela, 2017, Ganea et al., 2018], link prediction and node classification [Chami et al., 2019], and game playing [Nagano et al., 2019]. In most of these works, hierarchical structure is *explicitly* encoded in data. In contrast, we seek to capture *implicit* hierarchical structure arising from composition within 3D images.

Several architectures have been proposed in order to learn hyperbolic representations, including hyperbolic VAEs [Mathieu et al., 2019, Nagano et al., 2019, Ovinnikov, 2019], feed-forward and recurrent hyperbolic neural networks architectures [Ganea et al., 2018], and hyperbolic graph convolutional networks [Chami et al., 2019]. We extend the hyperbolic VAE framework to the task of learning hyperbolic representations from subvolumes of complex 3D images, and use this to perform unsupervised segmentation.

Self-supervision Providing self-supervision by solving pretext tasks is one common approach for learning unsupervised visual representations. Pretext tasks leverage properties of the input data or prior knowledge as supervisory signals in order to learn better representations. Examples of pretext tasks include finding the relative position of two patches sampled from an image [Doersch et al., 2015], solving jigsaw puzzles [Noroozi and Favaro, 2016], and predicting pixel movements of videos in subsequent frames [Pathak et al., 2017]. In contrast, we propose the pretext task of reconstructing implicit hierarchy in 3D voxel-grid images, to learn effective hyperbolic representations for downstream segmentation.

3 Methods

In this section, we describe our approach for learning hyperbolic representations of subvolumes (3D patches) from 3D voxel-grid data. We propose a model that comprises a 3D convolutional variational autoencoder (VAE) with hyperbolic representation space and a gyroplane convolutional layer. We train our model with self-supervision through a novel hierarchical triplet loss and multi-patch sampling scheme. Then, we cluster the learned representations using hyperbolic k-means to produce 3D segmentations. In Section 3.1, we provide an overview of hyperbolic space. In Section 3.2, we describe our VAE framework with the proposed gyroplane convolutional layer and self-supervised hierarchical triplet loss. Finally, in Section 3.3, we discuss our approach of hyperbolic clustering for segmentation.

3.1 Hyperbolic formulation

Hyperbolic space We embed subvolumes of 3D voxel-grid data in hyperbolic space, a non-Euclidean space with constant negative curvature. In negative curvature spaces, the area of a disc increases exponentially with the radius. We can think of this growth as analogous to the exponential increase of leaves at each level of a tree. Hence hyperbolic space can encode trees with arbitrarily low error [Sarkar, 2011] and can be considered as the continuous version of hierarchical structures. Unlike trees, hyperbolic space is smooth, permitting our use of deep learning on representations. For additional background on geometry and hyperbolic space, see the Appendix.

Poincaré ball model of hyperbolic geometry In this work we use the Poincaré ball as our model of hyperbolic geometry. The Poincaré ball (of curvature $c = -1$) is the open ball of radius 1 centered at the origin equipped with the *metric tensor* $\mathfrak{g}_p = (\lambda_x)^2 \mathfrak{g}_e$, where the conformal factor $\lambda_x = \frac{2}{1-||x||^2}$ and $\mathfrak{g}_e$ is the Euclidean metric tensor (i.e., the Euclidean dot product). Formally, this makes the Poincaré ball a Riemannian manifold. For an introduction to Riemannian manifolds, see the Appendix.

The distance $\mathbf{d_p}$ between points on the Poincaré ball is given by:

$$\mathbf{d_p}(x, y) = \cosh^{-1}\left(1 + 2\frac{||x-y||^2}{(1-||x||^2)(1-||y||^2)}\right) \tag{1}$$

We use the exponential and logarithm maps to map from Euclidean space to the Poincaré ball and vice versa. On the Poincaré ball, we note that the exponential and logarithm maps have the closed form expressions

$$\exp_z(v) = z \oplus \left(\tanh\left(\frac{\lambda_z||v||}{2}\right)\frac{v}{||v||}\right) \tag{2}$$

$$\log_z(y) = \frac{2}{\lambda_z}\tanh^{-1}(||-z\oplus y||)\frac{-z\oplus y}{||-z\oplus y||} \tag{3}$$

where $\oplus$ denotes Mobius addition, which was introduced by Ungar [2001] as a way to define vector operations on hyperbolic space.

Wrapped Normal Distribution The importance of the normal distribution in Euclidean space has led to many generalizations of the normal distribution to Riemannian manifolds. We use the wrapped normal distribution [Mathieu et al., 2019, Nagano et al., 2019], which can be defined on an arbitrary Riemannian manifold as the push-forward measure obtained by mapping the normal distribution in Euclidean space along the manifold's exponential map. On the Poincaré ball, the probability density function of the wrapped normal with mean μ and covariance Σ is:

$$\mathcal{N}_P(z|\mu, \Sigma) = \mathcal{N}_E(\lambda_\mu(z)|0, \Sigma)\left(\frac{\mathbf{d_p}(\mu, z)}{\sinh(\mathbf{d_p}(\mu, z))}\right) \tag{4}$$

where the subscripts P, E indicate distributions over the Poincaré ball and Euclidean space, respectively. We use the sampling and reparametrization scheme of Mathieu et al. [2019] in order to sample and train our VAE using the wrapped normal distribution.

3.2 Unsupervised hyperbolic representation learning

3D hyperbolic variational autoencoder We propose a hyperbolic VAE that consists of a 3D convolutional encoder and decoder to handle 3D input. Our 3D convolutional encoder maps sampled 3D subvolumes of the input volume into hyperbolic space and produces the parameters of the variational posterior. Our 3D convolutional decoder then reconstructs the 3D subvolumes from sampled latent hyperbolic representations. To ensure that both the encoder and decoder respect the geometry of the latent space, we follow Mathieu et al. [2019] and apply an exponential map to the output of the encoder, and use our novel gyroplane convolutional layer as the first layer of the decoder. We define the prior and variational posterior to be the wrapped normal distribution, which encourages our hierarchical representations to spread out on the Poincaré ball. Figure 1 illustrates an overview of our VAE framework.

Our variational autoencoder takes as input a patch of fixed size $m \times m \times m$. The model learns representations of subvolumes from the input (X in Figure 1) that can subsequently be used to perform voxel-level segmentation of the whole volume. To learn hierarchical structure in 3D images, we train the VAE on 3D patches generated using a multi-scale sampling scheme that samples patches of size $r \times r \times r$, where size r is randomly sampled and resized to m. Our method learns to embed each patch as part of a hierarchy in hyperbolic space.

Gyroplane convolutional layer For learning better hyperbolic representations of 3D images, we introduce a gyroplane convolutional layer to effectively map from hyperbolic embedding space to Euclidean space. This allows us to keep the advantages of convolutional layers, such as locality, weight sharing, and translation equivariance. Our model's encoder output (μ in Figure 1) has a product manifold structure, since it is a Cartesian product of vectors in hyperbolic space. To map this to a 3D image in Euclidean space, we generalize the usual Euclidean convolutional layer by replacing the Euclidean affine transformation with an affine transformation on the manifold.

One way to define an affine transformation on the Poincaré ball is the gyroplane layer [Ganea et al., 2018]. The derivation of the gyroplane layer is motivated by the fact we can express a Euclidean affine

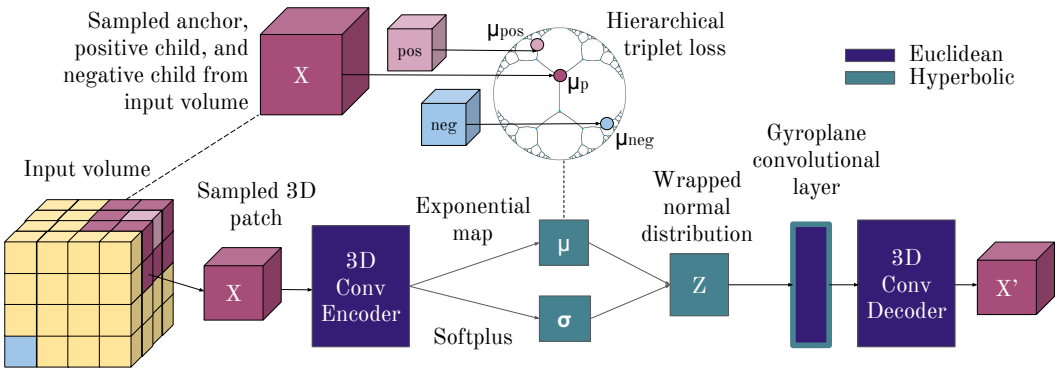


Figure 1: Our method learns hyperbolic representations of subvolumes of 3D voxel-grid data through a 3D hyperbolic VAE with a gyroplane convolutional layer. We enhance the VAE training objective with a self-supervised hierarchical triplet loss that facilitates learning implicit hierarchical structure within the VAE's hyperbolic latent space.

transformation as:$\langle a, z-p\rangle = (\langle a, z-p\rangle)||a||\mathbf{d_E}(z, H_{a,p})$ where $\mathbf{d_E}$ is Euclidean distance and $H_{a,p} = \{z \in \mathbb{R}^p | \langle a, z-p\rangle = 0\}$. $H_{a,p}$ is called the decision hyperplane. Ganea et al. [2018] defines the gyroplane layer $f_{a,p}$ from this formulation by replacing each component with its hyperbolic equivalent:

$$f_{a,p}(z) = \left(\langle a, \log_p(z)\rangle_p\right) |a|_p \mathbf{d_p}(z, H_{a,p}) \tag{5}$$

where $H_{a,p}$ is the hyperbolic decision boundary $H_{a,p} = \{z \in \mathcal{B} | \langle a, \log_p(z)\rangle = 0\}$, and the distance to the hyperbolic decision boundary $\mathbf{d_p}(z, H_{a,p})$ is

$$\mathbf{d_p}(z, H_{a,p}) = \sinh^{-1}\left(\frac{2|\langle -p \oplus z, a\rangle|}{(1 - ||-p \oplus z||^2)||a|}\right) \tag{6}$$

We can now define our gyroplane convolutional layer by replacing the Euclidean affine transformation with the gyroplane layer. For simplicity, suppose x is a 4D tensor containing elements of the Poincaré ball and our kernel size is $k \times k \times k$, with an odd k value. Our gyroplane convolutional layer is defined as:

$$y_{r,s,t} = \sum_{\alpha=r-\lfloor k/2\rfloor}^{r+\lfloor k/2\rfloor} \sum_{\beta=s-\lfloor k/2\rfloor}^{s+\lfloor k/2\rfloor} \sum_{\gamma=t-\lfloor k/2\rfloor}^{t+\lfloor k/2\rfloor} f_{a,p}(x_{\alpha,\beta,\gamma}) \tag{7}$$

The gyroplane convolutional layer can be extended in the same way as Euclidean convolutional layers to incorporate even kernel size k, input and output channels, padding, stride, and dilation.

Self-supervised hierarchical triplet loss As our model is trained on subvolumes of the 3D input, we cannot easily obtain the *implicit* hierarchical structure of the whole volume. To encode this structure in our model, we introduce self-supervision through the reconstruction of inferred hierarchy as a pretext task. This task encourages our learned representations on the Poincaré ball to reflect parent-child relationships of the input's implicit hierarchy.

Our self-supervision takes the form of a triplet loss that encourages hierarchically-related patches to have more similar representations. Since any two arbitrary patches may have some hierarchical relationship, we sample patches for our triplet loss to capture hierarchy in a tractable way. To sample 3D patches for our triplet loss, we first sample an anchor patch that acts as our parent patch (red volume X in Figure 1). We then sample the positive patch as a smaller subpatch that resides within the anchor patch (pink volume pos in Figure 1). The anchor and positive patches form a parent-child relationship that we encourage to have closer representations in hyperbolic space, which has the interpretation as belonging to the same branch of the hierarchy (μ_p and μ_{pos} in Figure 1). The exponentially growing surface area near the edge of hyperbolic space allow this natural parent-child structure to form. We then sample a negative patch as a spatially distant patch (blue volume neg)

that does not overlap with the anchor patch. The triplet loss encourages the the negative patch's representation to belong to a different branch of the hierarchy (μ_p and μ_{neg}).

Our choice of positive and negative patches is motivated by the compositional hierarchy of 3D volumes. The hierarchical triplet loss encourages the anchor patch and a sub-patch (parent and positive child) to have similar representations, while encouraging the anchor patch and a distant patch (parent and negative child) to have dissimilar representations. In hyperbolic space, this has the interpretation of belonging to one branch of the hierarchy on the Poincaré ball and belonging to a different branch, respectively. We use this implicit hierarchy reconstruction as a pretext task to encourage learning relationships between nested objects in 3D biomedical images. Our multi-patch sampling scheme and triplet loss formulation allows representations to encode implicit structure in hyperbolic space.

Our hierarchical triplet loss can be formulated with any dissimilarity measure $\mathbf{d}$ between the encoder outputs μ (see Figure 1) of the anchor μ_{p}, positive child μ_{pos}, and negative child μ_{neg}. For our model, we take $\mathbf{d}$ to be the Poincaré ball distance $\mathbf{d_p}$ and define our triplet loss with margin α as:

$$L_{\text{triplet}}(\mu_{\text{p}}, \mu_{\text{pos}}, \mu_{\text{neg}}) := \max(0, \mathbf{d_p}(\mu_{\text{p}}, \mu_{\text{pos}}) - \mathbf{d_p}(\mu_{\text{p}}, \mu_{\text{neg}}) + \alpha)$$

This formulation can be extended to any metric space by taking the dissimilarity measure $\mathbf{d}$ to be the space's metric. In particular, for our ablations using an Euclidean latent space, we take the dissimilarity measure $\mathbf{d}$ to be the Euclidean distance.

Optimization We optimize a loss function that can be decomposed as the standard evidence lower bound (ELBO) loss for variational autoencoders and our hierarchical triplet loss that encourages the learning of structure in hyperbolic space. Mathieu et al. [2019] generalized the ELBO loss to Riemannian manifold latent spaces as

$$L_{ELBO} := \int_{\mathcal{M}} \log \left(\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} \right) q_\phi(z|x) d\mathcal{M}(z) \leq \log p(x) \tag{8}$$

where $d\mathcal{M}(z) = \sqrt{|G(z)|}dz$ is the measure induced on the manifold by the Riemannian metric $G(z)$ (see Appendix). Our total loss is then formulated as

$$L_{\text{total}} = L_{\text{ELBO}} + \beta L_{\text{triplet}} \tag{9}$$

where β is a hyperparameter that controls the strength of the hierarchical triplet loss.

3.3 Segmentation by clustering representations

Hyperbolic clustering In 3D segmentation, we seek to assign each voxel v a segmentation label $s_{\text{v}} \in \{1, \ldots, n\}$, where n is the number of segmentation classes. We perform segmentation by clustering the representations of patches centered at each voxel. We first use our trained VAE encoder to generate latent representations μ_{v} for each voxel v. We do this by taking a patch of fixed size $p \times p \times p$ centered at v, upsampling or downsampling it to the encoder input size $m \times m \times m$, and then encoding the patch to retrieve μ_{v}. We then cluster the μ_v into n clusters, and produce a segmentation by assigning each v the cluster label of μ_v. We clustering through a k-means algorithm that respects hyperbolic geometry, which we derive by replacing the Euclidean centroid and distance computations of classical k-means with their counterparts in Riemannian geometry, the Fréchet mean and manifold distance. We calculate the Fréchet mean using the algorithm of Lou et al. [2020].

4 Experiments

We evaluate our method quantitatively on both synthetic 3D datasets simulating biological image data as well as the real-world Brain Tumor Segmentation (BraTS) tumor segmentation dataset. Our biologically-inspired synthetic dataset allows quantitative evaluation of segmentation at multiple levels of hierarchy, while the BraTS dataset is a well-known benchmark for 3D MRI segmentation.

Implementation details For all models, the encoder of our variational autoencoder is comprised of four 3D convolutional layers with kernel size 5 of increasing filter depth $\{16, 32, 64, 128\}$. The decoder has the same structure, except with decreasing filter depth and a gyroplane convolutional

layer as the initial layer. We use $\beta = 1e3$ as the weighting factor between L_{ELBO} and L_{triplet} and $\alpha = 0.2$ as the triplet margin. In all experiments, we fix the representation dimension to be $d = 2$, and show latent dimension ablations in the Appendix. We train our model using the Adam optimizer [Kingma and Ba, 2014]. For inference, we obtain the latent representations of $5 \times 5 \times 5$ patches densely across the full volume, and then perform hyperbolic k-means clustering, where the number of clusters k is a hyperparameter that controls the granularity of the segmentation. For quantitative evaluation, we then use the Hungarian algorithm [Kuhn, 1955] to match each predicted segmentation class with a corresponding ground truth label.

We utilize two anchor patch sampling schemes, one for input of smaller sizes and one for larger sizes. In both schemes, for a given 3D volume, we sample i patch centers v_i uniformly with patch size r, and upsampling or downsampling to size $m \times m \times m$. In the sampling scheme for smaller inputs, the patch size r is sampled uniformly, whereas in the sampling scheme for larger inputs, r is sampled log-uniformly. This scheme is motivated by the following observations: for larger patches, a small change in r is less likely to correspond to significant semantic difference, and inherent structure causes the different levels of hierarchy to naturally follow a log scale. For training on the synthetic dataset, we sample 3D volume sizes uniformly, and for BraTS we sample using the log scale.

For every sample of an anchor patch of size $r \times r \times r$, we generate a positive child patch as a smaller patch of the anchor patch as follows: the positive child patch is a subvolume within the anchor patch with size $r_{\text{child}} \times r_{\text{child}} \times r_{\text{child}}$, where $r_{\text{child}} \sim \mathcal{U}(\ell_{\min}, r - r_{\text{gap}})$, and r_{gap} is a hyperparameter representing the gap in size between the anchor size and the child size. The negative child is a patch of size $r_{\text{child}} \times r_{\text{child}} \times r_{\text{child}}$ that does not overlap with the anchor patch.

4.1 Biologically-inspired synthetic dataset

Since we want to evaluate segmentation performance at multiple levels of hierarchy and most 3D datasets do not have the necessary annotation, we first generate a synthetic dataset. This dataset enables a more thorough evaluation of the effectiveness of our model for unsupervised 3D segmentation. Our synthetic dataset is inspired by cryo-ET images of cells. Each volume in our synthetic dataset contains multiple levels of hierarchy with the objects at each level differentiated by texture, size, and shape. Figure 2 shows an example input volume with sampled slices shown. Our dataset consists of 120 total volumes, which we split into 80 training, 20 validation, and 20 test examples. Each synthetic volume has size $50 \times 50 \times 50$.

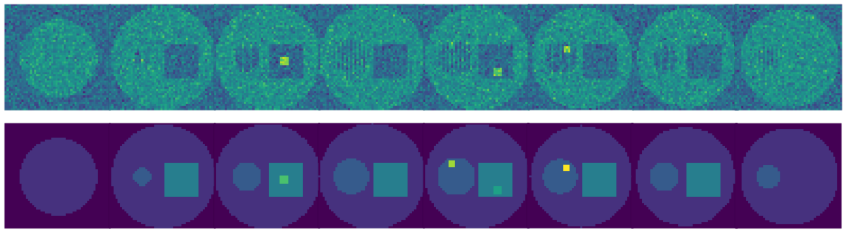


Figure 2: Sampled 2D slices from a 3D volume in our biologically-inspired synthetic dataset. The top row showcases the raw input data, and the bottom row showcases the ground truth segmentation.

Each 3D image of our biologically-inspired synthetic dataset consists of three levels of hierarchy. The first level of hierarchy (*Level 1*) has a noisy background and an outer sphere of radius $r \sim \mathcal{N}(25, 1)$. Using a cell analogy, this represents the entire cell. The second level (*Level 2*) consists of spheres ("vesicles") and cuboids ("mitochondria"). Their sizes are randomly sampled with radius of $r \sim \mathcal{N}(8, 0.5)$ and with side length of $s \sim 2 \cdot \mathcal{N}(8, 0.5)$, respectively. In the third level (*Level 3*) we introduce small spheres and cuboids ("protein aggregates") in the vesicle spheres and mitochondria cuboids respectively. The *Level 3* proteins have a radius of $r \sim \mathcal{N}(2, 0.2)$ and side length of $s \sim 2 \cdot \mathcal{N}(3, 0.15)$, respectively. The location of each object is sampled randomly, subject to the restriction that objects in Level $i + 1$ are entirely contained within an object in Level i.

Each instance of a shape with a particular size is also given its own unique texture to mimic the different organelles of the cell. The color of each object is chosen randomly, according to a standard normal distribution. We also apply pink noise with magnitude $m = 0.25$ to the volume as it is commonly seen in biological data.

To demonstrate segmentation performance on objects at different scales, we evaluate on the three levels of hierarchy defined above and use the average class DICE score to compare segmentation performance. Since our model is unsupervised, we assign segmentation classes to ground truth labels using the Hungarian algorithm. See results in Table 1 and Table 2. We also show results on a more challenging irregular synthetic dataset in the Appendix.

Table 1: Comparison with prior approaches on synthetic dataset

	DICE *Level 1*	DICE *Level 2*	DICE *Level 3*	SUPERVISION TYPE
ÇIÇEK ET AL. [2016]	0.968	0.829	0.668	3D SEMI-SUPERVISED
ZHAO ET AL. [2019]	0.989	0.655	0.357	3D SEMI-SUPERVISED
NALEPA ET AL. [2020]	0.530	0.276	0.112	3D UNSUPERVISED
JI ET AL. [2019]	0.589	0.291	0.150	2D TO 3D UNSUPERVISED
MORIYA ET AL. [2018]	0.628	0.311	0.141	3D UNSUPERVISED
OURS	**0.952**	**0.541**	**0.216**	3D UNSUPERVISED

Table 2: Ablation studies on synthetic dataset

LATENT SPACE	CONFIGURATION	DICE *Level 1*	DICE *Level 2*	DICE *Level 3*
EUCLIDEAN	BASE	0.784	0.322	0.109
	TRIPLET	0.761	0.342	0.153
HYPERBOLIC	BASE	0.832	0.352	0.135
	GYROCONV	0.905	0.473	0.204
	TRIPLET	0.945	0.534	**0.222**
	GYROCONV & TRIPLET	**0.952**	**0.540**	0.216

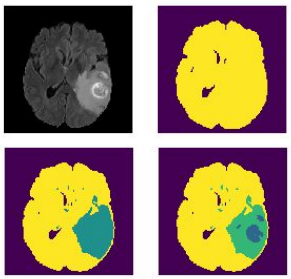
Comparison with prior approaches Table 1 shows quantitative comparison of our method with prior state-of-the-art 3D unsupervised and 2D unsupervised (which we extend to 3D) models. In addition, we also compare our method to prior semi-supervised work, as unsupervised 3D segmentation is a relatively unexplored field, and we provide baselines with different levels of supervision for additional reference. Çiçek et al. [2016] was trained with 2% of the ground truth slices in each of the xy, yz, and xz planes, and Zhao et al. [2019] was trained with one fully annotated atlas, which can both still be expensive given the large size of many 3D biomedical images. Ji et al. [2019] was implemented using the authors' original code and extrapolated to 3D by applying the method to each slice. For Nalepa et al. [2020] and Moriya et al. [2018], we re-implemented their methods as the original code was unavailable. Our model performs significantly better than all unsupervised prior work at all levels of hierarchy. We also perform comparably to the semi-supervised approach of Zhao et al. [2019], despite not using any labels.

Ablation Table 2 presents ablation studies on the hierarchical synthetic dataset comparing our contributions: Euclidean vs. hyperbolic representations, the addition of our gyroplane convolutional layer, and the addition of our hierarchical triplet loss. The base Euclidean configuration is the 3D convolutional VAE with Euclidean latent space, no gyroplane convolutional layer, and trained with just the ELBO loss. The triplet Euclidean configuration adds the hierarchical triplet loss to the base Euclidean configuration. The base hyperbolic configuration is the same as the base Euclidean configuration except with hyperbolic latent space. The triplet configuration is the hyperbolic analogue of the Euclidean triplet configuration, and gyroconv configurations have the addition of the gyroplane convolutional layer.

Hyperbolic representations outperform their Euclidean counterparts in all experiments. We attribute this to the more efficient and better organization of hyperbolic representations. When we introduce our self-supervised triplet loss, performance improves significantly for our hyperbolic models. Performance for our Euclidean model does not improve as much, likely due to information loss in representing hierarchical input. Introducing the gyroplane convolutional layer shows clear improvement over the base hyperbolic model, which demonstrates the benefit of having a convolutional layer that respects the geometry of the latent space. The combination of the triplet loss and gyroplane

Table 3: Table shows comparison on BraTS 2019 dataset. Figure shows a qualitative example where top left image is a slice from a 3D test volume, and the three other images show segmentations with $2, 3, 4$ numbers of clustering centroids respectively, illustrating multiple levels of hierarchy learned.

BRATS DATASET	DICE WT	ALGORITHM TYPE
SOTA [JIANG ET AL., 2019]	0.888	3D FULLY-SUP.
ZHAO ET AL. [2019]	0.648	3D SEMI-SUP.
ÇIÇEK ET AL. [2016]	0.760	3D SEMI-SUP.
JI ET AL. [2019]	0.211	2D-TO-3D UNSUP.
MORIYA ET AL. [2018]	0.425	3D UNSUP.
NALEPA ET AL. [2020]	0.495	3D UNSUP.
OURS	**0.684**	3D UNSUP.



convolutional layer exhibits the most gain over the base hyperbolic model, with smaller gains over the model with just the added triplet loss. We show the importance of our hierarchical self-supervision for learning effective representations that capture implicit hierarchical structure.

4.2 Brain Tumor Segmentation challenge dataset

The BraTS 2019 dataset is a public, well-established benchmark dataset containing 3D MRI scans of brain tumors and voxel-level ground truth annotations of tumor segmentation masks. The scans are of dimension $200 \times 200 \times 155$ and have four modalities; we use the FLAIR modality, which is the most commonly used one-modality input. We use the same evaluation metric as in the BraTS challenge, and compare DICE score on whole tumor (WT) segmentation, which is detectable solely from FLAIR. There are 259 high grade glioma (HGG) labelled training examples, which we split into 180 train, 39 validation, and 40 test examples. We do not use the official validation or test sets because the ground truth annotations for these sets are not publicly available.

Table 3 shows the comparison of our results against prior work. For fair comparison, all baselines are trained with only the FLAIR modality. The only exception is the current state-of-the-art fully-supervised result [Jiang et al., 2019] in Table 3, which uses all 4 modalities. We show this for reference as an upper bound; the reported number is trained on the full train set and evaluated on the BraTS test set.

Our best model performs significantly better than all prior unsupervised methods, and in addition outperforms one 3D semi-supervised model. This demonstrates the ability of our self-supervised hyperbolic representations to effectively capture the hierarchical structure in individual brain scans. We use a granular segmentation with three clusters for quantitative evaluation in order to capture the tumor, brain, and background, then use the Hungarian algorithm for assignment. In addition, we also show qualitative results for our model (see Figure 3), which include byproduct segmentations from the same model with different numbers of clusters specified, showcasing additionally discovered features in the scan that could be clinically useful.

5 Conclusion

We propose a method for learning hyperbolic representations of 3D voxel-grid images that captures the *implicit* hierarchical structure in biomedical data, and show that these representations are well suited for the task of unsupervised 3D segmentation. We conduct our representation learning through a hyperbolic 3D convolutional VAE with a novel gyroplane convolutional layer that respects hyperbolic geometry. We then enhance the VAE training objective with a self-supervised hierarchical triplet loss that learns *implicit* hierarchical structure within the VAE's hyperbolic latent space as a pretext task. We demonstrate that hyperbolic clustering of learned voxel-level representations can be used to achieve state-of-the-art unsupervised 3D segmentation on synthetic hierarchical datasets and the real-world BraTS dataset.

References

S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4:170117, 2017.

S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. T. Shinohara, C. Berger, S. M. Ha, M. Rozycki, et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint arXiv:1811.02629*, 2018.

M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

I. Chami, R. Ying, C. Re, and J. Leskovic. Hyperbolic graph convolutional neural networks. *Advances in Neural Information Processing Systems 32 (NIPS 2019)*, 2019.

J. Chen, L. Yang, Y. Zhang, M. Alber, and D. Z. Chen. Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. In *Advances in neural information processing systems*, pages 3036–3044, 2016.

Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

A. V. Dalca, J. Guttag, and M. R. Sabuncu. Anatomical priors in convolutional networks for unsupervised biomedical segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9290–9299, 2018.

C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

Q. Dou, L. Yu, H. Chen, Y. Jin, X. Yang, J. Qin, and P.-A. Heng. 3d deeply supervised network for automated segmentation of volumetric medical images. *Medical image analysis*, 41:40–54, 2017.

T. Falk, D. Mai, R. Bensch, Ö. Çiçek, A. Abdulkadir, Y. Marrakchi, A. Böhm, J. Deubner, Z. Jäckel, K. Seiwald, et al. U-net: deep learning for cell counting, detection, and morphometry. *Nature methods*, 16(1):67–70, 2019.

O.-E. Ganea, G. Becigneul, and T. Hoffmann. Hyperbolic neural networks. *Advances in Neural Information Processing Systems 31 (NIPS 2018)*, 2018.

S. Gur, L. Wolf, L. Golgher, and P. Blinder. Unsupervised microvascular image segmentation using an active contours mimicking neural network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10722–10731, 2019.

M. H. Hesamian, W. Jia, X. He, and P. Kennedy. Deep learning techniques for medical image segmentation: Achievements and challenges. *Journal of digital imaging*, 32(4):582–596, 2019.

X. Ji, J. F. Henriques, and A. Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019.

Z. Jiang, C. Ding, M. Liu, and D. Tao. Two-stage cascaded u-net: 1st place solution to brats challenge 2019 segmentation task. In *International MICCAI Brainlesion Workshop*, pages 231–241. Springer, 2019.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

T. Kitrungrotsakul, X.-H. Han, Y. Iwamoto, L. Lin, A. H. Foruzan, W. Xiong, and Y.-W. Chen. Vesselnet: A deep convolutional neural network with multi pathways for robust hepatic vessel segmentation. *Computerized Medical Imaging and Graphics*, 75:74–83, 2019.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

A. Lou, I. Katsman, Q. Jiang, S. Belongie, S.-N. Lim, and C. De Sa. Differentiating through the fr\'echet mean. *arXiv preprint arXiv:2003.00335*, 2020.

E. Mathieu, C. L. Lan, C. J. Maddison, and R. T. Y. W. Tee. Continuous hierarchical representations with poincaré variational auto-encoders. *Advances in Neural Information Processing Systems 32 (NIPS 2019)*, 2019.

B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

F. Milletari, S.-A. Ahmadi, C. Kroll, A. Plate, V. Rozanski, J. Maiostre, J. Levin, O. Dietrich, B. Ertl-Wagner, K. Bötzel, et al. Hough-cnn: deep learning for segmentation of deep brain regions in mri and ultrasound. *Computer Vision and Image Understanding*, 164:92–102, 2017.

T. Moriya, H. R. Roth, S. Nakamura, H. Oda, K. Nagara, M. Oda, and K. Mori. Unsupervised segmentation of 3d medical images based on clustering and deep representation learning. In *Medical Imaging 2018: Biomedical Applications in Molecular, Structural, and Functional Imaging*, volume 10578, page 1057820. International Society for Optics and Photonics, 2018.

Y. Nagano, S. Yamaguchi, Y. Fujita, and M. Koyama. A wrapped normal distribution on hyperbolic space for gradient-based learning. *Proceedings of Machine Learning Research*, 97, 2019.

J. Nalepa, M. Myller, Y. Imai, K. ichi Honda, T. Takeda, and M. Antoniak. Unsupervised segmentation of hyperspectral images using 3d convolutional autoencoders. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2020.

M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pages 6338–6347, 2017.

M. Nickel and D. Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. *arXiv preprint arXiv:1806.03417*, 2018.

M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

I. Ovinnikov. Poincar\'e wasserstein autoencoder. *arXiv preprint arXiv:1901.01427*, 2019.

D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2701–2710, 2017.

R. Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pages 355–366. Springer, 2011.

A. A. Ungar. Hyperbolic trigonometry and its application in the poincaré ball model of hyperbolic geometry. *Computers & Mathematics with Applications*, 41(1-2):135–147, 2001.

Z. Xu and M. Niethammer. Deepatlas: Joint semi-supervised learning of image registration and segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 420–429. Springer, 2019.

J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5147–5156, 2016.

A. Zhao, G. Balakrishnan, F. Durand, J. V. Guttag, and A. V. Dalca. Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8543–8553, 2019.

H. Zheng, Y. Zhang, L. Yang, P. Liang, Z. Zhao, C. Wang, and D. Z. Chen. A new ensemble learning framework for 3d biomedical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5909–5916, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Instructions to reproduce results can be found in Implementation details in Section 4. The BraTS dataset is open source. We also plan to publicly release all code and synthetic datasets.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix. All models are trained with 1 Titan RTX GPU.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We plan to publicly release both new synthetic datasets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]