Quantifying Topology In Pancreatic Tubular Networks From Live Imaging 3D Microscopy

Arnavaz, Kasra; Krause, Oswin; Zepf, Kilian; Bærentzen, Jakob Andreas; Krivokapic, Jelena Miskovic ; Heilmann, Silja; Nyeng, Pia; Feragen, Aasa

Quantifying Topology In Pancreatic Tubular Networks From Live Imaging 3D Microscopy

Kasra Arnavaz   kasra@di.ku.dk
Department of Computer Science, University of Copenhagen, Copenhagen, Denmark

Oswin Krause   oswin.krause@di.ku.dk
Department of Computer Science, University of Copenhagen, Copenhagen, Denmark

Kilian Zepf   kmze@dtu.dk
DTU Compute, Tecnhical University of Denmark, Lyngby, Denmark

Jakob Andreas Bærentzen   janba@dtu.dk
DTU Compute, Tecnhical University of Denmark, Lyngby, Denmark

Jelena M. Krivokapic   jelena.krivokapic@sund.ku.dk
DanStem, University of Copenhagen, Copenhagen, Denmark

Silja Heilmann   silja.heilmann@sund.ku.dk
DanStem, University of Copenhagen, Copenhagen, Denmark

Pia Nyeng*   pnyeng@ruc.dk
Department of Science and Environment, Roskilde University College, Roskilde, Denmark

Aasa Feragen*   afhar@dtu.dk
DTU Compute, Tecnhical University of Denmark, Lyngby, Denmark
* Shared last-authorship

Abstract

Motivated by the challenging segmentation task of pancreatic tubular networks, this paper tackles two commonly encountered problems in biomedical imaging: Topological consistency of the segmentation, and expensive or difficult annotation. Our contributions are the following: a) We propose a topological score which measures both topological and geometric consistency between the predicted and ground truth segmentations, applied to model selection and validation. b) We provide a full deep-learning methodology for this difficult noisy task on time-series image data. In our method, we first use a semisupervised U-net architecture, applicable to generic segmentation tasks, which jointly trains an autoencoder and a segmentation network. We then use tracking of loops over time to further improve the predicted topology. This semi-supervised approach allows us to utilize unannotated data to learn feature representations that generalize to test data with high variability, in spite of our annotated training data having very limited variation. Our contributions are validated on a challenging segmentation task, locating tubular structures in the fetal pancreas from noisy live imaging confocal microscopy. We show that our semi-supervised model outperforms not only fully supervised and pre-trained models but also an approach which takes topological consistency into account during training. Further, our approach achieves a mean loop score of 0.808 for detecting loops in the fetal pancreas, compared to a U-net trained with clDice with mean loop score 0.762.

Keywords: topological consistency, semisupervised, segmentation, tubular, confocal microscopy
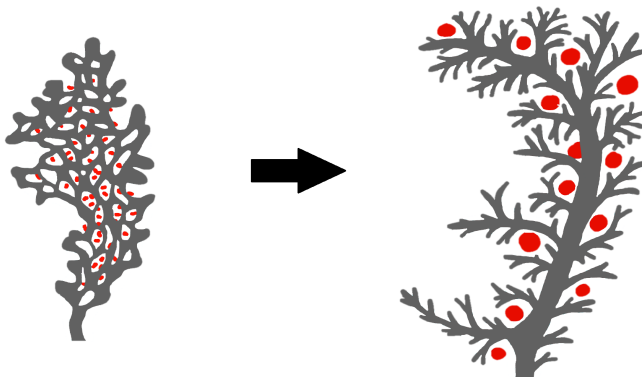
Figure 1: Tubular remodeling in pancreas during embryonic development. Gray areas represent tubes, while red dots represent β-cells.

1. Introduction

Segmentation of tubular structures is a common task in medical imaging, encountered when analyzing structures such as e.g. blood vessels (Zhang et al. (2019)), airways (Qin et al. (2019)), ductal systems (Wang et al. (2020b)), neurons (Li et al. (2019)). Segmentation is often a first step towards analysing the topological network structure of the organ, and it is therefore crucial that the segmented network structure is reliable.

Pancreatic tubes   This paper aims to solve a challenging tubular segmentation task from live fluorescence microscopy imaging. The pancreas produces enzymes and hormones, most importantly insulin which is produced by the so-called β-cells. Since insulin plays a crucial role in regulating blood sugar, malfunction of β-cells could lead to the development of diabetes.

The mammalian pancreas contains a tubular ductal network which transports enzymes into the intestines. Unlike superficially similar tubular structures such as the lung airways, the pancreatic tubes do not form by stereotypic branching. Rather, during embryonic development, the pancreatic tubes form by fusion of smaller lumens into a complex web-like network with many loops, which remodel to a tree-like structure, as discussed by Pan and Wright (2011) and Villasenor et al. (2010) and illustrated in Figure 1. As a result, the pancreatic tubes are complex and the overall structure varies from individual to individual, making segmentation and analysis of the network a hard task.

It has been suggested by Kesavan et al. (2009) and Bankaitis et al. (2015) that the emergence of β-cells depends on the remodeling of the tubular structure during embryonic development. To quantitatively verify this hypothesis, we require topologically accurate detection of the pancreatic tubes from noisy live imaging confocal 3D microscopy. In particular, we are interested in opening and closure of loops over time as the feature which might affect the emergence of β-cells. We tackle this problem for a dataset consisting of time-lapse 3D images from mouse pancreatic organs, which were recorded as the organs underwent the process of tubular re-organization and emergence of β-cells.
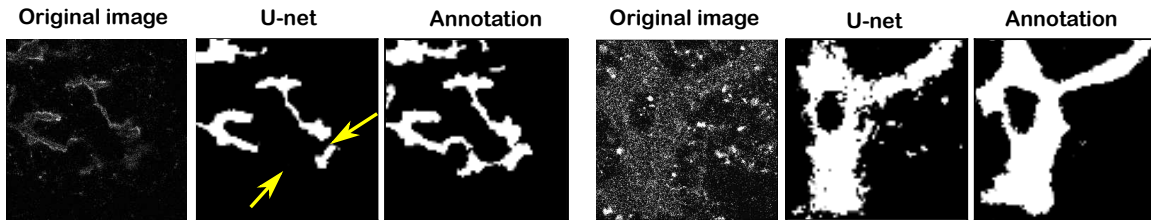
Figure 2: Visualized results on 2D slices. **Left:** The arrows point to errors in the U-net prediction which are small when measured in voxel overlap, but large when viewed as differences in the corresponding tubular network structure. **Right:** Noise and artefacts lead to challenges in automatic segmentation of tubular structures.

As can be seen from Figure 2 (right), these images have a low signal-to-noise ratio, which makes segmentation a difficult problem. Note that this is much lower than for other common tubular segmentation tasks, such as vessel segmentation. This is challenging not just because the images are noisy; also, the tubes appear at very different scales (compare the two images in Figure 2), and the image contrast is highest at the tubular boundary, meaning that tubes appear very different depending on their thickness relative to the image resolution as well as the depth in the image stack and local tissue depth.

Furthermore, most segmentation algorithms are trained to optimize voxel-based measures such as cross-entropy or accuracy. This, however, does not take into account how a small segmentation error viewed from the pixel point of view might, in fact, lead to a crucial error in the segmented network, see Figure 2 (left).

The challenge of annotation. While unlabeled data is abundant, manual segmentation of 3D tubes is time-consuming and can only be performed by a trained expert. This growing divide between availability of data and availability of annotations is generic in biomedical imaging: While the cost of image acquisition and storage has decreased significantly, the cost of annotation remains the same, and learning useful representations from unlabeled data is thus desirable.

As our application requires topologically accurate segmentations, it is also important to have examples of fully annotated 3D volumes in order to carefully assess the alignment of segmentations with annotations. As our training data consists of a few fully annotated 3D volumes, this means that the variation seen in the annotated training set is very limited.

Our contribution. We propose a segmentation- and processing pipeline that allows the quantification of topological features (with an emphasis on network loops) of pancreatic tubular networks from noisy live imaging 3D microscopy. This pipeline consists of i) a semisupervised training strategy accompanied with a manual annotation strategy supporting it; ii) a novel and highly interpretable topological score function which quantifies the topological similarity between the ground-truth tubular network and the predicted one; iii) the use of multi-object tracking to postprocess detected loops with the help of temporal data, filtering out those loops that do not remain present over time.

In particular, our topological score can be used both to rank the topological consistency of different models, as well as to encourage topological consistency through hyperparameter

3

tuning. We validate its use in both aspects against standard voxel-based model selection, as well as a topological loss function. We also compare with a segmentation network trained specifically with a topology-preserving loss function.

2. Related work

2.1 Topological consistency

This is not the first work to consider topological consistency of segmentation algorithms. A series of previous works define segmentation loss functions that encourage topologically consistent segmentations. Some of these are based on computational topology, such as Hu et al. (2019), Clough et al. (2020), Hu et al. (2020) or Wang et al. (2020a). Such losses compare the numbers of components and loops between ground truth and segmentation throughout a parameterized "filtration". While appealing, this approach suffers from two problems: First, these algorithms come with a high computational burden, which limit their application in typical biomedical segmentation problems: Early work on topological loss functions applied to 2D (or 2.5D) images, where computational limitations led to enforcing topological consistency of small patches rather than globally (Hu et al. (2019)). This is particularly problematic in our case, where the tubes appear on very different scales, making it hard to capture coarser scales with a single patch. This problem appears to persist with more recent work of Hu et al. (2020) utilizing discrete Morse theory. The topological loss function introduced by Clough et al. (2020) is computationally more feasible but requires the topology of the target object to be known a priori, even for test objects. This assumption holds for many tasks in bioimaging segmentation such as segmenting the 2D cross section of a tube or the boundary of a placenta imaged in 3D. However, this approach is not feasible for our setting, where we segment a full tubular structure with the goal of inferring its topology. For the same reason, methods such as Painchaud et al. (2020) or Lee et al. (2019), which use conditional variational autoencoders or registered shape templates to constrain the topology of the segmented object, are also not applicable to our segmentation problem.

A second problem with methods based on computational topology- is that they tend to focus on topological equivalence of structure without taking its *geometry* into account. More precisely, such methods tend to compare numbers of components and loops between prediction and ground truth without considering whether they actually *match*. This is suboptimal if two patches contain non-matching components or loops; a problem which does occur in our data.

Another direction of research in topologically consistent loss functions, initiated by Shit et al. (2021), are based on soft skeletonizations which are compared between prediction and ground truth. We find these losses more applicable to our type of data, and compare to such a loss function in our experiments.

We empirically find that topological loss functions are not helpful for our segmentation problem, and we hypothesize that this is caused by the challenging nature of our data. We choose to take a step back and address these problems through model selection, for models trained with voxel based loss functions. As we are not using our score function to train the model, we can allow ourselves to incorporate non-differentiable components such as geometric matching of loops and components. We thus introduce a topological score that measures topology preservation which is also geometrically consistent, in the sense

that topology is represented by network skeletons, whose components and loops are soft matched based on geometry. This ensures that our score function is really assessing each topological feature on a global scale, and not just counting that their numbers match up. This also leads to a highly decomposable and interpretable score, which both decomposes into loop and component scores, and into precision and recall scores. The final topological score is defined based on their agreement.

2.2 Semisupervised segmentation

The problem of attaining annotations is even more severe in biomedical segmentation. As a result, there have been numerous attempts to make use of the information from unlabeled images to improve the segmentation task.

One line of thought is to enforce feature embeddings of neighboring labeled and unlabeled images to be similar to one another. The work of Baur et al. (2019) defines an auxiliary manifold embedding loss to cause labeled and unlabeled images which are close in input space to be also close in latent space. The authors of Ouali et al. (2020) take a different approach to incorporating unlabeled data to learn better hidden representation features. They train a set of segmentation networks in the form of an encoder-decoder pair, where each network shares the same encoder. Only one of the decoders is trained on the labeled data, whereas all decoders predict a label map on the unlabeled data with the goal to minimize the distance between predictions. Variability between predictions is ensured via the introduction of random perturbations of the features generated by the encoder. Another approach by Zhang et al. (2017) uses adversarial networks. Along with a segmentation network, they also train a discriminator to distinguish between segmentation of labeled and unlabeled images. The authors in Cui et al. (2019) add noise to the same unlabeled data as a regularization for a mean teacher model (Tarvainen and Valpola (2017)).

A different semi-supervised approach is self-training (Zoph et al. (2020)). In this approach, a segmentation model is trained on labeled images and then applied to unlabeled images to get pseudo-labels which are then added to the labeled pool for retraining. Generative adversarial networks (GAN) are used by Hung et al. (2018) and Mittal et al. (2019) to select better pseudo-labels by training a discriminator that distinguishes between pseudo-labels and ground-truth labels. Inspired by curriculum learning of Bengio et al. (2009), the work of Feng et al. (2020) selects pseudo-labels gradually from easy to more complex. More recently, Chen et al. (2021) have combined the consistency regularization with self-training. They train two networks with the same structure but different initializations, one for labeled data and one for unlabeled. Then they enforce consistency between the two networks while using the pseudo-labels from one network as supervisory signal for the other.

In our paper, we propose jointly training a segmentation network and an autoencoder in order to obtain features that generalize across a large, unlabeled training set. Other architectures have appeared in the literature that combine autoencoders and segmentation networks for regularization. This includes the anatomically constrained network of Oktay et al. (2018), which utilizes an autoencoder to learn a global loss-function that enforces the network to produce anatomically realistic segmentations. Also related is the segmentation network used by Myronenko (2018). In this work, a variational autoencoder (VAE) branch is added to their original encoder-decoder architecture with a purpose to regularize the
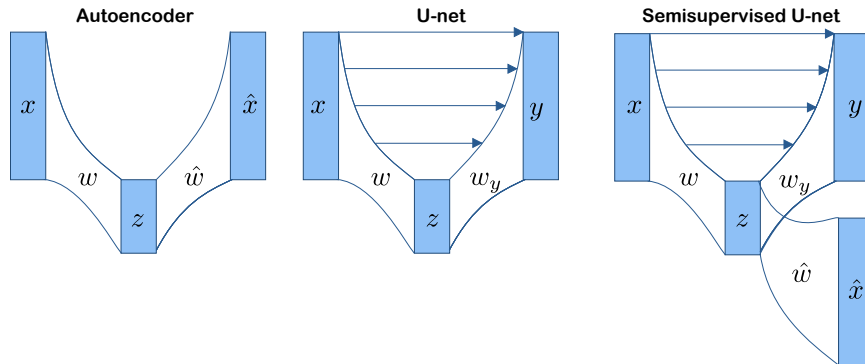
Figure 3: We apply semisupervised training of a segmentation U-net by combining an autoencoder (left) with a U-net (middle) into a combined network (right) which is simultaneously trained for segmentation and reconstruction. This allows us to learn features on a large, unlabeled dataset with rich variation, although we only have annotated segmentations for a small subset with little variation. In the architectures, x denotes the input image, y its corresponding binary segmentation map, and $\hat{x}$ is the reconstruction of the input image.

common encoder network. While both of these archictectures bear similarities with ours, their use is different, as they apply the network to labeled data as a means to regularize the training. In our work, however, we utilize the autoencoder as a means to learn better latent space features that represent our entire dataset in a situation where the labeled part of the data is very scarce.

3. Methods

We start by discussing our semisupervised segmentation algorithm along with our baselines. Next, we design a novel and interpretable score function for assessing topological correctness of segmentation in a geometry-aware fashion. Finally, we introduce a topology-oriented post-processing scheme which utilizes temporal data and multi-object tracking to filter detected topological features depending on their presence over time.

3.1 Semisupervised training of U-net

Our application offers abundant unlabeled data and a small labeled subset of limited variation – a very common situation in biomedical imaging. By learning features also from the unlabeled data, the increased diversity obtained by including unlabeled images might improve generalization to the full distribution of data expected at test time. To this end, we combine a 2D U-net (Ronneberger et al. (2015)) with an autoencoder, see Figure 3 (right). We refer to this as a semisupervised U-net.

The loss function for the semisupervised U-net is the weighted combination $L = L_R + \alpha L_S$ of the reconstruction loss L_R and the segmentation loss L_S where α is a hyperparameter. In our experiments, L_R is the mean squared error, and L_S is binary cross-entropy. The

segmentation loss is set to zero for unlabeled images. Although the segmentation decoder w_y is trained only on the labeled images, it is implicitly affected by the unlabeled data via its dependence on the encoder w which, along with $\hat{w}$, is trained on both labeled and unlabeled images.

We compare the semisupervised U-net framework to i) a U-net initialized randomly and trained solely on labeled images and ii) a U-net trained on labeled images, whose encoding weights w are initialized at the optima of an autoencoder trained on both labeled and unlabeled images. What separates the semisupervised U-net architecture from the U-net pre-trained on an autoencoder is the joint optimization of the two losses. We believe—as backed up by our experiments—that this joint optimization tailors the representations we learn from the unlabeled images to suit the segmentation task, and help them generalize to more diverse images than those annotated for training.

3.2 Topological score function

In order to perform model selection that prioritizes topologically consistent segmentations, as well as to evaluate our ability to correctly detect topological features such as loops and components, we design a topological score function. While existing topological loss functions often focus on getting the correct number of topological features such as components and loops, it is extremely important for our application to detect not only the correct number of loops, but the correct loops. In order to quantify this, our topological score therefore relies on first matching loops and components, and then measuring their correctness via overlap. In this way, the score function is also geometry-aware.

The loop- and component matching problem has some important differences from the graph matching problems widely covered in the literature. Our loops and components are retrieved via skeletonization of the manual segmentation and the estimated segmentation, respectively. We expect these skeleta to be qualitatively different: As the manual segmentations are more smooth than the estimated ones, their skeleta have significantly fewer nodes. This is illustrated in Figs. 4 and 9 One consequence of this is that traditional graph matching algorithms, which are typically designed to make a series of local compromises in order to optimize a global objective, would be very difficult to tune. Instead, we take advantage of knowing that the ground truth- and estimated skeleta both represent the same underlying geometric tubular system. We therefore expect these skeleta to be spatially close together, and we use local correspondences between points on the two skeleta to obtain loop- and component matching, as detailed below.

The topology of a segmented tubular structure is represented via its skeleton. We utilize a robust skeletonization algorithm recently proposed by Bærentzen and Rotenberg (2020) and then use NetworkX of Hagberg et al. (2008) to identify the loops and components, see Figure 4 for an example. We design our topological score via the following steps: i) matching skeletal nodes; ii) soft-matching components and loops using a point cloud Intersection-over-Union (IoU) score; and iii) collecting these into a topological score. By matching topological features based on geometric affinity, we ensure that our measured topological consistency is also *geometrically* consistent, meaning that the IoU score considers overlap of geometric features that appear near each other.
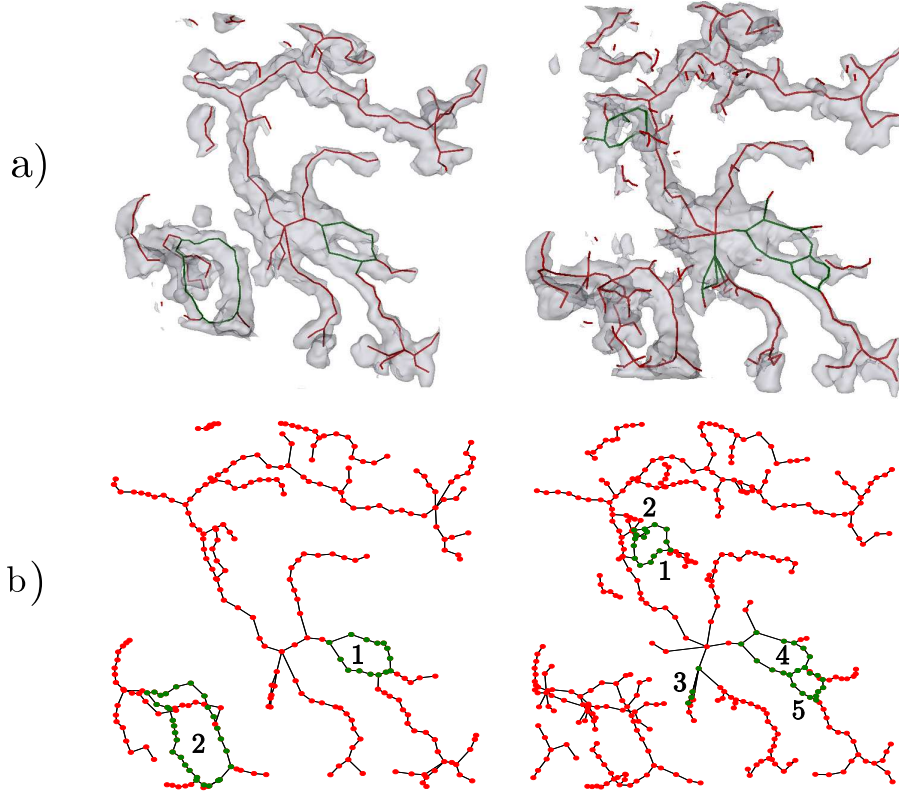
Figure 4: Ground truth (left) and prediction (right): a) segmentation and their skeletons. b) 2D projections of the skeletons onto the (x, y)-plane shown as nodes and edges. Green lines in a) and green nodes in b) take part in loops. Note that the ground truth segmentation has 2 loops and 7 components, while the prediction has 5 loops and 10 components. The loops are identified by numbers, for the sake of the discussion below.

For the example shown in Figure 4, each step explained below is illustrated in Figure 6.

Step i) Matching individual nodes. We denote the skeleton graphs of ground truth and prediction by S_{gt} and S_p, respectively. Due to noise, predicted segmentations are bumpier than the ground truth, giving higher node density in S_p than in S_{gt}. Examples of this phenomenon can be seen in Figs. 4 and 9. The topological score needs to be robust to this. As a first step towards creating a density-agnostic score, we therefore match skeletal components and loops by first matching their nodes as follows (see also Figure 5):

For every loop in one of the skeletons S_{gt}/S_p, each node is matched to its nearest node in a loop from the *opposite* skeleton S_p/S_{gt} if such a node exists within a fixed radius $r > 0$; otherwise the node is considered unmatched. Matching several nodes in one skeleton to the same node in the other skeleton is allowed, as can be seen from Figure 5.
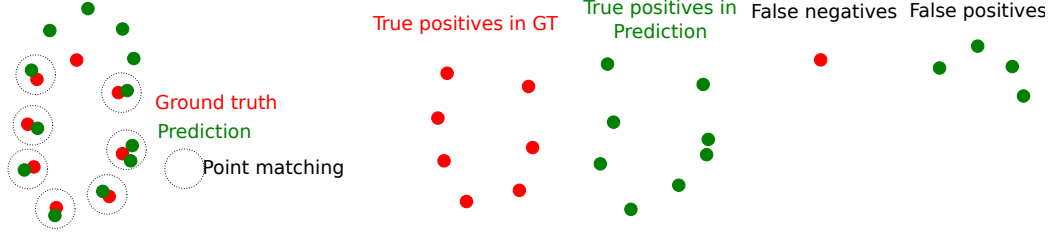
Figure 5: An example of point matching for two loops (edges not drawn).

The same matching procedure is applied to components.

Step ii) Soft matching of loops and components via Point Cloud IoU. Next, we perform partial matching of components and loops on nodes of the 3D skeleta of ground-truth and prediction. With skeleton nodes matched, some components or loops may overlap partly, as shown in Fig 5. We need to measure this, again being robust to differences in skeleton node density.

Define TP_p to be the number of nodes in the predicted loop that is matched to a point in the ground truth (GT) loop; TP_{gt} the number of nodes in the GT loop that is matched to a point in the predicted loop, FP the number of nodes in the predicted loop that is not matched to a point in the GT loop, and FN the number of loops in the GT loop that is not matched to a point in the predicted loop. Similarly for components. Now, we might naïvely define the loop's IoU score as

$$\frac{TP_{gt} + TP_p}{TP_{gt} + TP_p + 2FP + 2FN}.$$

However, this is not robust to differences in node density, as a very dense, but incomplete, prediction would get an inflated score. Instead, in analogy with samples from probability distributions, we define a point cloud IoU, which we use to measure loop-wise performance:

$$IoU = \frac{\dfrac{TP_{gt}}{TP_{gt} + FN} + \dfrac{TP_p}{TP_p + FP}}{\dfrac{TP_{gt} + 2FN}{TP_{gt} + FN} + \dfrac{TP_p + 2FP}{TP_p + FP}}, \tag{1}$$

where TP_{gt} and FN have been normalized by the number of nodes in the ground truth loop ($TP_{gt} + FN$), and TP_p and FP have been normalized by the number of nodes in the predicted loop ($TP_p + FP$).

Our score function depends highly on the matching radius r. As r is increased, it is more likely for nodes between distant loops or components to become matched and consequently, FP and FN become smaller. Thus, with large matching radius, IoU will become one. This makes intuitive sense, as large r indicate that different skeletonizations of the same structure have large spatial variability and this must be taken into account by the matching. A consequence of this is that r can only be chosen by hand, based on visual inspection and hand-annotation of matching pairs between ground-truth and prediction.

$$\begin{bmatrix} 0 & 0 & 0 & 0.86 & 0.29 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{Threshold}} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} \mathbf{1} \\ \mathbf{0} \end{matrix} \begin{matrix} \text{Row} \\ \text{scores} \end{matrix}$$

$$\begin{matrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{matrix}$$
Column scores

$\Longrightarrow$ Loop score $= \frac{1}{2}(\frac{1}{2} + \frac{1}{5})$
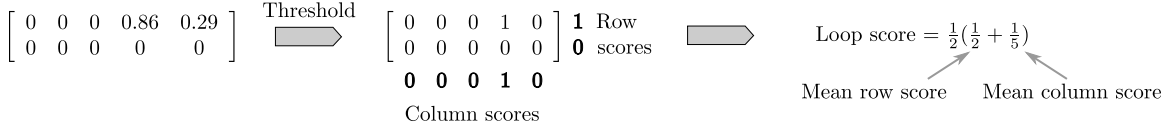
Mean row score   Mean column score

Figure 6: Example computation of the loop score for the prediction shown in Figure 4. The ground truth segmentation has 2 loops while the prediction has 5. Here, loop #1 in S_{gt} has overlap with loops #4 and #5 in S_p, where the first is thresholded as a perfect match and the second is thresholded as no match. The row score $= [1, 0]$ means loop #1 from the ground truth is very confidently matched to a predicted loop; the column score $= [0, 0, 0, 1, 0]$ means that the loop #4 from prediction is very confidently matched to a loop in ground-truth.

Step iii) A topological score for comparison of segmentations. Given skeletons S_{gt} and S_p, we separate our measure of topological consistency into performance for loops and performance for components. To this end, we define two topological scoring matrices; one for components and one for loops. As in the example of Figure 6, the loop score matrix has size $L_{gt} \times L_p$, where L_{gt} and L_p is the number of loops in S_{gt} and S_p, respectively. For each pair of GT and prediction loops, their matrix entry is given by their point cloud IoU of Eq. (1); this completes the matrix on the left side of Figure 6. Any IoU below a threshold t_{low} is set to 0, indicating no match, and any IoU over t_{high} is set to 1, indicating perfect match. If an IoU is between t_{low} and t_{high}, then it is left unchanged in the matrix. (Figure 6 middle matrix)

The thresholds t_{low} and t_{high} are necessary because without such a thresholding, we easily lose the distinction between many poor matches (with small but nonzero contribution to the score) and a single good match (with decent but not maximal contribution to the score). Without such a thresholding, even perfect matches will not give scores near 1 because of the dependence on the imperfect skeleta.

By adding horizontally we obtain a row score indicating the degree to which a ground truth loop is matched to a loop in the predicted topology. Similarly, summing vertically, the column score indicates the degree to which a predicted loop is matched to a loop in the ground truth topology. The elements in these row and column scores can, in principle, add up to more than one when nodes appear in multiple loops; they are clipped to be at most 1.

We can now take the mean of row scores and column scores to draw an analogy to confusion matrix terminology. The mean row score is similar to recall, rewarding fewer false negatives; the mean column score is similar to precision, rewarding fewer false positives; and the mean of the two mean scores is similar to the F1 score; we refer to this as the loop score (Figure 6 right side). Note that in this way, the topological score is decomposable into a series of highly interpretable sub-scores, which makes it both interpretable and adaptable to a wide range of applications.

A component score is defined in the same manner as the loop score. The final topology score is given by the mean of the loop and component scores.

The above does not handle cases where either S_{gt} or S_p have no loops (or components). If neither skeleton has loops, the loop score is 1. If S_{gt} has no loops, then the loop score is $1/(1 + \#err)$, where $\#err$ is the number of loops found in S_p. If the S_p has no loops, then the final score is $1/(1 + \#err)$, where $\#err$ is the number of loops found in S_{gt}. Components are handled similarly.

3.3 Postprocessing of loops using temporal information

In this section, we use the temporal structure of our data, which consists of consecutive 3D images taken over time, to filter out spurious loops as a postprocessing step. To do so, we first need to track loops, and then remove those trajectories which last a very short time.

We track loops by tracking their centers, and treat this as a multi-object tracking problem, whose validation can be broken into two questions: 1. Are predicted loop centers matched to a loop center in the ground-truth? 2. Do loop centers in different frames belong to the same loop? The first question is known as detection and the second as association. It is important to emphasize that the association of loops is implicitly affected by the detection of loops, which has been one of the topological features we have been interested in.

HOTA (Luiten et al. (2021)) is a multi-object tracking metric capturing both the detection accuracy (DetA) and the association accuracy (AssA). The final HOTA metric is defined as the geometric mean of DetA and AssA. The detection accuracy is based on a bijective matching between predicted- and ground-truth loop centers, where the distance between a matched pair cannot exceed a fixed threshold γ. To quantify association accuracy given predicted- and ground-truth trajectories of loop centers, Luiten et al. (2021) propose a measure which captures how accurate the alignment of predicted- and ground-truth trajectories is for every matched center. Finally, both detection- and association accuracies can be further broken down to recall (Re) and precision (Pr) constituents, i.e. DetRe, DetPr, AssRe, AssPr, which are useful for interpreting the results.

We use Trackpy (Allan et al. (2021)) to first track predicted loop centers,and next remove trajectories which last a short time. In order to do that, Trackpy includes a few hyperparameters: *search range* (the maximum distance loop centers can move between frames), *adaptive stop* (the value used to progressively reduce search range until solvable), *adaptive step* (the multiplier used to reduce search range by), *memory* (the maximum number of frames during which a loop can vanish, then reappear nearby, and be considered the same loop), and *threshold* (the minimum number of frames a feature has to be present in order not to be removed). These hyperparameters are tuned using the HOTA metric by grid-search on movies for which we have target trajectories annotated.

For components, we take a different approach as tracking component centers is not reasonable. At every time point, we compare all components on the current frame to all components from consecutive frames. Given a fixed component from the current frame, if there is no component in its 'vicinity' from neighboring frames then that particular component is removed. We use the minimum Euclidean distance between the nodes of two components as a measure of distance between them, and if this distance is lower than a threshold then the two components are in the same vicinity.

4. Data and preprocessing

The data used in this paper is extracted from 3D images obtained by live imaging of embryonic mouse pancreas (E14.0) explants. An image is recorded every 10 minutes during a period of 48 hours using a Zeiss LSM780 confocal microscope equipped for live imaging. The image resolution is $0.346 \times 0.346 \times 1.0 \ \mu m$ (x, y, z), and the dimension size is $1024 \times 1024 \times D$ pixels, where D varies between 27 and 40.

The imaged mouse pancreas expresses a reporter gene leading to production of a red fluorescent protein (mCherry), which is localized to the cell's apical side facing the tube's inner surface. In the images, we see the fluorescence as a bright tubular boundary surrounding a dark inner lumen, see Figure 7. Due to biological variation between individual samples, the fluorescence intensity is lower in some data sets, leading to a lower intensity boundary. In addition, low fluorescence is seen at the furthest end of the z-dimension due to intensity decay through the z-dimension. An additional segmentation problem is presented by the fact that the tubes have very varying diameters: Some tubes are so wide that the inner dark space can be mistaken for space between tubes. In other cases, segmentation is hard because the tube is so narrow that the dark inner space is hard to detect. These effects lead to a challenging tubular segmentation problem.

The problem is further complicated by several common artifacts (Figure 7 bottom row): First, we see imaging artifacts in the form of big clusters of very bright pixels in some images. These are caused by red blood cells and dead cells which fluoresce in the same wavelength as the reporter. There are also small clusters of bright pixels, caused by the production of reporter protein inside the cells, that are being transported to the tubular surface. These are mainly located at the end of the z-dimension where the cells attach to the dish and move significantly over time, which is helpful in ignoring them.

Intensity normalization   The mean intensity is close to zero, as the foreground volume is relatively small. The intensity variance, however, differs considerably across images, in particular due to the bright spot artifacts mentioned above. As preprocessing, every image is standardized to have zero mean and standard deviation 1, and image intensities are clipped to take values within 3 standard deviations.

Annotation strategy   As the image content can vary a great deal within a single image, each 3D image which was later fully or partially annotated, was first divided into a 4×4 grid of patches in the x-y plane, each of size $256 \times 256 \times D$. Those patches that had no discernible tubular structures were labeled as such and left out from the manual annotation and downstream analysis. We refer to these patches as "low information patches" below.

For the training set, 6 full 3D images from 3 different movies were manually segmented (2 3D images from each movie). The choice of annotating a few full 3D volumes for training was made in order to have some manually segmented examples available early on in which a complete tubular network was visible. For the sake of validating and testing on as diverse a dataset as possible, 10 3D patches were selected for validation, and 20 3D patches were selected for testing, in which the tubular structure was annotated by the laboratory assistant. There is no overlap in tissue samples between the test split and the training/validation splits.
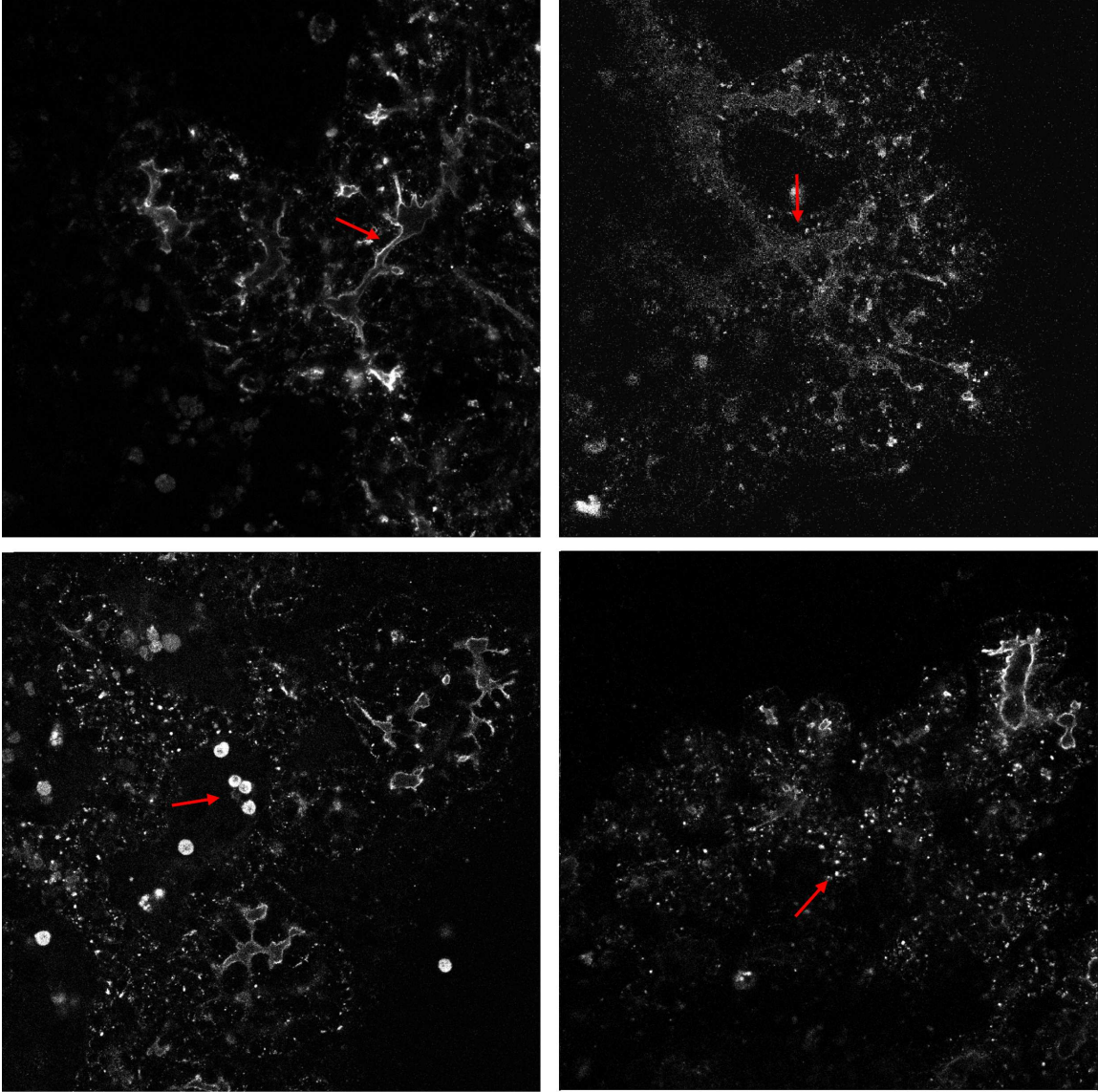
Figure 7: 2D slices from different movies extracted to show different image artifacts. **Top left:** a case where the boundary of tube has a high intensity. **Top right:** a case where the intensity of boundaries are no higher than the rest of the tube. **Bottom left:** large bright spots due to red blood cells and other dead cells. **Bottom right:** small bright spots due to the reporter protein in cells being transported to the tubular surface.

Training on patches   All our models are trained on image patches. Labeled training patches were obtained by drawing them at random from the 6 fully annotated 3D images; these patches were restricted not to overlap with those patches labeled as low information patches. The training set for the autoencoder and semisupervised U-net further includes

patches from 20 unannotated images from 10 movies, also referred to as unlabeled training set.

To avoid artifacts and degradation of performance near patch boundaries, we pad the image patches with a 32 pixel wide border, making boundary effects negligible. We thus use 320×320 patches, extracting predictions for the inner 256×256 region.

Annotated loops Our ultimate goal is to robustly detect loops in the tubular structure. To assess our ability to do so, loops were manually tracked and annotated over time for a small number of movies. More precisely, the x- and y- coordinates of loop centers were recorded in 5/1/3 movies from the unlabeled training set/validation set/test set, respectively. Moreover, the loops detected by our proposed pipeline (postprocessed output of semisupervised U-net) were assessed by a trained expert on 4 test movies.

5. Experiments and results

In Table 1, we compare three objectives for hyperparameter selection, i.e. voxel IoU, clDice measure, and the proposed topology score in this paper, and apply them to three models, namely a fully supervised 2D U-net, a U-net pre-trained on an autoencoder (AE+U-net), and the semisupervised U-net. Moreover, we take the same fully supervised U-net architecture and compare it to the case where the topology-preserving loss of Shit et al. (2021) has been used during training (clDice U-net).

Training setting All models are optimized using Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate for all architectures is 10^{-5}, except for clDice U-net which is set to 10^{-2} for better training; these were selected from the values $\{10^{-2}, 10^{-3}, 10^{-5}\}$ based on the robustness of the training- and validation set learning curves. Random weight initializations are done by Xavier uniform initializer, and biases are set to zero initially. The U-net and clDice U-net are trained for 200 epochs. For the AE+U-net, the autoencoder was trained for 20 epochs, which resulted in satisfactory reconstructions (Figure 8), and its U-net part was trained for 200 epochs. For the semisupervised U-net, the number of epochs was set to a 40, as it contains both labeled and unlabeled data. The hyperparameter α in the combined loss of the semisupervised U-net was selected from the range $\{0.1, 1, 10, 50\}$ by tuning performance on the validation set, obtaining $\alpha = 10$. The weighting between soft-Dice and clDice in the loss function of clDice U-net is set to be equal.

5.1 Segmentation performance

The hyperparameter that we tuned in our experiments was the cutoff threshold applied to the outputs of the sigmoid function to make a binary segmentation. The thresholds of interest ranged from 0.1 to 0.9 with a 0.1 increment. The thresholds performing best on the validation set were applied to the test set, and then skeletons were computed using the robust GEL skeletonization algorithm (Bærentzen and et al. (2020)). Skeleton components of size < 5 nodes were ignored; the radius $r = 10$ pixels was used for node matching; and the thresholds $t_{low} = 0.3$ and $t_{high} = 0.7$ were used for the component and loop score matrices. The radius r was chosen by visual inspection (using the training- and validation sets) of differences between ground truth- and estimated skeleta on the validation set. The

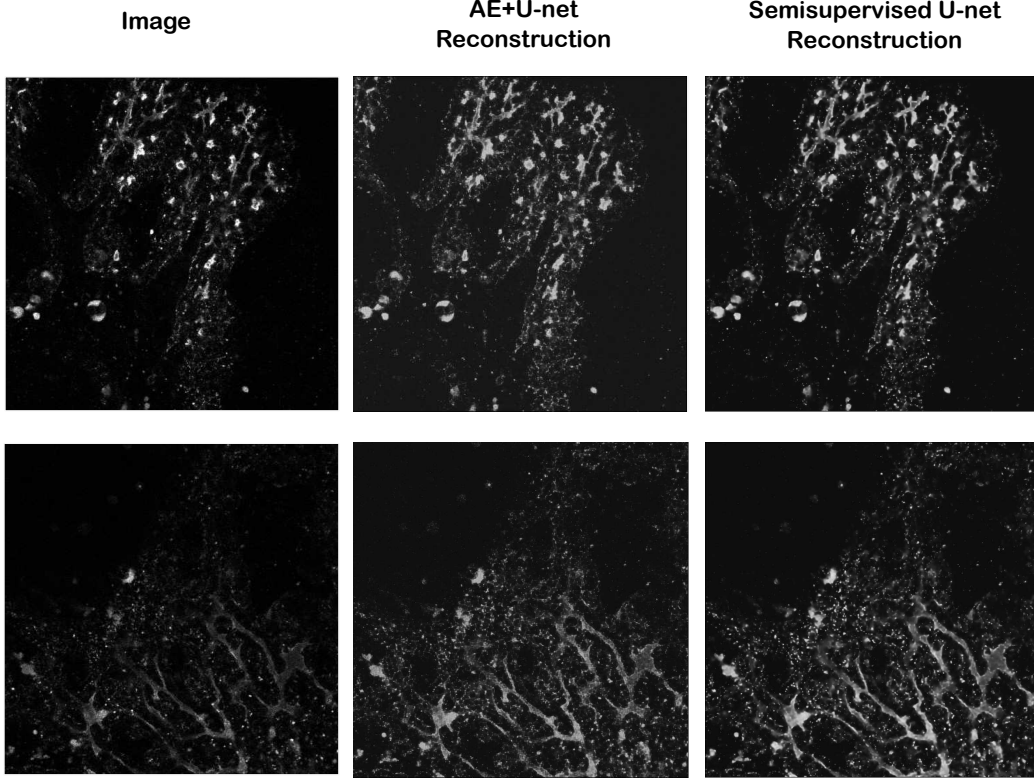| Image | AE+U-net Reconstruction | Semisupervised U-net Reconstruction |

Figure 8: Two 2-dimensional slices from the test set (left column) and their corresponding reconstruction at the final epochs of AE+U-Net (middle column), and semisupervised U-net (right column).

thresholds t_{low} and t_{high} were similarly set by visual inspection on the validation set of those loop matching being thresholded to the values 1 or 0.

In Table 1, we report mean and standard deviation of patch-wise scores on the test set. Note that this standard deviation indicates variation in performance over different patches, not robustness over multiple training runs. It is worth noting that, given a fixed model, all performance measures are only dependant on the threshold value. As a result if multiple tuning criteria reach the same threshold value for a model, all performance measures for those tuning criteria would be identical. For visual comparison, Figure 9 shows four randomly selected patches from the test set to compare the ground truth annotation with different model predictions at their best-performing thresholds according to the topological score. As we believe the difficulty of the segmentation problem varies greatly among different patches, in Figure 10 we have plotted topology score against entropy (left) and distribution of topology-, component- and loop scores for each level of difficulty, evaluated by an expert. Both plots use the test set predictions of the semisupervised U-net.

Table 1: Segmentation performance assessed via voxel IoU, clDice measure, the topological score, and its sub-components, the loop score and the component score. (Sec. 3.2). For each of the 3 models, the best-performing threshold (Thr) was selected according to Voxel IoU, clDice measure, and Topological Score on the validation set. Numbers in bold indicate the highest mean across both architectures and tuning criteria. Note that the three rows for "Semisupervised U-net" are identical because all tuning criteria chose the same segmentation threshold.

The last row shows the performance of "clDice U-net", a U-net model trained with the topology-preserving loss "clDice" of Shit et al. (2021). As the pixel-wise probability values of this model clustered as either less than 0.1 or more than 0.9 on the validation set, we chose 0.5 as the threshold to apply on the test set.

Architecture	Tuning Criterion	Thr	Voxel IoU	clDice	Topological Score	Loop Score	Component Score
	Voxel IoU	0.8	0.474±0.208	0.535±0.192	0.395±0.263	0.312±0.297	0.477±0.285
U-net	clDice	0.8	0.474±0.208	0.535±0.192	0.395±0.263	0.312±0.297	0.477±0.285
	Topology Score	0.9	0.449±0.214	0.529±0.212	0.424±0.265	**0.390±0.339**	0.458±0.297
	Voxel IoU	0.5	0.447±0.210	0.504±0.196	0.356±0.243	0.308±0.299	0.404±0.264
AE+U-net	clDice	0.6	0.447±0.207	0.504±0.206	0.382±0.264	0.333±0.320	0.430±0.279
	Topology Score	0.7	0.418±0.214	0.488±0.219	0.375±0.222	0.325±0.311	0.425±0.271
	Voxel IoU	0.7	**0.490±0.226**	**0.555±0.213**	**0.436±0.253**	0.386±0.352	**0.486±0.271**
Semisupervised U-net	clDice	0.7	**0.490±0.226**	**0.555±0.213**	**0.436±0.253**	0.386±0.352	**0.486±0.271**
	Topology Score	0.7	**0.490±0.226**	**0.555±0.213**	**0.436±0.253**	0.386±0.352	**0.486±0.271**
	clDice U-net	0.5	0.449±0.193	0.443±0.187	0.337±0.263	0.314±0.346	0.360±0.275

5.2 Loop tracking and filtering

We use the temporal data to remove false positives from the detected loops. Due to computational constraints, we only apply this to the best performing model, i.e the semisupervised U-net, and the clDice U-net.

The hyperparameters of Trackpy, used to track loops as well as filter them, were tuned on 6 movies with annotated loops. Of these, 5 came from the unlabeled training set and 1 came from the validation set. The resulting optimal hyperparameters are found in Table 2. We have used Euclidean distance to match loop centers, and the upperbound for the distance in HOTA measure is $\gamma = 50$ pixels.

After filtering, the loop scores on the test set patches were improved from **0.386±0.352** to **0.808±0.281** for the the semisupervised U-net and from **0.314±0.346** to **0.762±0.301** for the clDice U-net.

These final results were also evaluated by a trained expert, who annotated true positive (TP), false positive (FP), and missing (false negative, or FN) loops. This was carried out for 4 full movies from the test set; see Figure 11. As can be seen from the results, there is a strong correlation between predicted and true loops. In the bottom right movie, we see a large number of false positives; this movie was also annotated as challenging by the trained expert due to the high number of loops and increase in noise towards the end of the movie.

Table 3 shows the HOTA scores and sub-scores for the 3 movies in the test set for which we have had manual tracking of loop centers.
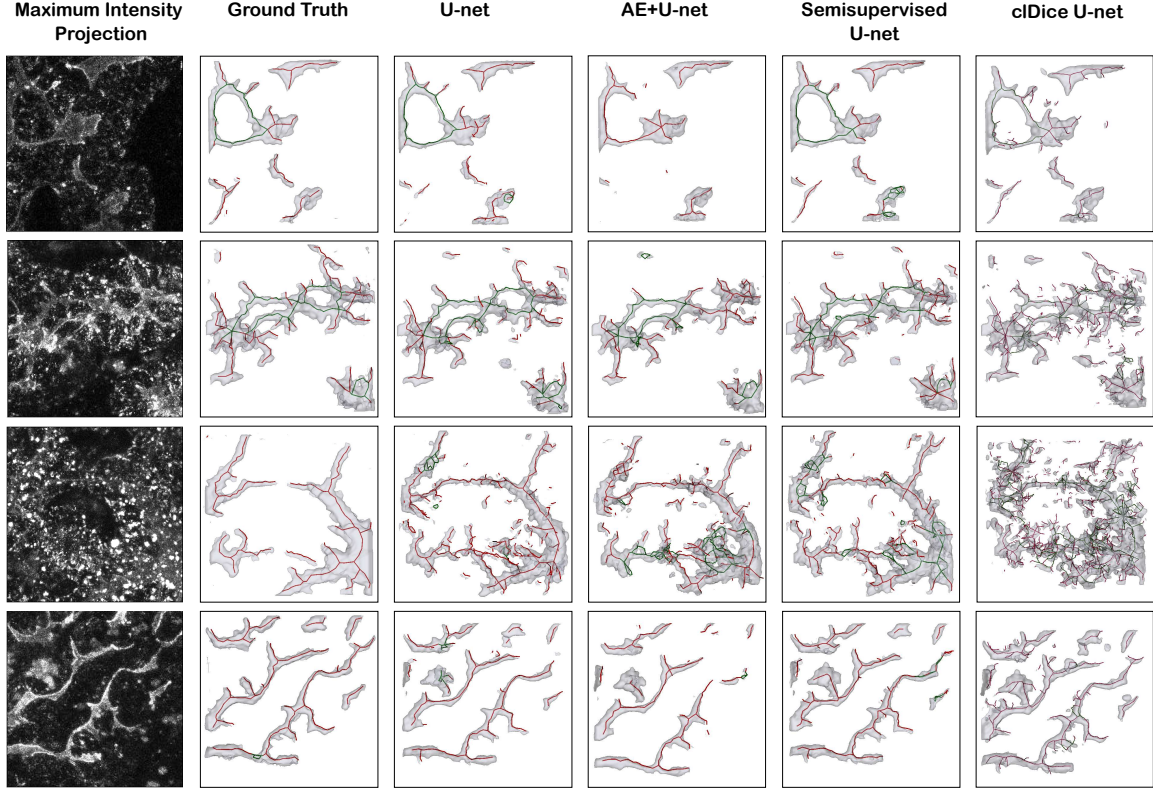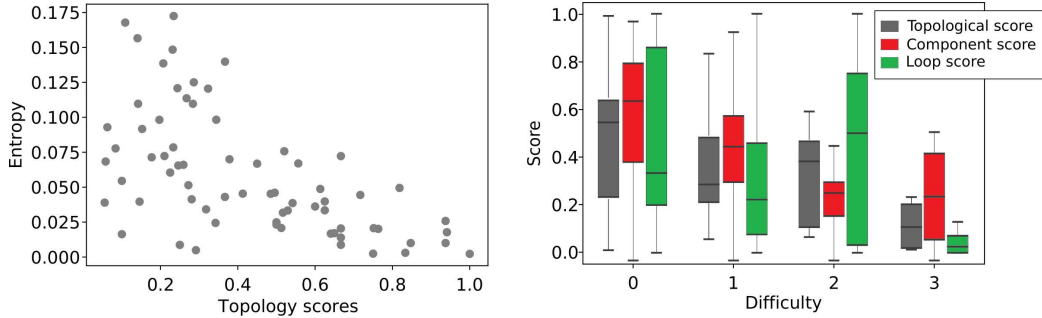
Figure 9: Segmentation results for 4 example test patches.

Figure 10: **Left**: Patch-wise topology score versus segmentation entropy. **Right**: Distribution of topology, component and loop scores divided over difficulty of manual labeling.

6. Discussion and Conclusion

We have tackled the challenging problem of segmenting pancreatic tubes from live-imaging microscopy, in order to track loops in the complex and dynamic tubular network that they form. This came with three important challenges: 1) Having a topologically accurate segmentation in order to correctly track the loops of the tubular network; 2) Utilizing
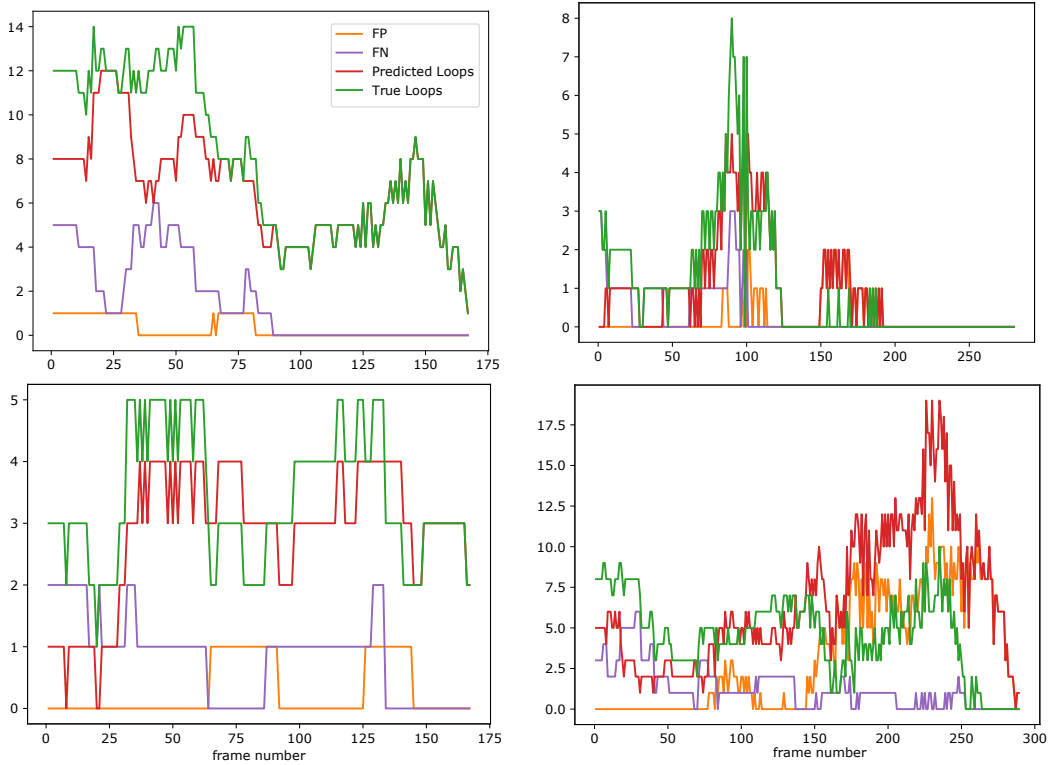
17

Figure 11: Each plot is a movie from the test set showing the performance of loop detection evaluated by an expert. The y-axis measures the number of false positives (FP), false negatives (FN), predicted loops, and true loops.

Table 2: Parameters of tracking and filtering loop centers in Trackpy for two architectures.

Architecture	search range	adaptive stop	adaptive step	memory	threshold
Semisupervised U-net	15	5	0.9	1	15
clDice U-net	10	5	0.9	1	12

temporal data to postprocess the segmentation results; and 3) Utilizing unlabeled data to aid the segmentation in generalizing to diverse test data, as our annotated training data has very limited variation.

To obtain topologically accurate segmentations, we derive a topological score, which splits into loop- and component scores, to quantify the topological correctness of the segmentations. To aid generalization, we train a segmentation U-net in a semisupervised fashion, combining an autoencoder reconstruction loss applied to a large set of un-annotated images, with a segmentation loss applied to the annotated images. Finally, we make use of large amounts of unannotated temporal data to enhance temporal consistency in our detected loops.

18

Table 3: Performance of tracking loop centers for two architectures.

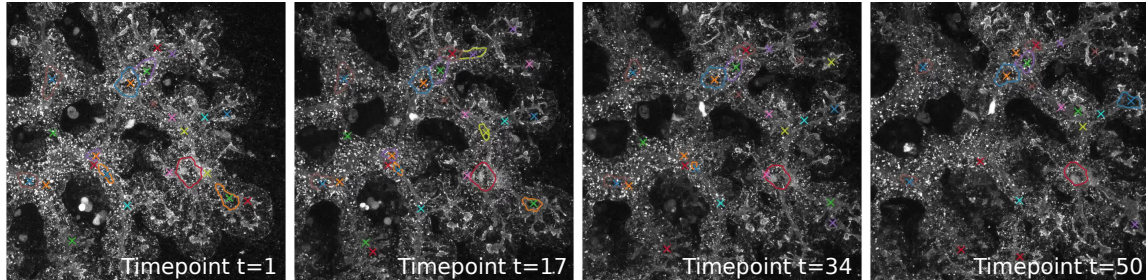| Architecture	HOTA	DetA	DetRe	DetPr	AssA	AssRe	AssPr
Semisupervised U-net	0.337	0.256	0.361	0.649	0.469	0.478	0.956
clDice U-net	0.275	0.233	0.279	0.705	0.324	0.330	0.944

Figure 12: Frames from one of the test set movies with loops manually annotated by an expert shown as crosses. The predicted loops are shown as colored loops. Note that most of the mistakes made are false negative, i.e. not detecting a loop that was annotated by the expert. This supports the use of the detected loops for testing biological hypotheses, in the sense that the detected loops are likely to be real.

A topological score function   We have derived a topological score function from highly interpretable constituents for both loops and components, which split into subscores analogous to precision and recall. These are joined into F1-like scores, again for loops and components individually, before merging them to a final topological score.

While from a biological point of view, we are primarily interested in the loops, the topological consistency of components is likely to be highly related to the topological consistency of loops, and we thus find it valuable to include both for model selection in order to base our selections on more data. This compositionality of the score makes it highly versatile and interpretable.

Topological model selection   We seek to select those model hyperparameters that produce the best segmentations from a topological point of view. The topological score provides one way to quantify which segmentations are "best", but alternative scores may produce different "best" models. Indeed, in the context of hyperparameter tuning (Section 5.1), we see, comparing the performance measures in Table 1, that a given model typically performs best on the particular metric on which its hyperparameter has been tuned. This emphasizes the importance of selecting a topological score function that fits the application well.

Topological loss functions   In our paper, the enforcement of topological consistency from the network's point of view was restricted to tuning the thresholding hyperparameter, while the neural network training used a standard voxel-wise cross-entropy loss. This differs from recent works deriving topological loss functions that encourage topological consistency also in the training of the segmentation network (Shit et al. (2021); Hu et al. (2019); Clough

et al. (2020); Hu et al. (2020)). We consider having a topological-preserving loss to be a very promising strategy, as it has been demonstrated to work well on a number of simpler segmentation problems. In our experiments, however, we find that topology-preserving loss functions were not sufficiently powerful: the U-net trained with the pixel-wise loss and hyper-parameters tuned on our topology score outperforms the U-net trained using the clDice loss across all measures (see Table 1 and Figure 9).

We believe that this is caused in part by the challenging nature of our segmentation problem. The signal-to-noise ratio in our dataset is low and we have a number of artifacts. Moreover, our tubular structures appear at very different scales, meaning that training with small patches, which is e.g. performed by Hu et al. (2019) to obtain scalability, would likely have trouble capturing all scales. These issues are also likely the reason behind the suboptimal results obtained using the soft-skeletonization algorithm of Shit et al. (2021) on our data (see Figure 9).

Note that both the signal-to-noise-ratio and the scale challenge are different from what we typically see in e.g. vessel- or road segmentation. We believe this is also part of the explanation why methods developed for such tasks are challenged on our dataset.

Segmentation models and data Out of the three architectures we experimented with, the semisupervised U-net has proven to outperform the other two, regardless of which performance measure is used (see Table 1). The joint optimization of the autoencoder and the U-net seems to have driven latent space representations to be more suitable for the segmentation task, compared to the pretrained AE+U-net which acts as an initialization for the U-net. Slightly more surprising is the fact that AE+U-net is outperformed by a randomly initialized U-net. However, this was observed over multiple training runs, indicating that pretraining actually led to a worse initialization. Similar effects have also been observed for natural images in He et al. (2019). We speculate that this is due to the artifacts present in our data: The autoencoder may emphasize reconstruction of artifacts, which does not behave as conventional noise.

The convolutions in all our architectures were in 2D. While the 3D U-net of Çiçek et al. (2016) utilizes useful spatial structure, it also has more parameters, which with our limited labeled data led to poorer performance than the 2D counterpart.

Our annotation strategy assumed that a few, detailed annotations of full 3D images would be sufficient to train a good segmentation model, and we chose to focus on having more variance in the validation and test images. To verify that this was a sound assumption, we experimented with swapping the role of the validation and training sets, i.e. training the 2D U-net on the validation set, and using the training set to tune its hyperparameters. The results were again evaluated on the test set. However, the performance did not improve compared to the U-net reported in Table 1.

Although the semisupervised U-net utilized the diversity of unlabeled data to improve segmentation, its predictions are not perfect. As shown in Figure 10, our performance correlates negatively with segmentation entropy as well as with segmentation difficulty as assessed by the trained expert. Thus, endowing the extracted tubular network with a notion of uncertainty could ensure safe interpretation.

Loop tracking   Using temporal information to filter out loops which were present only a short time, helped increase loop score drastically (see Sec. 5.2). This postprocessing step has proved useful in coping with artifacts and noise in our data.

As mentioned before, having accurate detection and tracking of tubular loops is biologically important. Thus, we evaluated our model performance against expert opinion. Table 3 provides us with both detection and tracking performances. One can observe that the detection precision is higher than detection recall, meaning there are comparably more FN than FP. This is also partially supported by the results in Fig. 11.

Conclusion and Outlook   In this paper, we tackled segmenting pancreatic tubular networks, which are dissimilar to other anatomical networks in several aspects. In doing so, we proposed a novel and highly interpretable topological score function, which we used for evaluation as well as model selection. We also compared various training schemes, showing that it is advantageous to incorporate unlabeled data to learn more generalizable latent features. Finally, we showed that postprocessing the detected loops by applying multi-object tracking over time to greatly reduced false positive loops.

Compared to the segmentation of common tubular structures such as vessels or neurons, our images are challenging and our segmentations are imperfect. Recall, however, that the purpose of our segmentation is to provide topological features for testing hypotheses regarding the development of pancreatic organs. One alternative to automatic segmentation is manual segmentation of individual time point volumes, which is too time consuming to be feasible. The other remaining alternative is to manually annotate the topological features (existence of loops and beta cells). This forms the basis of previous biological papers such as Kesavan et al. (2009) and Bankaitis et al. (2015). However, such manual annotation, which is motivated by finding a dependency between tubular structure and nearby beta cells, is also at risk of being biased: A human who expects to find cells near loops, may also have a tendency to over-annotate loops near cells. An automatic method for quantifying loops does not suffer from this bias. Therefore, an imperfect segmentation still may have great value in its ability to confirm – or not confirm – the underlying biological hypothesis in an unbiased way.

Our segmentation problem is challenging, and in addressing it, we have experienced that simple, established methods are more robust than complex ones. By remaining in the realm of established methods, we are able to define a topological score which is also faithful to geometry, which we believe adds robustness. Going forwards, we hope that utilizing geometry to strengthen the matching of topological features, may form a starting point for more robust loss functions that can also be used to train topologically consistent models.

Acknowledgments

Ethical Standards

The work follows appropriate ethical standards in conducting research and writing the manuscript, following all applicable laws and regulations regarding treatment of animals or human subjects.

Conflicts of Interest

We declare we don't have conflicts of interest.

References

Daniel B. Allan, Thomas Caswell, Nathan C. Keim, Casper M. van der Wel, and Ruben W. Verweij. soft-matter/trackpy: Trackpy v0.5.0, April 2021. URL `https://doi.org/10.5281/zenodo.4682814`.

Andreas Bærentzen and Eva Rotenberg. Skeletonization via local separators. *arXiv preprint arXiv:2007.03483*, 2020.

Eric D Bankaitis, Matthew E Bechard, and Christopher VE Wright. Feedback control of growth, differentiation, and morphogenesis of pancreatic endocrine progenitors in an epithelial plexus niche. *Genes & development*, 29(20):2203–2216, 2015.

Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Fusing unsupervised and supervised deep learning for white matter lesion segmentation. In *International Conference on Medical Imaging with Deep Learning*, pages 63–72, 2019.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Andreas Bærentzen and et al. The GEL library. `http://www2.compute.dtu.dk/projects/GEL/`, 2020.

Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf 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.

J Clough, N Byrne, I Oksuz, VA Zimmer, JA Schnabel, and A King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Wenhui Cui, Yanlin Liu, Yuxing Li, Menghao Guo, Yiming Li, Xiuli Li, Tianle Wang, Xiangzhu Zeng, and Chuyang Ye. Semi-supervised brain lesion segmentation with an

adapted mean teacher model. In *International Conference on Information Processing in Medical Imaging*, pages 554–565. Springer, 2019.

Zhengyang Feng, Qianyu Zhou, Guangliang Cheng, Xin Tan, Jianping Shi, and Lizhuang Ma. Semi-supervised semantic segmentation via dynamic self-training and classbalanced curriculum. *arXiv preprint arXiv:2004.08514*, 1(2):5, 2020.

Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019.

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5658–5669, 2019.

Xiaoling Hu, Yusu Wang, Li Fuxin, Dimitris Samaras, and Chao Chen. Topology-aware segmentation using discrete morse theory. In *International Conference on Learning Representations*, 2020.

Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*, 2018.

Gokul Kesavan, Fredrik Wolfhagen Sand, Thomas Uwe Greiner, Jenny Kristina Johansson, Sune Kobberup, Xunwei Wu, Cord Brakebusch, and Henrik Semb. Cdc42-mediated tubulogenesis controls cell specification. *Cell*, 139(4):791–801, 2009.

Matthew Chung Hai Lee, Kersten Petersen, Nick Pawlowski, Ben Glocker, and Michiel Schaap. Tetris: Template transformer networks for image segmentation with shape priors. *IEEE Transactions on Medical Imaging*, 38(11):2596–2606, 2019. doi: 10.1109/TMI.2019.2905990.

Rui Li, Muye Zhu, Junning Li, Michael S Bienkowski, Nicholas N Foster, Hanpeng Xu, Tyler Ard, Ian Bowman, Changle Zhou, Matthew B Veldman, et al. Precise segmentation of densely interweaving neuron clusters using g-cut. *Nature communications*, 10(1):1–12, 2019.

Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129(2):548–578, 2021.

Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high-and low-level consistency. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Andriy Myronenko. 3d mri brain tumor segmentation using autoencoder regularization. In *International MICCAI Brainlesion Workshop*, pages 311–320. Springer, 2018.

Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitsas, Mattias Heinrich, Wenjia Bai, Jose Caballero, Stuart A. Cook, Antonio de Marvao, Timothy Dawes, Declan P. O'Regan, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Anatomically constrained neural networks (acnns): Application to cardiac image enhancement and segmentation. *IEEE Transactions on Medical Imaging*, 37(2):384–395, 2018. doi: 10.1109/TMI.2017.2743464.

Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020.

Nathan Painchaud, Youssef Skandarani, Thierry Judge, Olivier Bernard, Alain Lalande, and Pierre-Marc Jodoin. Cardiac segmentation with strong anatomical guarantees. *IEEE Transactions on Medical Imaging*, 39(11):3703–3713, 2020. doi: 10.1109/TMI. 2020.3003240.

Fong Cheng Pan and Chris Wright. Pancreas organogenesis: from bud to plexus to gland. *Developmental Dynamics*, 240(3):530–565, 2011.

Yulei Qin, Mingjian Chen, Hao Zheng, Yun Gu, Mali Shen, Jie Yang, Xiaolin Huang, Yue-Min Zhu, and Guang-Zhong Yang. Airwaynet: A voxel-connectivity aware approach for accurate airway segmentation using convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 212–220. Springer, 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Suprosanna Shit, Johannes C Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien PW Pluim, Ulrich Bauer, and Bjoern H Menze. cldice-a novel topology-preserving loss function for tubular structure segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16560–16569, 2021.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

Alethia Villasenor, Diana C Chong, Mark Henkemeyer, and Ondine Cleaver. Epithelial dynamics of pancreatic branching morphogenesis. *Development*, 137(24):4295–4305, 2010.

Fan Wang, Huidong Liu, Dimitris Samaras, and Chao Chen. Topogan: A topology-aware generative adversarial network. In *European Conference on Computer Vision*, volume 2, 2020a.

Yan Wang, Xu Wei, Fengze Liu, Jieneng Chen, Yuyin Zhou, Wei Shen, Elliot K Fishman, and Alan L Yuille. Deep distance transform for tubular structure segmentation in ct scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3833–3842, 2020b.

Shihao Zhang, Huazhu Fu, Yuguang Yan, Yubing Zhang, Qingyao Wu, Ming Yang, Mingkui Tan, and Yanwu Xu. Attention guided network for retinal image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 797–805. Springer, 2019.

Yizhe Zhang, Lin Yang, Jianxu Chen, Maridel Fredericksen, David P Hughes, and Danny Z Chen. Deep adversarial networks for biomedical image segmentation utilizing unannotated images. In *International conference on medical image computing and computer-assisted intervention*, pages 408–416. Springer, 2017.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*, 2020.