TUM Data Innovation Lab

Munich Data Science Institute (MDSI)

Technical University of Munich

&

TUM Chair of Mathematical Modeling of Biological Systems and Helmholtz AI

Final report of project:

Topology regularised foundation model for medical image segmentation

Authors	Philipp Endres, Mikhail Konov, Hanyi Zhang, Enric Rabasseda i Raventós
Mentor(s)	Dr. Tingying Peng, Dr. Bastian Rieck, Dr. Carsten Marr, M.Sc. Valentin Koch, M.Sc. Lion Gleiter
TUM Mentor	Dr. Alessandro Scagliotti
Project lead	Dr. Ricardo Acevedo Cabra (MDSI)
Supervisor	Prof. Dr. Massimo Fornasier (MDSI)

Feb 2024

Acknowledgements

We would like to thank our mentors for their continuous support and guidance. We also want to thank the MDSI for making this project possible. This work was supported by the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI).

Abstract

Image segmentation can aid medical professionals by highlighting crucial structures. The recently released Segment Anything Model (SAM) allows to use prompts for a more interactive image segmentation. In this project we investigated the applicability of SAM in the medical domain, using OCT and Organoid datasets. For both datasets the segmentation results obtained with SAM out of the box proved to be suboptimal. We therefore followed recent approaches to fine-tune SAM on our data and improved the performance by additional techniques like pseudocoloring and topological data analysis. In the end we obtained mean IOU scores of 43.05% on OCT data and 79.8% on Organoid data. Moreover, we present a semi-automatic annotation process for organoid images.

Contents

1 Introduction

1.1 Problem definition and goals of the project

Foundation models like GPT3 [1] are recently getting popular in the deep learning community. They are trained on broad sets of data and aim to be generally applicable with minimal fine-tuning. Therefore they can be used as a foundation of task-specific models, e.g. BioGPT [2] for biomedical data. The segment anything model (SAM) is such a foundation model released in 2023 by Kirillov et al.[3]. This model is particularly interesting for image segmentation, since it allows to use prompts, e.g. a bounding box around the object that should be segmented.

We think this feature could be very interesting in the medical domain. Here experts like doctors could be assisted in their diagnosis by an interactive segmentation application. Therefore, we want to explore the effectiveness of the segment anything model on medical image data. For this we look at optical coherence tomography data (OCT) and organoid data. In order to fine-tune SAM we also want to incorporate topological information into our model by using recent developments in topological data analysis. [1]

In the following, we first discuss the state of the art foundation models for image segmentation in Section 1.2. Then we introduce topological data analysis in Section 2. Following this we present our results on OCT data in Section 3 and organoid data in Section 4.

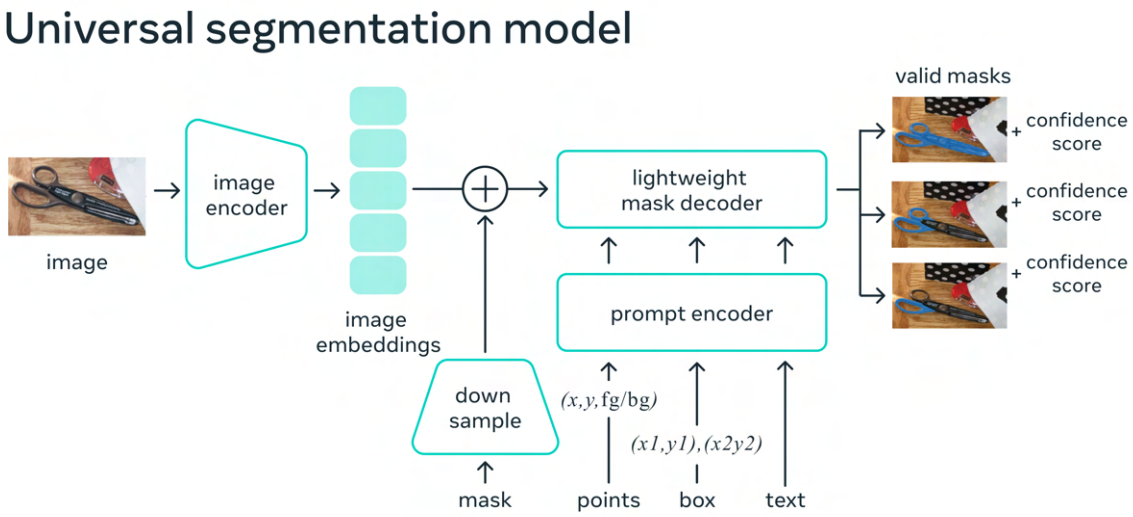


Figure 1: *The Segment Anything Model consists of an image encoder, a prompt encoder and a mask decoder. Prompts can be points, boxes and text. It also has the option to reuse masks that were previously predicted to obtain better masks and the option to return multiple masks with corresponding confidence scores (Image source: [3]).*

[1] Our code for OCT data can be found at https://github.com/philippendres/DILabHelmholtzOCT. Our code for Organoid data can be found at https://github.com/enricrabasseda/DILabHelmholtzOrganoid.

1.2 State of the art foundation models in image segmentation

The Segment Anything Model is currently the state-of the-art foundation model for image segmentation [3]. It was trained on a set of 1 billion masks and aims to allow instance segmentation for any given image. SAM consists of an image encoder, a prompt encoder and a mask decoder, i.e. SAM takes an image and a prompt as input and outputs a segmentation mask (see Figure 1 for details). This prompt can be a bounding box or a point in the image.

SAM's image encoder is a Vision transformer which has 632 million parameters [3]. The default image encoder is called ViT-Huge. There are also the variants ViT-Large and ViT-Base with far less parameters. In our experiments we predominantly used the ViT-Base variant, since its performance is still good and it is faster than ViT-Huge. SAMs prompt encoder and mask decoders on the other hand are smaller models with 4 million parameters each.

Recent papers started to fine-tune SAM on specific domains. In the medical domain Ma et al. and Zhang et al. presented fine-tuned SAM models. Ma et al. propose to re-train the mask decoder, since it is comparatively small and therefore allows fast training [4]. Zhang et al. propose to use low-rank-based fine-tuning techniques to the image encoder[5]. We mostly follow the ideas of Ma et al..

2 Topological Data Analysis

One of the main objectives of this project has been to implement recent advancements of topological data analysis [6, 7, 8, 9, 10]. All the techniques of this pioneer area are nowadays extensively getting used for different tasks. More concretely, in our case, the topological loss presented in [10] is employed.

2.1 Relevant definitions and techniques

Extracting the topological information of volumes can be done in different ways. For the characteristics and the goal of this project, cubical complexes represent the data. Persistent homology is used to compare topological features. Then, these topological features are compared with persistence diagrams.

2.1.1 Cubical complexes

A given volume $\mathcal{V}$, which is a d-dimensional tensor of shape $n_1 \times \cdots \times n_d$, is represented as a *cubical complex* C. This cubical complex contains individual voxels of the volume $\mathcal{V}$ as vertices, and connectivity information about their neighborhoods via edges, squares and higher-dimensional counterparts.

In the case of images, like in the data of this project, only vertices and lower-dimensional connectivity information are considered. The concept of a volume can be understood as a generalisation of an "image" (i. e. every image is a 2-dimensional cubical complex).

Cubical complexes provide a fundamental way to represent volume data. With them it is possible to study topological features of different dimensions. These topological features comprise connected components (0D), cycles (1D) and voids (2D). In our case, each entry of a cubical complex (voxel) is the analogue of a pixel.

2.1.2 Persistent homology

Persistent homology is a technique to calculate multi-scale topological features. This technique is particularly appropriate in our setting. Our model learns a likelihood function $f : \mathcal{V} \mapsto \mathbb{R}$. To every voxel $x \in \mathcal{V}$ this function f assigns a probability of detecting an object in the voxel.

For a likelihood threshold $\tau \in \mathbb{R}$, a cubical complex $C^{(\tau)} \coloneqq \{x \in \mathcal{V} \mid f(x) \geq \tau\}$ is obtained, and consequently a different set of topological features corresponding to this cubical complex $C^{(\tau)}$. Given that volumes are finite, their topology only changes at a finite number of thresholds $\tau_1 \geq \cdots \geq \tau_m$, and a nested sequence of cubical complexes $\emptyset \subseteq C^{(\tau_1)} \subseteq \cdots \subseteq C^{(\tau_m)} = \mathcal{V}$ is obtained. Computing this sequence can be computationally expensive, but recent papers showed that this can also be done efficiently (see [6]).

We observe that cubical complexes are related to simplicial complexes. Indeed, cubical complexes follow the same concept as simplicial complexes but use squares as their build-

ing blocks instead of triangles. Thus, persistent homology can be thought as a discretized version of simplicial homology.

2.1.3 Persistence diagrams

Persistent homology tracks topological features across all complexes in the filtration presented above, representing each topological feature as a tuple (τ_i, τ_j), where $\tau_i \geq \tau_j$, indicating the cubical complexes $C^{(\tau_i)}$ and $C^{(\tau_j)}$ in which the topological feature was "created" and "destroyed", respectively. These tuples form a multi-scale shape descriptor of all topological features of a dataset.

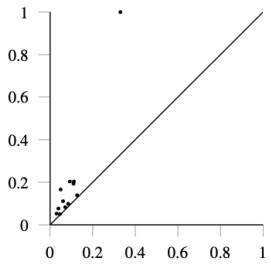


Figure 2: A persistence diagram of 1-dimensional topological features (cycles). Source: [7].

All this information is saved in the persistence diagrams. The tuples of k-dimensional features, with $0 \leq k \leq d$, are stored in the *k-th persistence diagram* $\mathcal{D}_f^{(k)}$ of the data set. These diagrams encode all possible thresholds τ at the same time, thus capturing geometrical information of the data. See Figure 2 for an example. We will denote as $\mathcal{D}_f$ the combination of all persistence diagrams' tuples of dimensions $0 \leq k \leq d$.

Given a tuple (τ_i, τ_j) in a persistence diagram, its *persistence* is defined as $\text{pers}(\tau_i, \tau_j) \coloneqq |\tau_i - \tau_j|$. It measures the "duration" over which the topological feature occurs, with large values typically assumed to correspond to more stable features. All the persistence values of a persistence diagram can be summed, defining the *degree-p total persistence*:

$$\text{Pers}_p(\mathcal{D}_f) \coloneqq \sum_{(\tau_i,\tau_j)\in\mathcal{D}_f} \text{pers}(\tau_i, \tau_j)^p. \tag{1}$$

This measure can be understood as a statistic summary of all the topological activity. And it is indeed considered in the topological loss presented in the next Section 2.2.

2.2 Topological loss

Section 2.1 above presents different techniques and topological representations of the dataset. The remaining step is to compare topological information between data samples. This will be done by comparing their persistence diagrams, however note that these finite sets can contain a different number of tuples. Therefore, these are endowed with a metric by using optimal transport.

2.2.1 Wasserstein distance

Given two diagrams $\mathcal{D}$ and $\mathcal{D}'$ containing features of the same dimensionality (here we drop f and f', but in our case each diagram indeed depends on a likelihood function), their p-th *Wasserstein distance* is defined as:

$$W_p(\mathcal{D}, \mathcal{D}') \coloneqq \left(\inf_{\eta: \mathcal{D} \longrightarrow \mathcal{D}'} \sum_{x \in \mathcal{D}} \|x - \eta(x)\|_\infty^p \right)^{\frac{1}{p}}. \tag{2}$$

Where $\eta(\cdot)$ denotes a bijection. As said, $\mathcal{D}$ and $\mathcal{D}'$ have different cardinalities, but it is considered that they contain an infinite number of points of persistence zero, i.e. (τ, τ), thus a suitable bijection $\eta(\cdot)$ can be found with modern optimal transport algorithms.

It is relevant to know that persistence diagrams are stable to noise. There is a recent theorem showing that Wasserstein distance between persistence diagrams of functions is bounded by their p-norms. Additionally, persistent homology allows the calculation of gradients with respect to the parameters of the likelihood function f. These two properties support strongly the implementation of Wasserstein distance in the topological loss presented below. For more information on these two attributes refer to [10].

2.2.2 Loss term construction

Given a true likelihood f and an estimated likelihood function f', the topology-aware loss term is defined as

$$\mathcal{L}_T(f, f', p) \coloneqq \sum_{i=0}^{d} W_p\left(\mathcal{D}_f^{(i)}, \mathcal{D}_{f'}^{(i)}\right) + \mathrm{Pers}\left(\mathcal{D}_{f'}^{(i)}\right). \tag{3}$$

The first part of Eq. (3) searches for similarity between f and f' with respect to their topological features, measuring their differences with Wasserstein distance of Eq. (2). The second part can be thought as a regularisation term, which incentivizes the model to reduce overall topological activity, measured with the total persistence of Eq. (1).

Given a task-specific geometrical loss term $\mathcal{L}_G$, such as Dice Loss or Cross Entropy loss, a combined loss term is defined as $\mathcal{L} \coloneqq \mathcal{L}_G + \lambda \mathcal{L}_T$, where $\lambda \in \mathbb{R}_{>0}$ is a hyperparameter adjusted to control the topological-loss impact. In practice, to speed up calculation of this loss term, each volume is downsampled to $M \times \cdots \times M$ voxels with bilinear interpolation.

3 OCT

3.1 Motivation

Optical Coherence Tomography (OCT) is a non-invasive imaging method. It is wildly used to diagnose diseases in live tissue, e. g. age-related macular degeneration in retinas [11]. In order to aid doctors in these diagnoses, researchers are developing tools to process these images. Especially, deep learning-based methods, e.g. for direct disease classification or semantic segmentation, are getting more popular [11].

They have been however limited to non-interactive image processing. The emergence of prompt-based models like SAM [3] on the other hand suggests the feasibility of an interactive model to process OCT images. With such prompts the doctor could for example mark regions of interest in the image and thereby guide the model to give more accurate predictions. Inspired by this idea, we set out to fine-tune SAM on OCT data.

3.2 Dataset

We used a private retinal OCT dataset consisting of 552 images. Each image had a corresponding ground truth segmentation with 14 classes. A sample with corresponding ground truth can be seen in Figure 3. We binarized the ground truth into binary masks for each class, since we wanted to build a model that puts out a binary mask for each prompt. We further binarized these class masks into component masks by splitting the class masks into connected components to focus on more fine-grained details. These component masks were our ground truth.

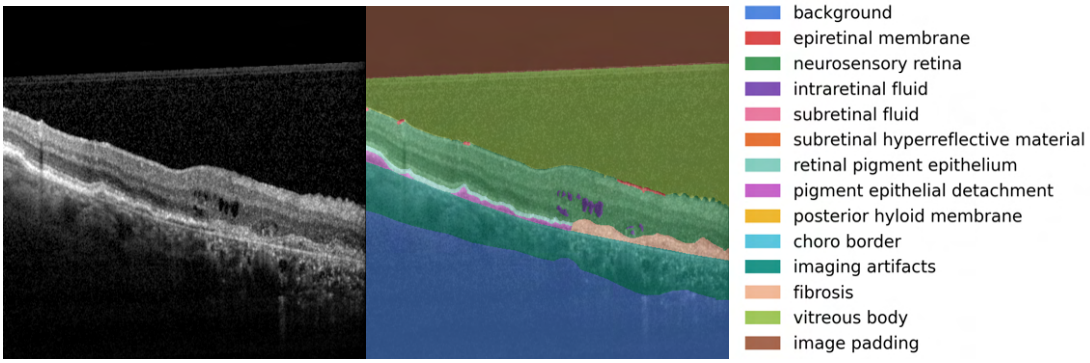


Figure 3: *Raw OCT image (on the left) and the corresponding ground truth (in the middle) colored according to the segmentation classes (on the right)*

3.3 Methodology

3.3.1 Default SAM

We began our project by applying the pre-trained SAM [3] on our data. We noticed that SAM is able to segment the rough structure of the image, but has problems with fine-grained details like intraretinal fluids. We therefore looked into different methods to

increase the performance, like incorporating adapters into the model or retraining parts of it. In the end the success of MedSAM made us follow their idea.

3.3.2 Prompt-based SAM: Bounding boxes

Following the idea of MedSAM [4], we fine-tuned SAM's mask decoder on our dataset. Since default SAM does not allow for multi-label segmentation, we used multiple bounding box prompts to differentiate between 14 different classes within our images. These prompts, were created using a ground truth segmentation so that each bounding box encompasses the ground truth segmentation mask for the respective class.

It should be noted that such an approach likely results in some form of data leakage during training, because our prompts are based on ground truth and therefore contain some information about it. However, such an approach is justified, because our model is meant to be used by medical professionals, and therefore we can expect the input prompts to be close to the actual ground truth. The overall architecture of our pipeline can be seen in Figure 4.

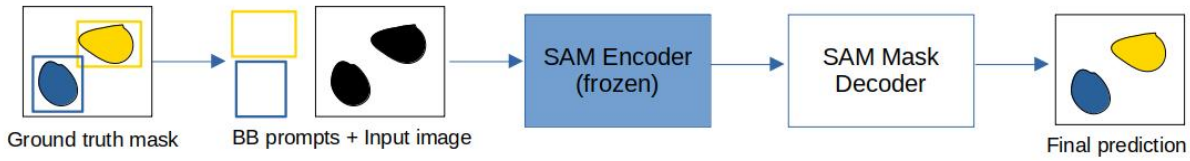


Figure 4: *The pipeline of the bounding box prompt-based SAM: Boxes around the connected components of the ground truth segmentation and images are processed by SAM. SAM's mask decoder is retrained.*

3.3.3 Prompt-based SAM: Points

We also trained a model with point prompts. Here we sampled random points from the the ground truth masks and put these as prompts. The pipeline can be seen in Figure 5.

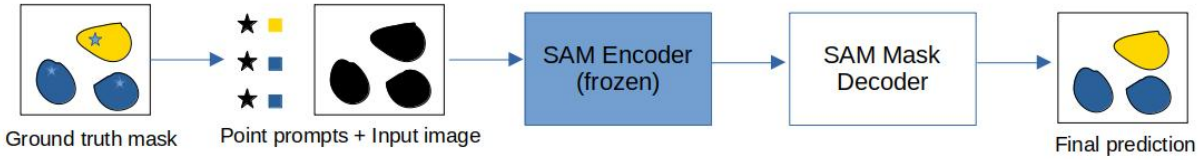


Figure 5: *The pipeline of the point prompt-based SAM: Points from each connected component and corresponding images are processed by SAM. SAM's mask decoder is retrained.*

3.3.4 Automatic SAM

Since both our prompt-based models rely on ground truth, we also wanted to train a model that does not rely on it. For this we used a bounding box around the whole image as a prompt. The goal was to predict the original ground truth mask with 14 classes.

However, since SAM only predicts one mask at a time, i.e. one class at a time, we modified its architecture to be able to predict 14 masks at a time. We achieved this by duplicating its mask-decoder 14 times, i.e. each mask decoder should predict one class. Then we applied a softmax across the classes and compared with a one-hot encoded version of the ground truth mask. During inference the softmax is replaced by a regular max. The pipeline can be observed in Figure 6

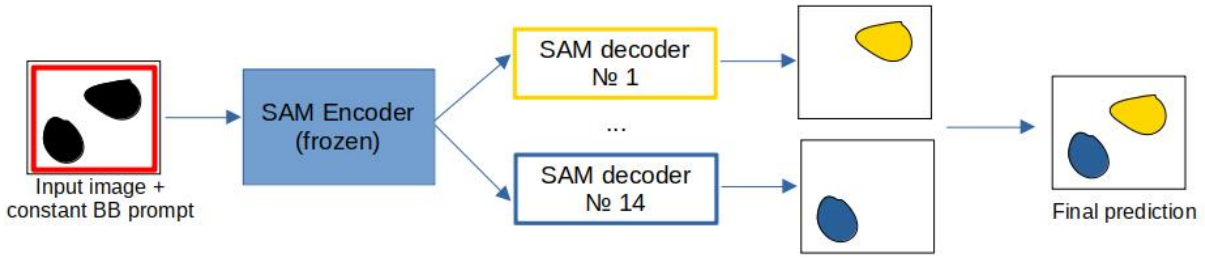


Figure 6: *The pipeline of the fine-tuned automatic SAM: Bounding boxes around the whole image and the respective image are processed by SAM's encoder. The output is then processed by 14 mask decoders (They are clones of SAM's mask decoder) where each has the task to segment one class. These mask decoders are trained to segment their class.*

3.3.5 Quality improvement

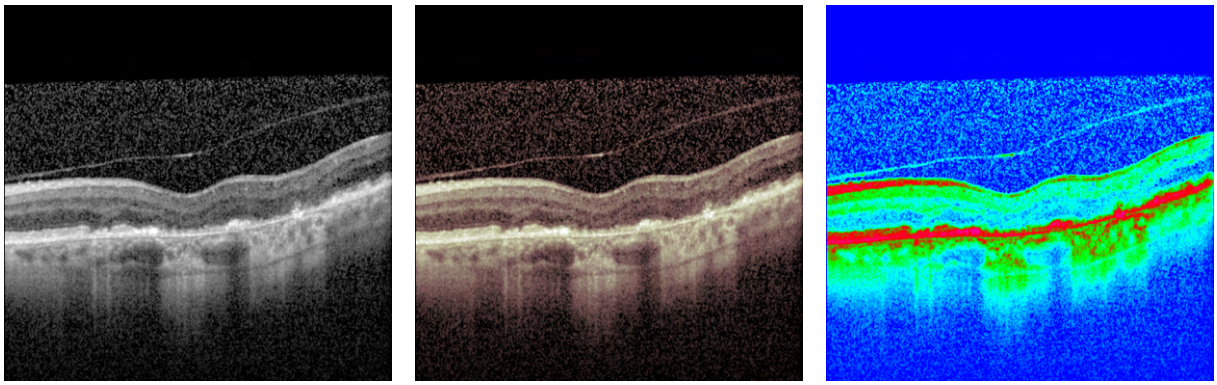


Figure 7: *Examples of pseudocoloring schemas. Left to right: 1) Initial grayscale OCT scan 2) Bone pseudocoloring schema 3) Rainbow pseudocoloring schema*

For quality improvement we focused on 3 techniques:

1. A pseudocoloring - technique that involves the recoloring of the input images to increase their contrast, thus, possibly, improving segmentations of small fine-grained regions.

2. Increasing the base model size - apart from SAM ViT-Base we also used the SAM ViT-Large model, using the SAM ViT-Huge model was discarded because of training time constraints.

3. Usage of topological loss - this loss penalizes topological (shape) dissimilarities between predicted and ground truth segmentation maps.

In practice, the techniques 2 and 3 are applied at every training step, thus, severely increasing the overall training time of the final model. That is why the highlight of this section is the pseudocoloring technique. It is applied during the image preprocessing phase and therefore has little impact on the overall training time.

Since the amount of pseudocoloring schemas that are available is fairly large (e.g. 22 in OpenCV's library) for our experiments we visually selected 2 that better highlighted fine-grained details (e.g. small pockets of intraretinal fluid within the membrane). These schemas were Bone (overall good contrast) and Rainbow (visually better highlighting intraretinal irregularities). See Figure 7 for an example.

3.4 Evaluation

3.4.1 Setup

Our 3 approaches were evaluated on our test dataset (20% of the whole datset) using 5 metrics: IoU, Accuracy, Specificity, Dice coefficient and mean AP (Average Precision). Definition of these metrics can be found in appendix section 5. Since some of our 14 classes are irrelevant (e.g. background, imaging artefacts, image padding) we calculated all of the metrics on a per-class basis. Metrics were calculated for each class based on binary masks (1 - class present, 0 - other class). Each metric was also obtained using 2 different kinds of averaging:

1. Global averaging - the metric is calculated at the same time across all pixels within all of the samples.

2. Sample averaging - the metric scores are calculated for each sample separately and then the scores are averaged across all samples.

In other words, for any suitable metric $Metric$, collections of predictions and ground truths $\{P_k\}$,$\{GT_k\}$ for our dataset of size N and a metric-specific function $Join$ that is used to join ground truths and predictions into a single data array (e.g. for Accuracy $Join$ is equal to stacking all predictions/ground truths and flattening the resulting 4D array) the formulas for global and sample averaging can be written as follows:

$$GlobalAverage = Metric(Join(P_1, ..., P_N), Join(GT_1, ..., GT_N))$$

$$SampleAverage = \frac{1}{N}\sum_{k=1}^{N} Metric(P_k, GT_k)$$

In practice, only images that contain a class were used for its metric calculation. The final model score was obtained by averaging on per-class scores for each metric.
In the following sections we will present metric scores that were obtained using sample averaging exclusively. The best scores will be highlighted in **bold**. Full tables that include global averaging scores can be found in appendix section 5.

3.4.2 Results: fine-tuning

	IoU	Accuracy	Specificity	Dice	mAP
BB, SAM ViT-Base	0,0550	0,0836	0,9274	0,0773	0,0992
BB, MedSAM	0,0162	0,0662	0,9405	0,0273	0,0894
BB, fine-tuned	**0,3660**	**0,5677**	**0,9764**	**0,4590**	**0,5216**
PT, SAM ViT-Base	0,0594	0,2874	0,6436	0,0753	0,1052
PT, MedSAM	0,0323	0,0676	**0,9370**	0,0452	0,1135
PT, fine-tuned	**0,1764**	**0,9388**	0,8084	**0,2265**	**0,2686**
Automatic SAM	0,3237	0,3697	**0,9888**	0,3697	0,5623

Table 1: *Comparison of untrained foundational models (SAM ViT-Base and MedSAM) with 2 different prompt types (BB - Bounding boxes, PT - points) to our 3 fine-tuning approaches. We see that fine-tuning improves significantly improves the model performance and the approach with BB prompts proved to be the best.*

Firstly, we evaluated how well our fine-tuning approaches perform compared to the baseline models (in our case, SAM ViT-Base and MedSAM). The results are shown in Table 1. It can be seen that for both prompt types (points and bounding boxes) fine-tuning drastically improves prediction quality with respect to our metrics (a single exception being specificity for point prompts, where untrained MedSAM takes the lead). It is also interesting to note that point prompts in general perform worse compared to bounding box prompts with regards to IoU, Dice, mAP and Specificity, while having better Accuracy (see the appendix for a definition of these metrices).

The reason for that is likely two-fold. First, point prompts, in contrast to bounding box prompts, do not contain information about the size of the segmentation region, therefore they are by design more susceptible to "overpredicting" the less prevalent class at the cost of a more prevalent class. Secondly, our design of point prompts likely exacerbated this problem by selecting a point from each connected component. As a result, less prevalent, but spatially irregular classes (e.g. intraretinal fluids) received much more related point prompts compared to more prevalent, but regular classes (e.g. neurosensory retina). Considering that most of the classes and therefore point prompts are concentrated in a relatively small region around the retina, this results in very radical changes to segmentation mask sizes. Examples of that can be seen in Figure 8, where most of the neurosensory retina is segmented as intraretinal fluid, because the latter has much more connected components and, as a result, more point prompts associated with it are evaluated compared to a single connected point prompt for the retina (because it only has one connected component).

The automatic SAM approach, on its part, proved to be worse compared to the bounding-box based fine-tuning (with regards to all metrics except Specificity), which is expected, because it doesn't have access to any information about the ground truth via prompts.

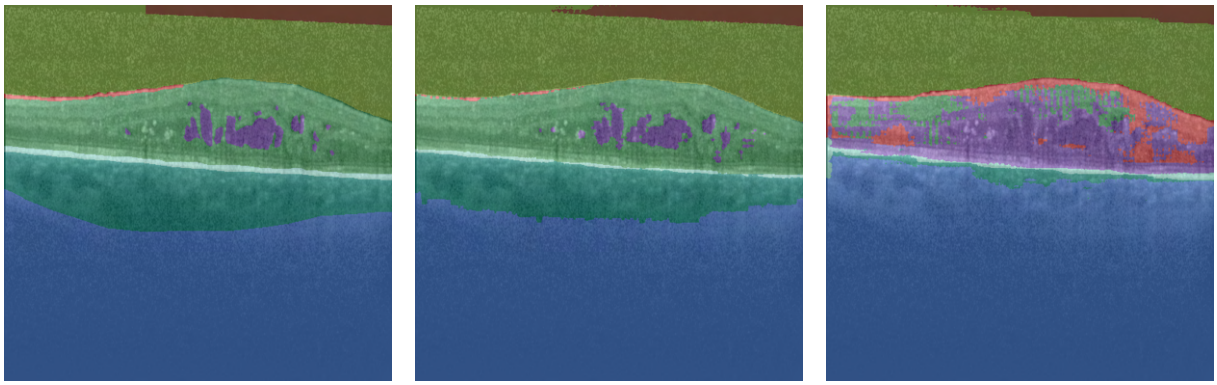


Figure 8: Left to right: 1) Ground truth segmentation 2) Fine-tuning with bounding box prompts 3) Fine-tuning with point prompts. Most of the neurosensory retina is segmented as intraretinal fluid for point prompts because of the higher number of point prompts.

3.4.3 Results: pseudocoloring

	IoU	**Accuracy**	**Specificity**	**Dice**	**mAP**
BB	0,3660	0,5677	**0,9764**	0,4590	**0,5216**
BB + Bone	**0,3723**	**0,6848**	0,9667	**0,4705**	0,5126
BB + Rainbow	0,3312	0,6702	0,9742	0,4183	0,4542

Table 2: *Evaluation results for fine-tuning with bounding box prompts on recolored images (Bone and Rainbow pseudocoloring schemas). The Bone coloring schema results in small improvements, while Rainbow worsens our results.*

Since our bounding box based approach yielded the best results during evaluation we decided to further improve it with pseudocoloring technique. The results, summarized in Table 2, were quite mixed. While the application of the Bone coloring schema resulted in an overall improvement of the metric scores (except a small decrease in Specificity and mAP), the application of the Rainbow coloring schema led to lower metric scores. It is likely, that in some cases the ability of the Rainbow coloring schema to better highlight intraretinal irregularities can confuse the model, so it will give attention to the wrong membrane parts. An example of that can be seen in Figure 9

3.4.4 Results: topological loss

In general the application of topological loss improves IoU, Specificity and Dice metrics at the cost of Accuracy (based on results from table 3). It should also be noted that using topological loss yields better results alongside the pseudocoloring technique, as the obtained improvement in case of images recolored with the Bone schema is larger than in the grayscale case. The decrease in accuracy of predictions can be explained by the per-class design of the metric calculation, which results in severe class imbalance in favor of negative class, since no class occupies the majority of the image.

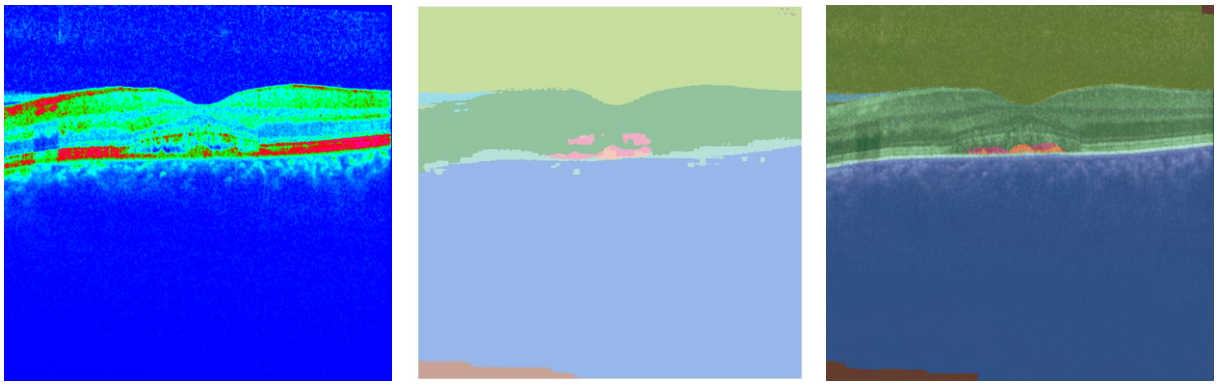


Figure 9: *Left to right: 1) Input with Rainbow pseudocoloring schema 2) Prediction 3) Ground truth. Here, the subretinal fluid is predicted along red and green segments in the recolored image, that stand out of the predominantly blue neurosensory retina, while in reality it should be predicted along the "blue pockets", confined by the red and green segments.*

	IoU	**Accuracy**	**Specificity**	**Dice**	**mAP**
BB	0,3660	**0,5677**	0,9764	0,4590	**0,5216**
BB+TL	**0,3764**	0,5621	**0,9802**	**0,4627**	0,5081
BB+Bone	0,3723	**0,6848**	0,9667	0,4705	0,5126
BB+Bone+TL	**0,3766**	0,5380	**0,9836**	**0,4657**	**0,5207**

Table 3: *Evaluation results for the fine-tuned SAM ViT-Base with bounding box prompts (BB) that utilizes topological loss (TL) during training. Results with Bone pseudocoloring schema are also shown. Using topological loss results in overall metric improvements. Topological loss also positively interacts with pseudocoloring.*

3.4.5 Results: SAM ViT-Large

Finally, we explored how our training pipeline and quality improvement techniques, mentioned in the previous sections, will work with the SAM ViT-Large model. The results of these evaluations are shown in table 4. Here we can see that SAM ViT-Large fine-tuned with bounding boxes has better metrics compared to the fine-tuned SAM ViT-Base (Specificity slightly decreases, but the overall improvement is quite clear).
Additionally, techniques that improved the quality of our predictions with SAM ViT-Base (Bone pseudocoloring, topological loss) had even more profound positive impact. For example, from table 3, IoU improvement granted by topological loss and Bone pseudocoloring was around **+0,01** for SAM ViT-Base, while for SAM ViT-Large this improvement is approx. **+0,06**. Similar improvements can also be seen with Dice, mAP and Accuracy. Therefore, we conclude that our quality improvement techniques are scalable to bigger model sizes.

	IoU	**Accuracy**	**Specificity**	**Dice**	**mAP**
BB	0,3660	0,5677	**0,9764**	0,4590	0,4950
BB-L	**0,3731**	**0,7265**	0,9653	**0,4704**	**0,5146**
BB+Bone+TL	0,3766	0,5380	**0,9836**	0,4657	0,5207
BB-L+Bone+TL	**0,4305**	**0,6995**	0,9822	**0,5217**	**0,5652**

Table 4: *Evaluation results for SAM ViT-Large with bounding box prompts (BB-L) that utilizes topological loss (TL) and the Bone pseudocoloring schema during training. The final BB-L+Bone+TL model is the best among all that were evaluated.*

Class	**Prevalence**	**BB-L+Bone+TL**		**BB-L, untrained**	
		IoU	Accuracy	IoU	Accuracy
background	0,4029	0,8235	0,9371	0,3958	0,5934
vitreous body	0,2398	0,2598	0,3488	0,0482	0,0785
neurosensory retina	0,1402	0,7269	0,8288	0,0050	0,0094
image padding	0,1036	0,8764	0,9469	0,2135	0,2651
imaging artifacts	0,0712	0,4881	0,6609	0.0197	0,0414
retinal pigment epithelium	0,0120	0,4302	0,7339	0,0029	0,0079
pigment epithelial detachment	0,0075	0,2568	0,5086	0,0272	0,0463
fibrosis	0,0061	0,4132	0,5884	0,0076	0,0109
subretinal fluid	0,0057	0,2841	0,4425	0,0098	0,0122
intraretinal fluid	0,0038	0,2242	0,4763	0,0106	0,0190
epiretinal membrane	0,0025	0,3959	0,6169	0,0023	0,0045
choroid border	0,0021	0,3290	0,5365	0,0133	0,0250
subretinal hyperreflective material	0,0020	0,2782	0,4349	0,0308	0,0382
posterior hyaloid membrane	0,0006	0,1507	0,4817	0,0035	0,0085

Table 5: *Per-class evaluation results (only IoU and Accuracy) for the fine-tuned SAM ViT-Large model with topological loss and bone pseudocoloring (BB-L+Bone+TL), compared to the untrained SAM ViT-Large model. Classes are sorted by their prevalence in the dataset. Segmentation quality for lower-prevalence classes is lower. BB-L+Bone+TL is better for all classes compared to the original untrained model.*

3.4.6 Results: per-class metrics

As was previously mentioned in section 3.4.1, some of our classes do not contain useful information. In addition to that, most of our classes are unbalanced, meaning that sizes and shapes of their segmentations vary drastically. Therefore it is interesting to look at the per-class metric values. We calculated the prevalence of every class in the datasets as a total ratio of pixels belonging to the class over the total number of pixels within the dataset. From table 5, it can be seen that the quality of our segmentation is dependent on class prevalence. In general, more prevalent classes have higher IoU and Accuracy scores. However, our best model (SAM ViT-Large fine-tuned with bounding boxes using topological loss and Bone pseudocoloring) can obtain results that are close to the global average even for low-prevalence classes (e.g. subretinal hyperreflective material, subretinal/intraretinal fluids). Additionally, it outperforms untrained SAM ViT-Large on all classes by a significant margin.

4 Organoids

For the organoids dataset not only SAM was fine-tuned, but the private dataset was labelled before the training. The objective was to create a SAM version that can deal with organoids images, from different microscopes, and detect every organoid performing instance segmentation by manually giving a box prompt for every organoid.

4.1 Private dataset creation

We received a private dataset of images of organoids taken from different microscopes. However, these images were not labelled, so it was necessary to find all the organoids in the images and create ground-truth data before fine-tuning SAM. An image example of the private dataset can be seen in Figure 10.

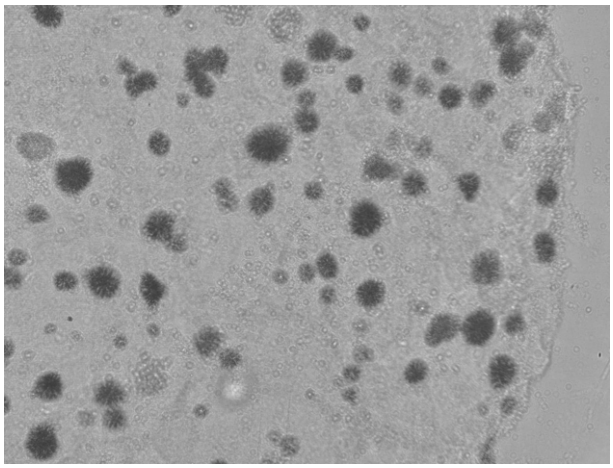


Figure 10: *Example of an image of our private dataset.*

Our first approach was to hand-label the data by finding all organoids in every image and recording a frame box for each one of them. However, this procedure was too complicated and it would have taken too long, since there were images with a big number of organoids and each box needed to be manually adjusted. Instead, after some research, we found Grounding DINO, a recent implementation of DINO [12] that could help us in doing this task and save a significant amount of time.

4.1.1 Grounding DINO

Grounding DINO [13] is an open-set object detection model which can detect any object in an image based on human inputs. The model processes (Image, Text) pairs and outputs bounding boxes for identified objects. It consists of three key components.

The model begins by extracting and enhancing features from images and text. It employs a Swin Transformer [14] for image features and a BERT-based [15] text backbone for textual features. These features are then fused using a Deformable Self-Attention-based enhancer, ensuring effective capture and combination of relevant information from both

modalities.

With enhanced text features, the model selects pertinent features based on input text. This ensures a focus on the most relevant aspects of the image, guided by accompanying textual information. This language-guided query selection is crucial for refining object detection and improving localization accuracy.

In the final stage, image and text features are integrated through a sophisticated cross-modality decoder. This decoder incorporates self-attention, image cross-attention, and text cross-attention layers, comprehensively combining information from both modalities. The result is a more nuanced understanding of the scene, enhancing the model's ability to precisely localize and identify objects in the given images.

Grounding DINO stands out as a potent tool for cross-modality information fusion, streamlining the labeling process in our organoid dataset project. The model efficiently integrates visual and textual inputs, providing precise object localization.

4.1.2 Prompt engineering

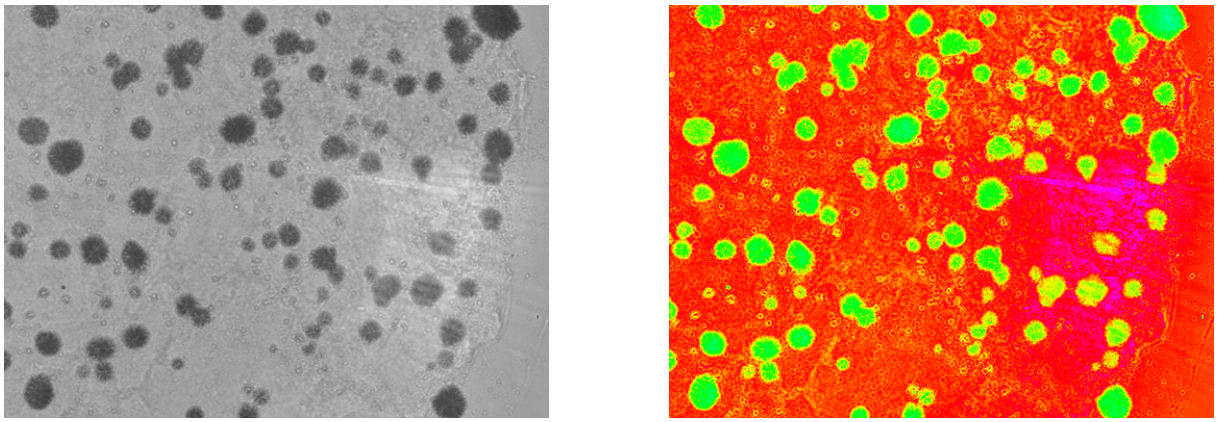
Grounding DINO has a good performance in almost all kinds of images and text prompts. However, this model was trained on the COCO dataset [16], so it is not accurate on medical images and it also does not understand technical words that are not in its vocabulary. This was a big impediment in our case, since we were dealing with organoids images and we needed to find "organoids", which is not a common word.

Our first idea to solve this problem was to use prompt-engineering techniques to find the best text prompt to describe organoids. For example, we used the CLIP (Contrastive Language-Image Pretraining) model [17] to describe organoids given an image of our dataset. Nevertheless, the result was not good, since this model was also trained on the COCO dataset and therefore giving misleading descriptions like: "*a group of frisbees sitting on a table*".

The second approach was to use Bio-GPT [2], a recent Generative Pre-trained Transformer for Biomedical Text Generation and Mining implemented by Microsoft. We implemented it into the Grounding DINO model. [2]But the modification of the architecture of Grounding DINO was too complicated and this implementation did not secure success.

Finally, we opted for the option of manually trying different text prompts and finding the best one. After many different tries we found out that the best text prompt that detects the biggest number of organoids in an image was: "*dark rounds*".

[2]An example of the generated text from BioGPT for organoids is as follows "*Organoid is an emerging technique where three dimensional organoid can be differentiated and grown in vitro. The organoid can be generated in vitro from normal cells as well as from cancer cells through the formation of organoids, and this is an important tool for investigating cancer biology in a three dimensional culture system.*"



(a) Raw Image (b) Pseudo-color Image

Figure 11: *Using HSV colormap on a grey image from our private dataset.*

4.1.3 Pipeline to generate the dataset

Given the best text prompt to Grounding DINO, the model was capable to generate framing boxes for almost all organoids in our dataset. However, there were some organoids that were not detected, and some detected objects were no organoids. We solved this issue with four additional steps in our procedure.

The first step was to use image preprocessing techniques on the original image: gamma correction and pseudo-coloring. These enhanced the performance of Grounding DINO in a large percentage of the private dataset's images.

Gamma correction is a technique to make the image and its shades look brighter or darker. This helped to make the organoids darker and the background lighter. Given an image $I \in \mathbb{R}^{H \times W}$, where each pixel $x \in I$ has bounded values $0 \leq x \leq 255$, we computed this correction pixel-wise as $\tilde{I} = (I/255)^{1/\gamma} \cdot 255$, where $\gamma \in \mathbb{R}_{>0}$ is a hyperparameter.

Pseudo-colouring, also known as false-colouring, is a technique for enhancing data visualisation by specifying colours to represent different values or features. This method is commonly applied to image or graphical representations to highlight specific features or patterns that are not easily discernible in greyscale.

In object detection, pseudo-colouring can be used to map the original greyscale image to a colour image. The logic behind this is that Grounding DINO and SAM were primarily trained using colour images. Therefore, better performance can be expected on colored images. For our objective, the colormap HSV has been used on the grey images. An example of a raw image and an HSV pseudo-colored image can be observed in Figure 11.

The second step was to delete all boxes with a significant overlap with the Non-Maximum suppression algorithm [18].

The third step was to manually delete all the boxes corresponding to objects that were

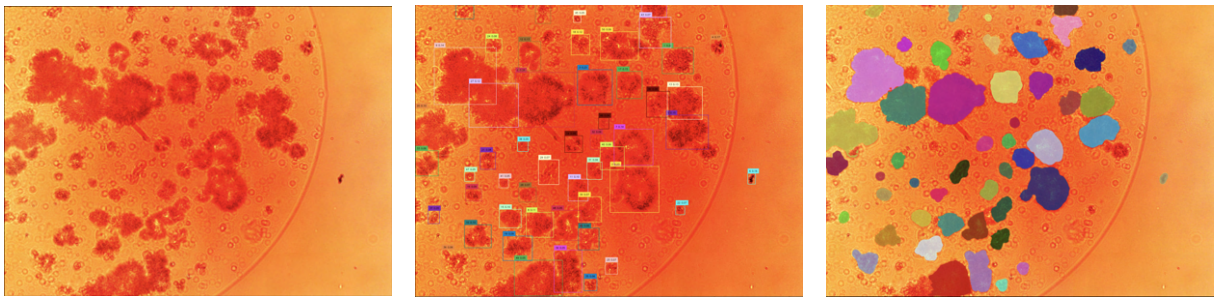


Figure 12: *Pipeline to obtain masks for the images of the private dataset. On the left, the original image; in the middle, the boxes found with Grounding DINO; on the right the masks given by SAM ViT-Large using the boxes of the second image as prompts.*

not organoids, most of these wrong boxes were framing cells.

In the fourth step we did the opposite, we manually created framing boxes for all organoids in the image that were not detected by Grounding DINO.

Finally, for every image of organoids, we obtained a set of framing boxes that surrounded all organoids in this image. To get the ground-truth, we used SAM ViT-Large to detect each of the organoids giving its corresponding framing box as a prompt to the model. Therefore, for every image, we obtained one mask and one framing box for each of the organoids present in the picture.

However, the images in our dataset also had a high resolution and SAM expects images of resolution 1024×1024. If the image does not have this resolution, SAM automatically employs bilinear interpolation to get the objective resolution. The mask output of SAM always has size 256×256, we used bilinear interpolation to resize every mask to the original image size. One example of the final result for one image can be seen in Figure 12.

We also split every original image in the dataset into different patches and adapt the found boxes and masks to every patch size. These patches were obtained with a sliding window with no overlapping. For every image 48 patches of size 324×324 were generated, with their corresponding boxes and masks. This step did not only help to deal with the image resizing of SAM, but also drastically reduced the size of the dataset.

4.2 Dataset collection and curation

The procedure explained in Section 4.1 was done for every image provided to us in the private dataset. These manual steps costed a big effort and time, but it was important to do the best possible job with every image and obtain a high-quality dataset, which would lead to a competitive model. Detailed information about the datasets can be seen in the Appendix.

After we got patches for every image, we obtained the largest dataset up to this day in organoid detection. We did research and found other open datasets of organoids. On

one hand we found semantic segmentation datasets such as OrganoID and OrganoSeg [19, 20]. On the other hand we found another instance segmentation dataset: OrgaQuant [21]. Some of these datasets were used as hold-out test datasets to see the performance of our model on non-seen organoid data. See the Appendix to observe some image examples of these datasets.

OrgaQuant [21] included intestinal organoid pictures and framing boxes for every organoid. These pictures had resolution 300×300 and 450×450. Therefore, SAM follows the same preprocessing of rescaling them to 1024×1024 and putting out masks of size 256×256 that are later rescaled to the original size of the image. We obtained one mask for every bounding box of the original dataset using SAM ViT-Large.

OrganoSeg [20] presented a dataset containing colon and colorectal-cancer organoid morphologies. These pictures had a resolution of 864×648. Every picture had a semantic segmentation mask containing all organoids in the image with the same resolution. First, we generated 4 patches with a sliding window that overlapped for every image with a final square resolution of 432x432. We did the same for the masks. Then, we employed connected component analysis to obtain one unique mask for every organoid from the original instance segmentation mask and a framing box. We saved every organoid instance in one of the three dataset's split randomly: train (60%), validation (20 %) and test (20 %).

4.3 Model training

The training of the model followed the approach of the most recent developments of fine-tuning SAM in the Medical Imaging field. We took inspiration of MedSAM and SAMed [4, 5]. Moreover, we implemented pioneer techniques such as the topological loss presented in Section 2.

4.3.1 Architecture of the model

As mentioned before in Section 1.2, SAM is composed of an image encoder, prompt encoder and mask decoder. In terms of training, it is sufficient to freeze the encoders' parameters and only train the mask decoder.

In our case, we only fine-tuned the SAM ViT-Base version. This is the smallest version of SAM but provides surprisingly good results when fine-tuned to a particular dataset. To fine-tune larger versions of SAM significant computational resources are needed.

4.3.2 Prompt in training and specifications of training

First, different experiments were done to decide which types of prompts provide better results in the training. We used boxes and points as prompts. Moreover, we gave these prompts with some generated artificial noise and without any noise. These different approaches were validated and compared on 10% of our total private dataset.

For the training, after experimenting, we loaded every mask from our private dataset with its corresponding image and bounding box. We added noise to the edges of the box and

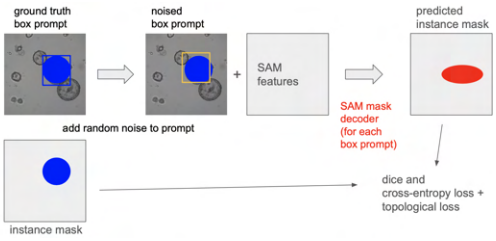


Figure 13: *Training schema of our model. We add noise to the box prompt and freeze the encoders.*

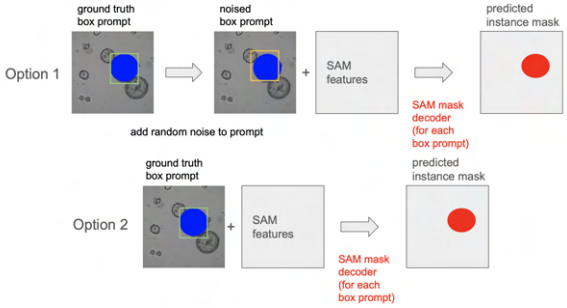


Figure 14: *Inference schema of our model, including the two possibilities: noised box prompt or ground-truth box prompt.*

gave it as a prompt to SAM to find the mask. Figure 13 shows the procedure to train the model.

Later we compared the output mask of SAM with the ground-truth mask using a combination of geometrical loss and topological loss $\mathcal{L} = \mathcal{L}_G + \lambda \mathcal{L}_T$. As geometrical loss $\mathcal{L}_G$ we used a mix of Dice and Cross Entropy loss. For more information regarding the loss functions see the Appendix. The chosen optimizer for the training was Adam.

Experimentally, we found out that it is best to use a small λ for the topological loss. We used bilinear interpolation to reduce the size of the masks when computing the topological loss and geometrical loss to 50×50 and 150×150, respectively, to save computational effort. The hyperparameters used to train the model can be found in the Appendix.

4.3.3 Inference

For inference the user needs to give a box prompt, jointly with the image, to the fine-tuned model. The model will then compute the logits for every pixel to be a mask.

We used binary interpolation to resize the given mask from size 256×256 to the given image's size. Later, the sigmoid activation function was used and a threshold of 0.5 pixel-wise to provide a binary mask. Figure 14 shows the inference procedure. As it can be seen, two different inference procedures can be done. One takes ground-truth boxes as prompts and the other adds some noise to the ground-truth boxes and later gives them as prompts.

4.4 Evaluation

4.4.1 Setup

Our model was tested on a hold-out test split consisting approximately of 17% of our private dataset and 2 hold-out datasets: test splits of OrgaQuant and OrganoSeg. We compared the fine-tuned version of SAM ViT-Base model using topological and geometrical loss to the original foundational SAM model versions, MedSAM model and fine-tuned

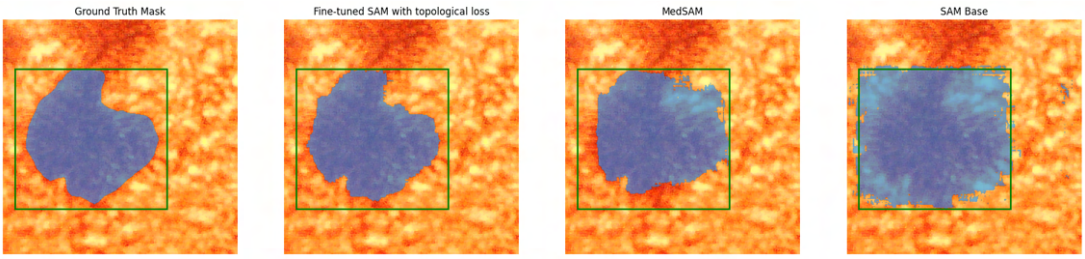


Figure 15: *Visual example of the results of the models. From left to right: 1) Original mask with the ground-truth bounding box, 2) Mask given by our fine-tuned version of SAM ViT-Base using topological loss, 3) Mask given by MedSAM, 4) Mask given by SAM ViT-Base.*

	Private				OrgaQuant				OrganoSeg			
Model	**IoU**	**Dice**	**Sens**	**AP**	**IoU**	**Dice**	**Sens**	**AP**	**IoU**	**Dice**	**Sens**	**AP**
MedSAM	0,606	0,711	0,695	0,860	0,644	0,763	0,688	0,953	0,626	0,751	0,651	0,942
SAM ViT-Base	0,701	0,796	0,891	0,876	0,945	0,967	0,970	0,992	0,801	0,881	0,840	0,929
SAM ViT-Large	0,722	0,811	0,914	0,899	**1,000**	**1,000**	**1,000**	**1,000**	0,807	0,886	0,847	0,945
SAM ViT-Huge	0,723	0,811	**0,924**	0,894	0,969	0,982	0,988	0,995	**0,810**	**0,889**	0,857	**0,948**
Fine-tuned SAM ViT-Base with Geom loss	0,792	0,864	0,899	0,953	0,670	0,790	0,989	0,942	0,742	0,837	0,895	0,926
Fine-tuned SAM ViT-Base with Topo+Geom loss	**0,798**	**0,869**	0,895	**0,956**	0,690	0,804	0,991	0,952	0,754	0,849	**0,917**	0,939

Table 6: *Table of the results for inference given the ground-truth box prompt. Contains all the metrics on the evaluation datasets for every model (**bold** best, underline second best). MedSAM did not work with organoid images and the foundational SAM models had the best performance on most of the organoid images.*

version of SAM ViT-Base but only with geometrical loss and not topological loss.

The metrics used to compare the models were Dice Coefficient (Dice), Average Precision (AP), Intersection over Union (IoU) and Sensitivity (Sens). For every dataset, these metrics were computed on each mask and we took the average on all the masks. The definitions of these metrics can be found in the Appendix. Other metrics such as Specificity (Spec) and F1 Score (F1) were evaluated. See the Appendix for the complete results.

4.4.2 Results: ground-truth box prompts

The first study was done for the case of giving ground-truth boxes to the models. Evaluation results can be checked in Table 6. The first conclusion was that MedSAM did not work well with organoid images. The second conclusion is that original foundational models could detect organoids well given a perfect bounding box. Especially, larger versions of the foundational SAM models such as SAM ViT-Large and SAM ViT-Huge gave nice results.

For SAM ViT-Large it was not surprising to see the metrics obtained on the OrgaQuant dataset. This perfect performance was due to the way the masks were generated for this dataset, explained in Section 4.2.

Regarding our fine-tuned version of SAM, we got the best results on the private dataset hold-out test split. Due to the training pipeline (see Figure 13) our model was fine-tuned

	Private				OrgaQuant				OrganoSeg			
Model	**IoU**	**Dice**	**Sens**	**AP**	**IoU**	**Dice**	**Sens**	**AP**	**IoU**	**Dice**	**Sens**	**AP**
MedSAM	0,510	0,621	0,574	0,793	0,502	0,634	0,538	0,881	0,546	0,681	0,566	0,922
SAM ViT-Base	0,636	0,737	0,801	0,823	0,704	0,775	0,733	0,850	0,733	0,831	0,769	0,904
SAM ViT-Large	0,663	0,757	0,827	0,847	0,686	0,751	0,707	0,865	0,737	0,834	0,774	0,927
SAM ViT-Huge	0,674	0,768	0,860	0,851	**0,738**	**0,801**	0,764	0,911	**0,770**	**0,860**	0,813	**0,936**
Fine-tuned SAM ViT-Base with Geom loss	0,783	0,857	**0,892**	0,951	0,671	0,789	0,980	0,928	0,734	0,830	0,886	0,912
Fine-tuned SAM ViT-Base with Topo+Geom loss	**0,790**	**0,863**	0,890	**0,953**	0,688	0,801	**0,983**	**0,942**	0,740	0,835	**0,900**	0,917

Table 7: *Table of the results for inference given a noised box prompt. Contains all the metrics on the evaluation datasets for every model (**bold** best, underline second best). MedSAM did not work on organoid images. Large versions of the foundational SAM models had a good performance. The model trained with topological loss was better than the one only trained with geometrically loss and worked well on all datasets.*

to deal with more flexible prompts instead of ground-truth boxes. However, these fine-tuned model got better results than MedSAM. For a more visual example, see Figure 15 to compare the output of the different models. More examples can be seen in the Appendix.

4.4.3 Results: noised box prompts

The second study was done for the case of giving noised box prompts to every model. This approach tried to imitate a real case usage, where a doctor will not draw perfect boxes for every organoid, since this person will not have computer precision, but will draw framing boxes close to the ideal ones.

The results, which can be seen in Table 7 are promising. Again, MedSAM did not work well on organoid images with noised prompts (even though the model was trained with noised framing boxes).

On one hand, for the private dataset, our fine-tuned version of SAM had the best performance on all the metrics. On the other hand, for the hold-out datasets, we saw that the SAM ViT-Huge model had the best performance. However, our fine-tuned version of SAM had close evaluation metrics to this larger SAM version and was also faster. Moreover, as it can also be seen above in Table 6, fine-tuning with topological loss indeed improved the performance of the resulting model compared to only using geometrical loss.

5 Conclusion

In this project we fine-tuned the Segment Anything Model [3] on medical data. We saw that this improved the performance for OCT and Organoid data. Furthermore, incorporating topological data analysis led to slight improvements.

Regarding OCT data, we mostly focused on the model design. Our main problem was how to deal with multiple segmentation classes where some were much more prevalent than others. We presented two ways to solve this problem. Firstly, we used prompts to segment one class at a time. Secondly, we modified SAM's architecture such that it produces segmentations for each class and superposes them in the end. We saw that the first approach performs better and fine-tuning on it shows significantly superior performance compared to the default SAM.

Regarding organoid data, we had to label the images first. For this we created a semi-automatic pipeline that combines Grounding DINO and pseudo-coloring techniques. The fine-tuned version of SAM using topological loss showed a competitive performance against SAM ViT-Large and SAM ViT-Huge on our private and other public datasets. Our fine-tuned model is a promising fast tool to segment organoid images and is able to deal with flexible box prompts.

For both datasets, OCT and Organoids, the results showed a significant improvement on the fine-tuned models if topological loss is employed. Furthermore, simple image preprocessing techniques such as pseudo-coloring enhanced the power of the models.

Finally, our results suggest further research on foundation models in the medical domain. We also look forward to future developments in topological data analysis.

References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[2] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, and T.-Y. Liu, "BioGPT: generative pre-trained transformer for biomedical text generation and mining," *Briefings in Bioinformatics*, vol. 23, no. 6, Sep. 2022. [Online]. Available: http://dx.doi.org/10.1093/bib/bbac409

[3] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," 2023.

[4] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," 2023.

[5] K. Zhang and D. Liu, "Customized segment anything model for medical image segmentation," 2023.

[6] H. Wagner, C. Chen, and E. Vuçini, *Efficient Computation of Persistent Homology for Cubical Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 91–106. [Online]. Available: https://doi.org/10.1007/978-3-642-23175-9_7

[7] B. Rieck, T. Yates, C. Bock, K. Borgwardt, G. Wolf, N. Turk-Browne, and S. Krishnaswamy, "Uncovering the topology of time-varying fmri data using cubical persistence," 2020.

[8] M. Moor, M. Horn, B. Rieck, and K. Borgwardt, "Topological autoencoders," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 7045–7054. [Online]. Available: https://proceedings.mlr.press/v119/moor20a.html

[9] F. Hensel, M. Moor, and B. Rieck, "A survey of topological machine learning methods," *Frontiers in Artificial Intelligence*, vol. 4, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frai.2021.681108

[10] D. J. E. Waibel, S. Atwell, M. Meier, C. Marr, and B. Rieck, *Capturing Shape Information with Multi-scale Topological Loss Terms for 3D Reconstruction*. Springer Nature Switzerland, 2022, p. 150â€“159. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-16440-8_15

[11] V. Koch, O. Holmberg, H. Spitzer, J. Schiefelbein, B. Asani, M. Hafner, and F. J. Theis, *Noise Transfer for Unsupervised Domain Adaptation of Retinal OCT Images*. Springer Nature Switzerland, 2022, pp. 699–708. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-16434-7_67

[12] M. Caron, H. Touvron, I. Misra, H. JÃ©gou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021.

[13] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding DINO: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *CoRR*, vol. abs/2103.14030, 2021. [Online]. Available: https://arxiv.org/abs/2103.14030

[15] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[16] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. DollÃ¡r, "Microsoft COCO: Common objects in context," 2015.

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[18] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013. [Online]. Available: https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013

[19] J. M. Matthews, B. Schuster, S. S. Kashaf, P. Liu, R. Ben-Yishay, D. Ishay-Ronen, E. Izumchenko, L. Shen, C. R. Weber, M. Bielski, S. S. Kupfer, M. Bilgic, A. Rzhetsky, and S. Tay, "OrganoID: A versatile deep learning platform for tracking and analysis of single-organoid dynamics," *PLOS Computational Biology*, vol. 18, no. 11, pp. 1–16, 11 2022. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1010584

[20] M. A. Borten, S. S. Bajikar, N. Sasaki *et al.*, "Automated brightfield morphometry of 3d organoid populations by OrganoSeg," *Scientific Reports*, vol. 8, p. 5319, 2018. [Online]. Available: https://doi.org/10.1038/s41598-017-18815-8

[21] T. Kassis *et al.*, "OrgaQuant: Human intestinal organoid localization and quantification using deep convolutional neural networks," *Scientific Reports*, vol. 9, no. 1, p. 12479, 2019.

Loss functions for segmentation

Here we give a short summary of common geometric segmentation losses. In our experiments we use a combination of Cross entropy loss and Dice loss called DiceCELoss. Cross entropy loss is defined for multiple classes like

$$L_{CE} \coloneqq -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(p_{i,c}),$$

where $y_{i,c}$ denotes the ground truth assignement, i.e. it is 1 if pixel i belongs to class c and 0 otherwise, and $p_{i,c}$ denotes the prediction logit.

Dice loss is defined for binary segmentation as

$$L_D \coloneqq \frac{2\sum_{i=1}^{N} p_i g_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} g_i^2},$$

where g_i is the binary ground truth at pixel i and p_i is the predicted logit. So, the numerator is the total number of pixels correctly predicted and the denominator is the total number of pixels predicted and ground truth pixels.

DiceCELoss is defined then as:

$$L_{DCE} \coloneqq \lambda_d L_d + \lambda_{ce} L_{ce}$$

with hyperparameters λ_d and λ_{ce} that balance the weight of the two loss functions on the global loss function.

Metrics for segmentation

The metrics implemented to evaluate the models have been the following:

- Intersection over Union (IoU): Measures the overlap between the predicted (A) and ground truth (B) masks.

$$\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

- Accuracy

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Negatives} + \text{Positives}}.$$

- Dice Coefficient (Dice): Similar to IoU, measures the overlap between the predicted and ground truth masks.

$$\text{Dice}(A, B) = \frac{2 \cdot |A \cap B|}{|A| + |B|}.$$

- Specificity (Spec): Ability of the model to correctly identify negative instances.

$$\text{Spec} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}.$$

- Sensitivity (Sens): Ability of the model to correctly identify positive instances.

$$\text{Sens} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}.$$

- Average Precision (AP): considers the precision-recall curve and provides a summary measure of the model's performance across different confidence thresholds. It is computed by integrating the precision-recall curve.

- F1 Score (F1): Harmonic mean of precision and recall. Offers a balanced measure of the model's performance.

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

OCT

More images from the dataset and some more evaluation results

Training hyperparameters

Hyperparameter	Value
Train split images	440
Validation split images	112
Batch size	2
Epochs	10
Learning rate	1e-4
Topological loss: dimensional topological features included	0 and 1
Topological loss: L-p in Wasserstein distance	L2-loss
Topological regularisation	None

Table 8: Hyperparameters for the models trained with topological and geometrical loss and box or point prompts.

Hyperparameter	Value
Train split images	440
Validation split images	112
Batch size	2
Epochs	10
Learning rate	1e-3

Table 9: Hyperparameters for the model trained with geometrical loss.

Hyperparameter	Value
Train split images	440
Validation split images	112
Batch size	8
Epochs	10
Learning rate	1e-3

Table 10: Hyperparameters for the automatic SAM models.

Detailed evaluation results

	IoU		Accuracy		Specificity		Dice		mAP	
	Global	Sample	Global	Sample	Global	Sample	Global	Sample	Global	Sample
BB, SAM ViT-Base	0,0736	0,0550	0,1251	0,0836	0,9307	0,9274	0,1167	0,0773	0,1197	0,0992
BB, MedSAM	0,0174	0,0162	0,0369	0,0662	0,9395	0,9405	0,0328	0,0273	0,0785	0,0894
BB, fine-tuned	**0,4487**	**0,3660**	**0,7135**	**0,5677**	**0,9767**	**0,9764**	**0,5819**	**0,4590**	**0,5745**	**0,5216**
PT, SAM ViT-Base	0,0462	0,0594	0,3069	0,2874	0,6440	0,6436	0,0784	0,0753	0,0651	0,1052
PT, MedSAM	0,0281	0,0323	0,0643	0,0676	**0,9365**	**0,9370**	0,0507	0,0452	0,0761	0,1135
PT, fine-tuned	**0,1783**	**0,1764**	**0,9138**	**0,9388**	0,8076	0,8084	**0,2572**	**0,2265**	**0,2795**	**0,2686**
Automatic SAM	0,3357	0,3237	0,3771	0,3697	**0,9885**	**0,9888**	0,3927	0,3697	0,5637	0,5623

Table 11: Comparison of untrained foundational models (SAM ViT-Base and MedSAM) with 2 different prompt types (BB - Bounding boxes, PT - points) to our 3 fine-tuned approaches.

	IoU		Accuracy		Specificity		Dice		mAP	
	Global	Sample	Global	Sample	Global	Sample	Global	Sample	Global	Sample
BB	0,4487	0,3660	0,7135	0,5677	**0,9767**	**0,9764**	0,5819	0,4590	**0,5745**	**0,5216**
BB + Bone	**0,4557**	**0,3723**	**0,8313**	**0,6848**	0,9673	0,9667	**0,5870**	**0,4705**	0,5730	0,5126
BB + Rainbow	0,4349	0,3312	0,7857	0,6702	0,9744	0,9742	0,5584	0,4183	0,5300	0,4542

Table 12: Evaluation results for fine-tuning with bounding box prompts on recolored images (Bone and Rainbow pseudocoloring schemas).

	IoU		Accuracy		Specificity		Dice		mAP	
	Global	Sample	Global	Sample	Global	Sample	Global	Sample	Global	Sample
BB	0,4487	0,3660	**0,7135**	**0,5677**	0,9767	0,9764	0,5819	0,4590	**0,5745**	**0,5216**
BB+TL	**0,4606**	**0,3764**	0,7131	0,5621	**0,9804**	**0,9802**	**0,5911**	**0,4627**	0,5671	0,5081
BB+Bone	0,4557	0,3723	**0,8313**	**0,6848**	0,9673	0,9667	0,5870	0,4705	0,5730	0,5126
BB+Bone+TL	**0,4805**	**0,3766**	0,6715	0,5380	**0,9837**	**0,9836**	**0,6073**	**0,4657**	**0,6150**	**0,5207**

Table 13: Evaluation results for SAM ViT-Base with bounding box prompts (BB) that utilizes topological loss (TL) and Bone pseudocoloring during training.

	IoU		Accuracy		Specificity		Dice		mAP	
	Global	Sample	Global	Sample	Global	Sample	Global	Sample	Global	Sample
BB	0,4487	0,3660	0,7135	0,5677	**0,9767**	**0,9764**	0,5819	0,4590	0,5745	0,4950
BB-L	**0,4591**	**0,3731**	**0,8686**	**0,7265**	0,9660	0,9653	**0,5872**	**0,4704**	**0,5980**	**0,5146**
BB+Bone+TL	0,4805	0,3766	0,6715	0,5380	**0,9837**	**0,9836**	0,6073	0,4657	0,6150	0,5207
BB-L+Bone+TL	**0,5168**	**0,4305**	**0,7910**	**0,6995**	0,9824	0,9822	**0,6415**	**0,5217**	**0,6639**	**0,5652**

Table 14: Evaluation results for SAM ViT-Large with bounding box prompts (BB-L) that utilizes topological loss (TL) and Bone pseudocoloring schema during training.

Class	Prevalence	IoU		Accuracy	
		Global	Sample	Global	Sample
background	0,4029	0,8454	0,8235	0,9470	0,9371
vitreous body	0,2398	0,9272	0,2598	0,9683	0,3488
neurosensory retina	0,1402	0,8019	0,7269	0,9386	0,8288
image padding	0,1036	0,9570	0,8764	0,9878	0,9469
imaging artifacts	0,0712	0,6353	0,4881	0,8706	0,6609
retinal pigment epithelium	0,0120	0,2696	0,4302	0,7511	0,7339
pigment epithelial detachment	0,0075	0,3969	0,2568	0,6976	0,5086
fibrosis	0,0061	0,4692	0,4132	0,6523	0,5884
subretinal fluid	0,0057	0,4376	0,2841	0,5672	0,4425
intraretinal fluid	0,0038	0,4243	0,2242	0,7409	0,4763
epiretinal membrane	0,0025	0,1956	0,3959	0,7160	0,6169
choroid border	0,0021	0,1525	0,3290	0,5802	0,5365
subretinal hyperreflective material	0,0020	0,4979	0,2782	0,6631	0,4349
posterior hyaloid membrane	0,0006	0,2213	0,1507	0,5743	0,4817

Table 15: Per-class evaluation results (only IoU and Accuracy) for fine-tuned SAM ViT-Large model with topological loss and bone pseudocoloring (BB-L+TL+Bone), sorted by class prevalence.

Class	Prevalence	IoU		Accuracy	
		Global	Sample	Global	Sample
background	0,4029	0,4249	0,3958	0,6407	0,5934
vitreous body	0,2398	0,1919	0,0482	0,3154	0,0785
neurosensory retina	0,1402	0,0057	0,0050	0,0129	0,0094
image padding	0,1036	0,2530	0,2135	0,3857	0,2651
imaging artifacts	0,0712	0,0316	0,0197	0,0723	0,0414
retinal pigment epithelium	0,0120	0,0005	0,0029	0,0054	0,0079
pigment epithelial detachment	0,0075	0,0423	0,0273	0,1276	0,0463
fibrosis	0,0061	0,0143	0,0076	0,0264	0,0109
subretinal fluid	0,0057	0,0060	0,0098	0,0100	0,0122
intraretinal fluid	0,0038	0,0143	0,0106	0,0269	0,0190
epiretinal membrane	0,0025	0,0003	0,0023	0,0033	0,0045
choroid border	0,0021	0,0023	0,0133	0,0159	0,0250
subretinal hyperreflective material	0,0020	0,0730	0,0308	0,1391	0,0382
posterior hyaloid membrane	0,0006	0,0054	0,0035	0,0151	0,0085

Table 16: Per-class evaluation results (only IoU and Accuracy) for untrained SAM ViT-Large model, sorted by class prevalence.

Organoids

Additional images of the datasets

Figures 16, 17, 18 and 19 show more examples of the organoid images in the datasets used for the training and the evaluation of the model.

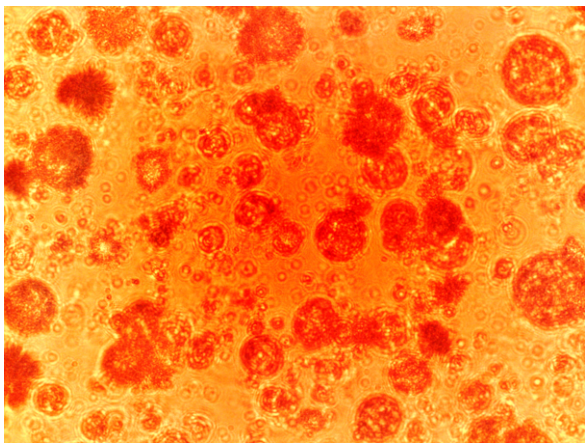


Figure 16: Orange image from the private dataset.

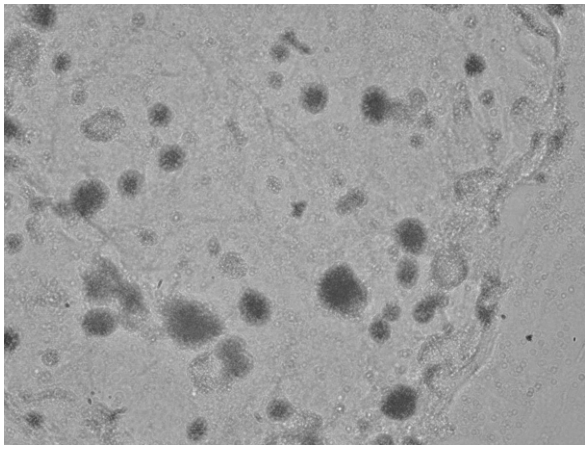


Figure 17: Black and white image from the private dataset.

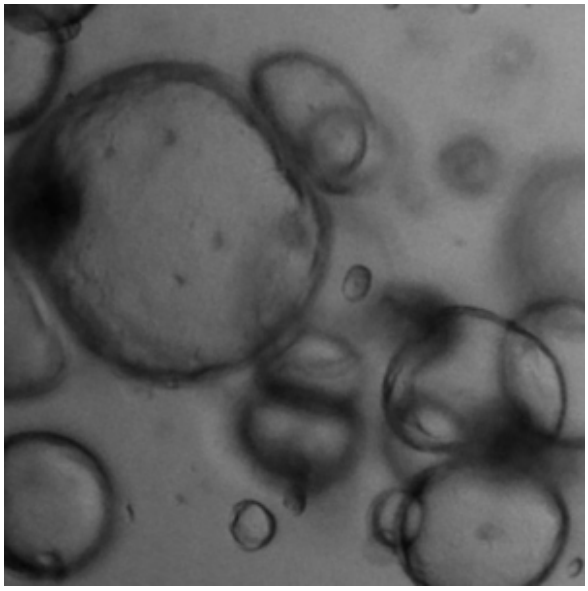


Figure 18: Image with intestinal organoids from Orgaquant dataset.

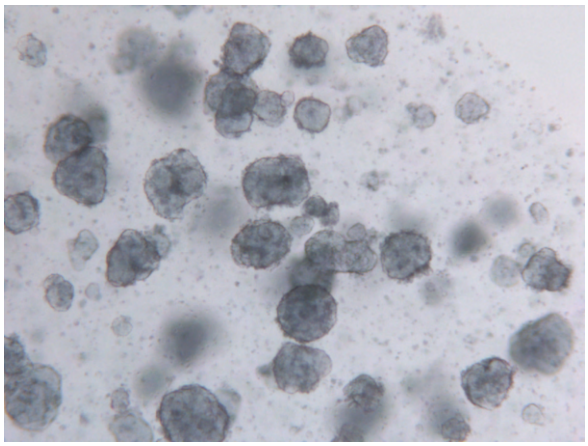


Figure 19: Image with colon organoids from OrganoSeg dataset.

Datasets information

Table 17 shows number of organoids in each split of every dataset used to evaluate and fine-tune the model.

	Split			
Dataset	**Train**	**Validation**	**Test**	**Total**
Private	16134	4630	4399	25163
OrgaQuant	13004	-	1135	14139
OrganoSeg	999	350	374	1723

Table 17: We have employed 3 different datasets. The private one has been used in the training. OrgaQuant and OrganoSeg have been hold-out datasets involved in the evaluation of the model.

Training hyperparameters

Table 18 and 19 specify the configuration and hyperparameters selected to fine-tune SAM ViT-Base with Topological Loss and Geometrical loss or only Geometrical loss, respectively.

Hyperparameter	Value
Train split images	16134
Validation split images	4630
Batch size	5
Epochs	10
Learning rate	5e-5
Weight decay	1e-4
Topological loss: λ	0.1
Topological loss: dimensional topological features included	0 and 1
Topological loss: L-p in Wasserstein distance	L2-loss
Topological regularisation	None

Table 18: Hyperparameters for the model trained with topological and geometrical loss.

Hyperparameter	Value
Train split images	16134
Validation split images	4630
Batch size	5
Epochs	10
Learning rate	5e-5
Weight decay	1e-4

Table 19: Hyperparameters for the model trained with geometrical loss.

Metrics results

Table 20, 21 and 22 provide all metric results for the hold-out test splits of private, OrgaQuant and OrganoSeg datasets. These are the mean over the metrics computed at

every mask of each corresponding dataset.

Model	IoU	Dice	Sens	Spec	AP	F1
Ground-truth box prompt						
MedSAM	0,606	0,711	0,695	0,993	0,860	0,711
SAM ViT-Base	0,701	0,796	0,891	0,972	0,876	0,796
SAM ViT-Large	0,722	0,811	0,914	0,976	0,899	0,811
SAM ViT-Huge	0,723	0,811	**0,924**	0,972	0,894	0,811
Fine-tuned SAM ViT-Base with Geom loss	0,792	0,864	0,899	0,991	0,953	0,864
Fine-tuned SAM ViT-Base with Topo+Geom loss	**0,798**	**0,869**	0,895	**0,994**	**0,956**	**0,869**
Noised box prompt						
MedSAM	0,510	0,621	0,574	0,993	0,793	0,621
SAM ViT-Base	0,636	0,737	0,801	0,971	0,823	0,737
SAM ViT-Large	0,663	0,757	0,827	0,975	0,847	0,757
SAM ViT-Huge	0,674	0,768	0,860	0,971	0,851	0,768
Fine-tuned SAM ViT-Base with Geom loss	0,783	0,857	**0,892**	0,991	0,951	0,857
Fine-tuned SAM ViT-Base with Topo+Geom loss	**0,790**	**0,863**	0,890	**0,993**	**0,953**	**0,863**

Table 20: Evaluation metrics for the hold-out test split of private dataset.

Model	IoU	Dice	Sens	Spec	AP	F1
Ground-truth box prompt						
MedSAM	0,644	0,763	0,688	1,000	0,953	0,763
SAM ViT-Base	0,945	0,967	0,970	1,000	0,992	0,967
SAM ViT-Large	**1,000**	**1,000**	**1,000**	**1,000**	**1,000**	**1,000**
SAM ViT-Huge	0,969	0,982	0,988	0,999	0,995	0,982
Fine-tuned SAM ViT-Base with Geom loss	0,670	0,790	0,989	0,996	0,942	0,790
Fine-tuned SAM ViT-Base with Topo+Geom loss	0,690	0,804	0,991	0,997	0,952	0,804
Noised box prompt						
MedSAM	0,502	0,634	0,538	0,998	0,881	0,634
SAM ViT-Base	0,704	0,775	0,733	0,999	0,850	0,775
SAM ViT-Large	0,686	0,751	0,707	**1,000**	0,865	0,751
SAM ViT-Huge	**0,738**	**0,801**	0,764	0,999	0,911	**0,801**
Fine-tuned SAM ViT-Base with Geom loss	0,671	0,789	0,980	0,996	0,928	0,789
Fine-tuned SAM ViT-Base with Topo+Geom loss	0,688	0,801	**0,983**	0,997	**0,942**	0,801

Table 21: Evaluation metrics for the hold-out test split of OrgaQuant dataset.

Additional visual examples of results

Figure 20, 21 and 22 show the outputs from our fine-tuned version of SAM, MedSAM and SAM ViT-Base on images from different datasets.

Model	IoU	Dice	Sens	Spec	AP	F1
Ground-truth box prompt						
MedSAM	0,626	0,751	0,651	0,999	0,942	0,751
SAM ViT-Base	0,801	0,881	0,840	**0,999**	0,929	0,881
SAM ViT-Large	0,807	0,886	0,847	0,999	0,945	0,886
SAM ViT-Huge	**0,810**	**0,889**	0,857	0,999	**0,948**	**0,889**
Fine-tuned SAM ViT-Base with Geom loss	0,742	0,837	0,895	0,996	0,926	0,837
Fine-tuned SAM ViT-Base with Topo+Geom loss	0,754	0,849	**0,917**	0,995	0,939	0,849
Noised box prompt						
MedSAM	0,546	0,681	0,566	0,999	0,922	0,681
SAM ViT-Base	0,733	0,831	0,769	0,999	0,904	0,831
SAM ViT-Large	0,737	0,834	0,774	**0,999**	0,927	0,834
SAM ViT-Huge	**0,770**	**0,860**	0,813	0,999	**0,936**	**0,860**
Fine-tuned SAM ViT-Base with Geom loss	0,734	0,830	0,886	0,995	0,912	0,830
Fine-tuned SAM ViT-Base with Topo+Geom loss	0,740	0,835	**0,900**	0,995	0,917	0,835

Table 22: Evaluation metrics for the hold-out test split of OrganoSeg dataset.

.

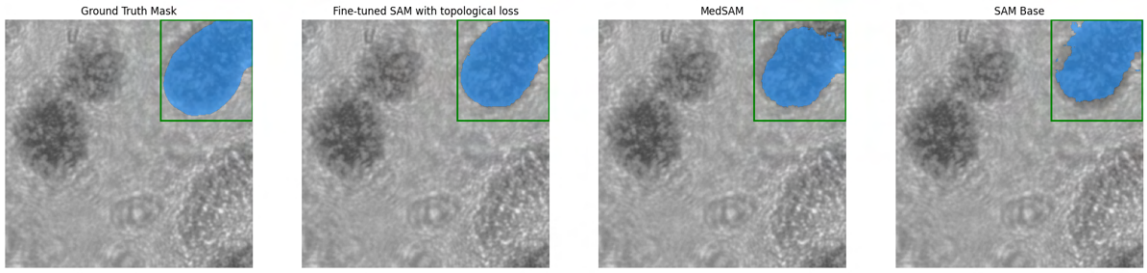


Figure 20: Different model outputs on image of the private dataset.

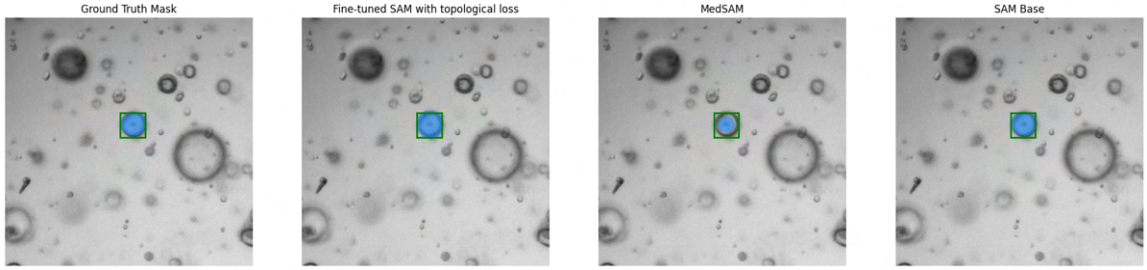


Figure 21: Different model outputs on image of OrgaQuant dataset.

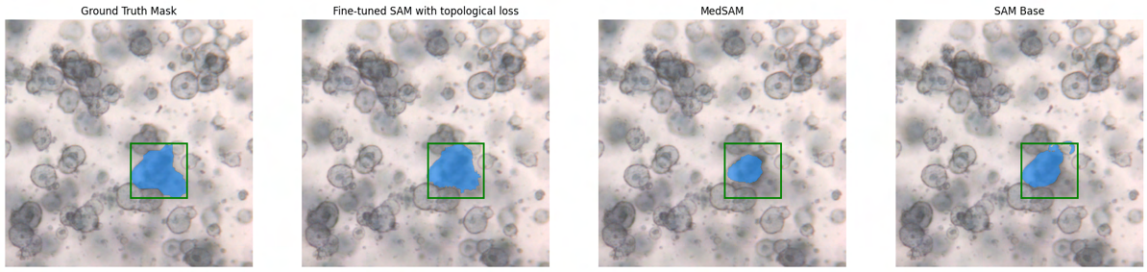


Figure 22: Different model outputs on image of OrganoSeg dataset.